SCALESWEEP: ACCURATE NVFP4 POST-TRAINING QUANTIZATION OF LLMS VIA BLOCK SCALE INITIALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

NVFP4 is a recently introduced hardware-supported FP4 format that improves the fidelity of 4-bit quantization through fine-grained block scales. However, existing NVFP4 scale initialization methods still rely on AbsMax initialization, which leaves a noticeable gap to the optimal solution. To address this, we propose ScaleSweep, a simple and efficient scale optimization method that sweeps over block scale candidates and selects the candidate that minimizes a target objective. We further provide a theoretical analysis of NVFP4 quantization and derive both lower and upper bounds for the required sweep range under mean square error (MSE) and weighted mean square error (WMSE) between the original tensor and the quantized reconstructed tensor. The proposed bounds substantially reduce the sweep space while preserving the optimal candidate and maintaining end-to-end inference throughput close to the baseline. Experiments on Llama and Qwen models demonstrate that ScaleSweep consistently improves quantization performance over existing initialization methods and further narrows the gap to full precision. In particular, under aggressive end-to-end quantization of weights, activations, KV cache, and query states, ScaleSweep preserves more than 93% of the full-precision performance.

1 INTRODUCTION

Recent advances in large language models (LLMs) have substantially increased memory footprint, bandwidth demand, and computational cost during deployment. Post-training quantization (PTQ) (Krishnamoorthi, 2018) has therefore become a key approach for efficient inference, enabling model compression without retraining or full fine-tuning (Frantar et al., 2023; Xiao et al., 2023; Ashkboos et al., 2024; Liu et al., 2025; Hu et al., 2025). Among low-precision quantization schemes, NVFP4 is particularly notable for combining an FP4 E2M1 format with both FP8 micro-block scales and a tensor-level global scale, with native support on NVIDIA Blackwell GPUs (Alvarez et al., 2025). This combination reduces memory and bandwidth requirements while preserving greater numerical flexibility than integer-only formats (Chen et al., 2025; Egiazarian et al., 2026). The FP8 micro-block scaling design of NVFP4 enables practical low-bit LLM inference under aggressive compression, making scale optimization increasingly critical in fine-grained low-precision quantization.

Despite the advantages of NVFP4, existing PTQ methods exhibit different behaviors under this format. Some methods, such as GPTQ (Frantar et al., 2023) and SmoothQuant Xiao et al. (2023), remain applicable to NVFP4, whereas rotation-based approaches (Ashkboos et al., 2024; Liu et al., 2025) may degrade performance (Egiazarian et al., 2026). This difference stems from two key distinctions between NVFP4 and conventional INT4 quantization: the use of micro-block scaling and FP4 data types. Various scale initialization techniques have been proposed for INT quantization (Zhang & Shrivastava, 2025; Lin et al., 2026), but they are not directly applicable to NVFP4 due to its two-level scaling structure. Existing NVFP4 initialization methods still primarily rely on AbsMax-based heuristics, including the 4/6 strategy (Cook et al., 2026), where a noticeable gap to the optimal solution remains. These characteristics make scale optimization and error distribution in NVFP4 fundamentally different from those in both INT4 and single-level FP quantization, thereby motivating dedicated scale optimization methods for NVFP4.

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.

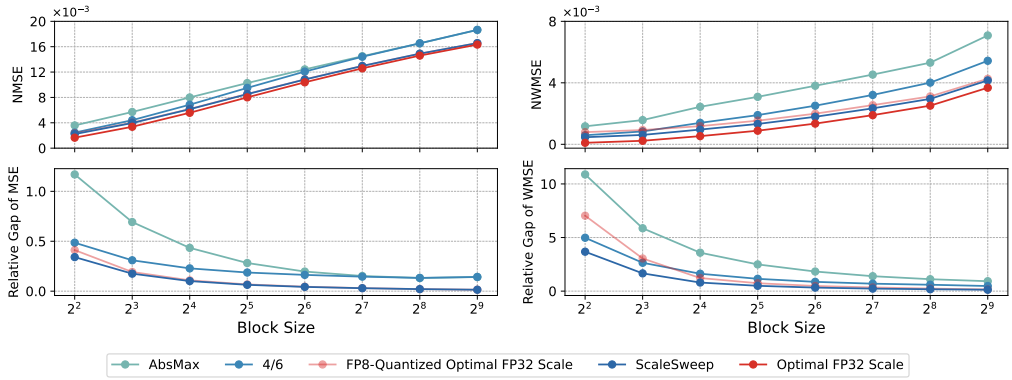


Figure 1: Normalized MSE and WMSE, together with their relative gaps to the optimal FP32 block scale, between the original tensor and the quantized reconstructed tensor under different NVFP4 block sizes using AbsMax, 4/6, ScaleSweep, and the FP8-quantized optimal FP32 scale. Definitions are provided in Section 3.

Existing NVFP4 scale initialization strategies such as AbsMax and 4/6 rely on simple heuristics based on the maximum representable FP4 values. However, as shown in Figure 1, they still exhibit a noticeable gap compared with the FP8-quantized optimal FP32 block scale[1] across different block sizes. This observation suggests that there remains substantial room for improving FP8 block scale selection. Since the number of representable FP8 scales is very limited, exhaustive scale sweep becomes computationally feasible. To this end, we propose ScaleSweep, an NVFP4-specific scale sweep method designed for FP4 quantization with FP8 block scales. For FP4 quantization, we further provide a theoretical analysis of block scale optimization under both MSE and WMSE objectives. In particular, through theoretical analysis and computer-assisted analysis, we derive theoretically justified lower and upper bounds for the optimal FP8 block scale, thereby reducing the feasible sweep range to a compact local neighborhood in the FP8 bit-pattern space and enabling efficient scale sweep. We evaluate ScaleSweep under increasingly challenging quantization settings, including weight-activation quantization, weight-activation quantization with KV-cache quantization, and weight-activation quantization with both KV-cache and query state quantization. Across all settings, ScaleSweep generally achieves stronger recovery than existing initialization methods for NVFP4.

Our main contributions are summarized below:

- We analyze FP4 quantization with FP8 block scales and derive lower bound and upper bound for the optimal block scale under MSE and WMSE objectives.
- Based on the derived bounds, we propose ScaleSweep, an NVFP4-specific calibration method that restricts FP8 block-scale optimization to a compact bit-pattern range, enabling efficient scale selection for end-to-end inference.
- We validate the effectiveness of ScaleSweep on Llama and Qwen models across weight-activation, KV-cache, and query-state quantization settings, where ScaleSweep generally outperforms existing initialization methods and recovers up to 93–95% of BF16 performance in the most aggressive setting, while introducing negligible operator overhead compared with the default NVFP4 quantization operators in vLLM.

2 Related Work

Integer quantization. Post-training quantization for integer quantization has been extensively studied for efficient large language model (LLM) inference. GPTQ Frantar et al. (2023) improves low-bit quantization through layer-wise reconstruction with approximate second-order information, while SmoothQuant (Xiao et al., 2023) mitigates activation outliers through smoothing transformations between weights and activations. More recent work further improves low-bit quantization

[1]For both the MSE and WMSE objectives, the optimal FP32 block scale can be solved exactly with low computational complexity. Details are provided in Appendix C.

by reshaping tensor distributions before quantization. QuaRot (Ashkboos et al., 2024) applies randomized Hadamard rotations to remove activation outliers and support 4-bit inference in rotated LLMs; SpinQuant (Liu et al., 2025) learns rotation transformations to better align tensors with low-bit quantization grids; and OSTQuant (Hu et al., 2025) combines orthogonal and scaling transformations to refine quantization through improved distribution fitting. Together, these methods show that reducing outliers and smoothing quantization-unfriendly distributions are central to accurate low-bit INT PTQ, where Hadamard, rotation, and orthogonal transformations have become increasingly important techniques.

FP4 quantization. FP4 quantization has recently emerged as an important direction for efficient low-precision LLM inference, particularly with the introduction of NVIDIA's NVFP4 format (Alvarez et al., 2025). Recent studies have begun to investigate FP4 quantization for both pretraining and post-training settings. NVIDIA's NVFP4 pretraining work demonstrates the feasibility of training large language models with NVFP4 precision (NVIDIA et al., 2026), while TetraJet-v2 (Chen et al., 2026) further improves NVFP4 training accuracy by addressing weight oscillation and outlier issues during low-precision training. For scale selection and refinement, 4/6 (Cook et al., 2026) extends AbsMax scaling by additionally evaluating a scale that maps the block maximum to 4 instead of 6 and selecting the one with lower quantization error. ScaleSearch (Gupta et al., 2026) only heuristically searches for better block scales, but applies the search only to inputs of Attention operator. H-Scale (Yu et al., 2026) uses a Hessian-guided formulation to refine the scale, but applies the refinement only to weight quantization. MR-GPTQ (Egiazarian et al., 2026) shows that directly applying rotation transformations such as QuaRot and SpinQuant can degrade performance under NVFP4 quantization, and instead proposes micro-rotation on top of GPTQ for hardware-supported FP4 formats. These results suggest that FP4 quantization requires techniques tailored to floating-point codebooks and microscaling structures, particularly for scale selection and format-aware reconstruction, rather than a direct reuse of INT4 quantization methods.

3 PRELIMINARY

3.1 NOTATION

Let $[N]$ denote the index set $\{0, 1, 2, \ldots, N-1\}$. For an original tensor x and its reconstructed tensor $\hat{x}$ of size n, the mean square error (MSE) and normalized mean square error (NMSE) are defined as $\mathrm{MSE}(x, \hat{x}) = \|x - \hat{x}\|_2^2/n$ and $\mathrm{NMSE}(x, \hat{x}) = \|x - \hat{x}\|_2^2/\|x\|_2^2$, respectively. Given an element-wise weight tensor w, the weighted mean square error (WMSE) and normalized weighted mean square error (NWMSE) are defined as $\mathrm{WMSE}(x, \hat{x}, w) = (\sum_i w_i(x_i - \hat{x}_i)^2)/(\sum_i w_i)$ and $\mathrm{NWMSE}(x, \hat{x}, w) = (\sum_i w_i(x_i - \hat{x}_i)^2)/(\sum_i w_i x_i^2)$, respectively.

For a numeric format $\mathcal{F}$, let $\mathcal{G}_\mathcal{F}$ denote its representable value set. In particular, $\mathcal{G}_{\mathrm{FP4}}$, $\mathcal{G}_{\mathrm{FP8}}$, and $\mathcal{G}_{\mathrm{FP32}}$ denote the representable value set of FP4 E2M1, FP8 E4M3, and FP32, respectively, where $\mathcal{G}_{\mathrm{FP4}}$ and $\mathcal{G}_{\mathrm{FP8}}$ are also referred to as quantization grids.

For $x \in \mathbb{R}$, let $\lfloor x \rceil_\mathcal{F}$ denote rounding x to the nearest value in $\mathcal{G}_\mathcal{F}$, let $\lfloor x \rfloor_\mathcal{F}$ denote rounding x downward to the largest value in $\mathcal{G}_\mathcal{F}$ not exceeding x, and let $\lceil x \rceil_\mathcal{F}$ denote rounding x upward to the smallest value in $\mathcal{G}_\mathcal{F}$ not smaller than x.

For FP4 quantization with scale s, define the scaled format $\mathrm{FP4}(s)$ as the standard FP4 format scaled by s. Its representable value set is

$$\mathcal{G}_{\mathrm{FP4}(s)} = \{s \cdot v \mid v \in \mathcal{G}_{\mathrm{FP4}}\}. \tag{1}$$

Let $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ denote non-negative weights. The weighted quantization loss under scale s is

$$\mathcal{L}(s; \mathbf{x}, \mathbf{w}) = \sum_{i=0}^{n-1} w_i \left(x_i - \lfloor x_i \rceil_{\mathrm{FP4}(s)}\right)^2. \tag{2}$$

When $\mathbf{w} = \mathbf{1}$, the objective reduces to the unweighted loss

$$\mathcal{L}(s; \mathbf{x}) = \sum_{i=0}^{n-1} \left(x_i - \lfloor x_i \rceil_{\mathrm{FP4}(s)}\right)^2. \tag{3}$$

For simplicity, both formulations are denoted by $\mathcal{L}$ when clear from context.

3.2 NVFP4 Quantization

NVFP4 combines FP4 values with two-level scaling to improve quantization fidelity under low bit-width constraints. The FP4 format used in NVFP4 follows the OCP Microscaling Formats Specification (Open Compute Project, 2023) with representable values

$$\mathcal{G}_{\mathrm{FP4}} = \{0, \pm 0.5, \pm 1, \pm 1.5, \pm 2, \pm 3, \pm 4, \pm 6\}. \tag{4}$$

Given an input tensor $\mathbf{x} \in \mathbb{R}^N$, NVFP4 represents it as $(\mathbf{q}, \mathbf{s}, S)$, where $\mathbf{q} \in \mathcal{G}_{\mathrm{FP4}}^N$ denotes FP4 values, $\mathbf{s} \in \mathcal{G}_{\mathrm{FP8}}^{N/16}$ denotes FP8 E4M3 micro-block scales shared across every 16 elements, and $S \in \mathcal{G}_{\mathrm{FP32}}$ denotes a global FP32 scale. The reconstructed tensor $\hat{\mathbf{x}}$ is computed as

$$\hat{x}_i = q_i \cdot s_{\lfloor i/16 \rfloor} \cdot S. \tag{5}$$

For NVFP4 quantization, a widely adopted initialization strategy is AbsMax (NVIDIA et al., 2026):

$$S = \frac{\max_i |x_i|}{448 \cdot 6}, \tag{6}$$

$$s_k = \left\lfloor \frac{\max_{\lfloor i/16 \rfloor = k} |x_i|}{S \cdot 6} \right\rceil_{\mathrm{FP8}}, \; k \in [N/16], \tag{7}$$

$$q_i = \left\lfloor \frac{x_i}{S \cdot s_{\lfloor i/16 \rfloor}} \right\rceil_{\mathrm{FP4}}, \; i \in [N], \tag{8}$$

where 448 and 6 are the maximum representable magnitudes of FP8 E4M3 and FP4 E2M1, respectively.

4 Method

4.1 Optimization Objective

MSE Objective For a weight tensor $W \in \mathbb{R}^{d_{\mathrm{in}} \times d_{\mathrm{out}}}$, reconstruction fidelity under NVFP4 quantization is naturally measured by the MSE objective. The resulting optimization objective is

$$\mathcal{L}_{\mathrm{NVFP4}}^{\mathrm{MSE}}(\mathbf{s}, S; W) = \sum_{i=0}^{d_{\mathrm{in}}/16-1} \sum_{j=0}^{d_{\mathrm{out}}-1} \mathcal{L}\Big(s_{i,j} \cdot S; W_{16i:16(i+1),\, j}\Big). \tag{9}$$

WMSE Objective For weight quantization, reconstruction error depends not only on the weight tensor itself, but also on the input activations propagated through the linear layer. Consider the local reconstruction error of the linear layer

$$\|XW - X\hat{W}\|_F^2 = \mathrm{tr}\Big((W - \hat{W})^T X^T X (W - \hat{W})\Big),$$

where $X \in \mathbb{R}^{T \times d_{\mathrm{in}}}$ is the input activation matrix and $\hat{W} \in \mathbb{R}^{d_{\mathrm{in}} \times d_{\mathrm{out}}}$ is the quantized weight tensor.

Directly optimizing Eq. equation 10 is impractical for quantization parameter optimization because the full Gram matrix $X^T X$ couples quantization errors across different input channels. Therefore, we adopt the diagonal approximation commonly used in model compression and quantization (LeCun et al., 1989; Guo et al., 2022; Liu et al., 2024), retaining only the diagonal entries of $X^T X$. Under this approximation, the reconstruction objective becomes

$$\sum_{i=0}^{d_{\mathrm{in}}-1} \sum_{j=0}^{d_{\mathrm{out}}-1} \mathrm{Imp}_i \left(W_{i,j} - \hat{W}_{i,j}\right)^2, \tag{10}$$

where the importance score of the i-th input channel is defined as

$$\mathrm{Imp}_i = (X_{:,i})^T X_{:,i} = \|X_{:,i}\|_2^2. \tag{11}$$

The resulting formulation naturally induces the following WMSE objective for NVFP4 quantization:

$$\mathcal{L}_{\mathrm{NVFP4}}^{\mathrm{WMSE}}(\mathbf{s}, S; W, \mathrm{Imp}) = \sum_{i=0}^{d_{\mathrm{in}}/16-1} \sum_{j=0}^{d_{\mathrm{out}}-1} \mathcal{L}\Big(s_{i,j} \cdot S; W_{16i:16(i+1),\, j}, \mathrm{Imp}_{16i:16(i+1)}\Big). \tag{12}$$

Similarly, for activation quantization in linear layers, the importance score of each input channel can be derived from the squared norm of the corresponding input channel in the associated weight matrix. For query state and KV cache quantization in attention operators, the attention output is computed as $\mathrm{softmax}(QK^T)VW^O$, where the inputs are the query states Q, key cache K, value cache V, and output projection matrix W^O. Accordingly, the importance scores for query states Q, key cache K, and value cache V are derived from the input-channel squared norms of K, Q, and W^O, respectively.

4.2 SCALESWEEP

4.2.1 GLOBAL SCALE SELECTION

As shown in Eq. equation 9 and Eq. equation 12, once the global scale is fixed, the optimization of different micro-blocks becomes independent. The two-level scaling mechanism in NVFP4 can be interpreted as a form of double quantization, where block scales are quantized into FP8 values and the global scale acts as the corresponding FP8 quantization scale. Prior analyses of FP8 quantization (Kuzmin et al., 2022; Micikevicius et al., 2022) show that the quantization error introduced by FP8 is relatively small. Therefore, it is sufficient to select a global scale that preserves adequate dynamic range for block scale optimization. Following Cook et al. (2026), we set

$$S = \frac{\max_i |x_i|}{256 \cdot 6}. \tag{13}$$

As illustrated in Figure 2, different choices of global scale produce only marginal differences in normalized MSE for both AbsMax and ScaleSweep.

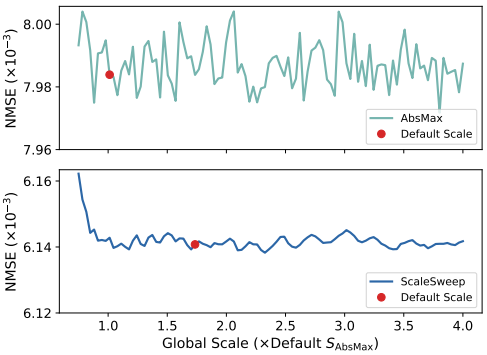


Figure 2: Normalized MSE between the original tensor and the quantized reconstructed tensor under different global scales for NVFP4, using AbsMax and ScaleSweep.

4.2.2 BLOCK SCALE OPTIMIZATION

For a single micro-block $\mathbf{x} = \{x_i\}_{i=0}^{n-1}$ with $n = 16$, once the global scale S is fixed, optimization can be equivalently performed on the normalized tensor $\mathbf{x} = \{x_i/S\}_{i=0}^{n-1}$, since the quantization loss differs only by a constant factor S^2.

Since the block scale is represented in FP8 format, the number of positive finite representable values is limited to only 126 candidates. Therefore, a natural strategy is therefore to sweep all candidate scales and select the one minimizing Eq. equation 2. However, sweeping over the entire FP8 scale space is inefficient. To improve optimization efficiency, it is desirable to derive tighter lower and upper bounds for the sweep range. To this end, define the base scale as

$$s_{\text{base}} = \frac{\max_i |x_i|}{6}, \quad s_{\text{base}}^{\text{FP8}} = \lfloor s_{\text{base}} \rfloor_{\text{FP8}}. \tag{14}$$

Block Scale Upper Bound

Lemma 4.1. *If* $s > \frac{\max_i |x_i|}{3.5}$, *then*

$$\mathcal{L}(s; \mathbf{x}, \mathbf{w}) \geq \mathcal{L}(s/2; \mathbf{x}, \mathbf{w}). \tag{15}$$

Based on Lemma 4.1, when $s > \frac{12}{7} s_{\text{base}}$, the scale s cannot achieve a lower quantization loss than $s/2$. Therefore, the upper bound of ScaleSweep is set to $\frac{12}{7} s_{\text{base}}$. Using the same technique, the corresponding upper bound for non-FP4 formats can be derived as $2\times$ the baseline scale.

Lemma 4.2. *Under the EeMm floating-point format introduced in Appendix A, for any* $1 \leq \alpha \leq 2$ *and any positive* x, *the following inequality holds:*

$$I_{\text{EeMm}}(\lfloor \alpha x \rfloor_{\text{EeMm}}) - I_{\text{EeMm}}(\lfloor x \rfloor_{\text{EeMm}}) \leq \lceil (1 - \frac{1}{\alpha}) 2^{m+1} \rceil, \tag{16}$$

where $I_{\text{EeMm}}(\cdot)$ *denotes the integer interpretation of the floating-point bit pattern of the EeMm value.*

Table 1: Bits-pattern upper bounds for $\alpha = \frac{12}{7}$ in E4M3. Here b denotes the mantissa bits of $s_{\text{base}}^{\text{FP8}}$, sup denotes the carry bit concatenated with the mantissa bits of the largest possible $\lfloor \frac{12}{7} s_{\text{base}} \rfloor_{\text{E4M3}}$, and $\Delta = \text{sup} - b$ denotes the upper bound on the bit difference relative to b.

b	000_2	001_2	010_2	011_2	100_2	101_2	110_2	111_2
sup	$0\,111_2$	$1\,000_2$	$1\,001_2$	$1\,010_2$	$1\,011_2$	$1\,100_2$	$1\,100_2$	$1\,101_2$
Δ	7	7	7	7	7	7	6	6

Table 2: Bit-pattern lower bounds for FP8 E4M3 micro-block scales when $n = 16$. Here b denotes the mantissa bits of $s_{\text{base}}^{\text{FP8}}$, inf denotes the borrow bit concatenated with the mantissa bits of the smallest possible $\lceil \frac{4}{5} s_{\text{base}}^{\text{FP8}} \rceil_{\text{E4M3}}$, and Δ denotes the upper bound on the bit difference relative to b.

b	000_2	001_2	010_2	011_2	100_2	101_2	110_2	111_2
inf	$1\,101_2$	$1\,111_2$	$0\,000_2$	$0\,001_2$	$0\,010_2$	$0\,011_2$	$0\,100_2$	$0\,100_2$
Δ	3	2	2	2	2	2	2	3

When the sweep upper bound is set to $\frac{12}{7} s_{\text{base}}$, the corresponding upper bound in the bit-pattern space becomes $I_{E4M3}\big(s_{\text{base}}^{\text{FP8}}\big) + 7$. This upper bound is obtained from Lemma 4.2 by setting $\alpha = \frac{12}{7}$, and the resulting upper bound for each mantissa field is summarized in Table 1.

Block Scale Lower Bound

Lemma 4.3. *For block size* 16*, namely* $n = 16$*, when* $\frac{11}{7} s_{\text{base}}^{\text{FP8}} \le 448$*, it follows that*

$$\arg \min_{s \in \mathcal{G}_{\text{FP8}}} \mathcal{L}(s; \mathbf{x}) \ge \frac{4}{5} s_{\text{base}}^{\text{FP8}}. \tag{17}$$

For the MSE objective, based on Lemma 4.3, the lower bound of ScaleSweep can be set to $\frac{4}{5} s_{\text{base}}^{\text{FP8}}$. Accordingly, the corresponding lower bound in the bit-pattern space is $I_{\text{E4M3}}\big(s_{\text{base}}^{\text{FP8}}\big) - 3$, and the resulting lower bound for each mantissa field is summarized in Table 2.

For the WMSE objective, the optimal scale can become arbitrarily small if the weight associated with $\max_i |x_i|$ is sufficiently small. Therefore, ScaleSweep empirically sets the lower bound to $\frac{1}{2} s_{\text{base}}$, and the corresponding lower bound in the bit-pattern space becomes $I_{\text{E4M3}}\big(s_{\text{base}}^{\text{FP8}}\big) - 8$.

Summary For the MSE objective in Eq. equation 3, the bit-pattern sweep space uses $I_{\text{E4M3}}\big(s_{\text{base}}^{\text{FP8}}\big)$ as the zero point with offset range $[-3, 7]$. For the WMSE objective in Eq. equation 2, the bit-pattern sweep space also uses $I_{\text{E4M3}}\big(s_{\text{base}}^{\text{FP8}}\big)$ as the zero point, but with offset range $[-8, 7]$. The proofs of all lemmas are provided in Appendix B.

5 Experiment

5.1 Experiment Setup

Models and metrics. We evaluate four instruction-tuned large language models: Llama-3.1-8B-Instruct, Llama-3.2-3B-Instruct (Meta AI, 2024a;b), Qwen3-8B, and Qwen3-4B (Yang et al., 2025). All Qwen3 evaluations are conducted in non-thinking mode. We report results on five benchmarks to cover different aspects of model capability: MMLU Pro for world knowledge and reasoning with 5-shot prompting (Wang et al., 2024), GSM8K for mathematical reasoning with 8-shot chain-of-thought prompting (Cobbe et al., 2021), IFEval for instruction following in the 0-shot setting (Zhou et al., 2023), and HellaSwag (HellaS) (Zellers et al., 2019) together with WinoGrande (WinoG) (Sakaguchi et al., 2021) for commonsense reasoning and language understanding in the 0-shot setting. **Avg** denotes the average score across the five benchmarks, and **Recovery (%)** denotes the performance recovery relative to the BF16 baseline.

Table 3: Comparison results of ScaleSweep and baseline initialization methods under RTN and GPTQ across different settings on Llama-3.1-8B-Instruct.

Setting	Method	Init Method	MMLU Pro	GSM8k	IFEval	HellaS	WinoG	Avg	Recovery (%)
Llama-3.1-8B-Instruct									
-	BF16	-	46.59	84.69	73.94	79.53	73.80	71.71	100
WA	RTN	AbsMax	41.48	78.32	69.69	77.91	72.53	67.99	94.81
		4/6	42.29	80.59	67.28	**78.52**	73.01	68.34	95.30
		ScaleSweep$_{\text{MSE}}$	42.77	**82.03**	**73.20**	78.17	**73.09**	**69.85**	**97.41**
		ScaleSweep	**43.09**	79.68	70.79	78.27	72.45	68.86	96.03
	GPTQ	AbsMax	41.94	**81.58**	69.13	77.94	73.32	68.78	95.92
		4/6	42.46	80.67	70.98	78.01	73.24	69.07	96.33
		ScaleSweep$_{\text{MSE}}$	**43.03**	81.43	69.13	78.19	**74.11**	69.18	96.47
		ScaleSweep	42.70	81.50	**74.49**	**78.35**	72.93	**69.99**	**97.61**
WAKV	RTN	AbsMax	38.21	76.95	68.95	77.31	71.35	66.55	92.81
		4/6	39.29	78.70	66.54	77.58	72.14	66.85	93.23
		ScaleSweep$_{\text{MSE}}$	41.06	78.85	**70.24**	78.09	**72.85**	68.22	95.13
		ScaleSweep	**41.29**	**79.91**	69.69	**78.12**	72.53	**68.31**	**95.26**
	GPTQ	AbsMax	38.40	77.94	70.61	**77.95**	71.35	67.25	93.78
		4/6	40.26	**80.06**	69.13	77.78	72.22	67.89	94.68
		ScaleSweep$_{\text{MSE}}$	40.86	79.45	70.61	77.71	72.22	68.17	95.07
		ScaleSweep	**41.26**	79.98	**71.90**	77.54	**72.77**	**68.69**	**95.80**
WAKVQ	RTN	AbsMax	34.04	69.90	**70.61**	76.99	70.88	64.48	89.93
		4/6	36.69	74.22	68.02	**77.43**	**72.14**	65.70	91.62
		ScaleSweep$_{\text{MSE}}$	37.86	74.98	69.32	77.38	71.59	66.23	92.35
		ScaleSweep	**39.72**	**77.33**	69.13	**77.43**	71.27	**66.98**	**93.40**
	GPTQ	AbsMax	34.53	73.16	68.21	76.92	71.51	64.87	90.46
		4/6	36.45	76.57	67.65	77.62	**72.14**	66.09	92.16
		ScaleSweep$_{\text{MSE}}$	38.60	**78.24**	68.58	77.42	72.06	66.98	93.41
		ScaleSweep	**39.67**	77.33	**70.24**	**77.67**	71.82	**67.35**	**93.92**

Quantization settings. We evaluate three quantization configurations: **WA** quantizes weights and activations with NVFP4, **WAKV** further quantizes the KV cache, and **WAKVQ** additionally quantizes the query states. Together, these configurations enable a progressive evaluation of increasingly comprehensive NVFP4 quantization across model components. All experiments are conducted with simulation quantization on NVIDIA L40 GPUs.

Baselines and initialization methods. We evaluate ScaleSweep under RTN and GPTQ with static activation reordering (Frantar et al., 2023) post-training quantization frameworks. Calibration uses 128 sequences of length 2048 sampled from RedPajama (Weber et al., 2024). We additionally compare MR-GPTQ with GPTQ in Appendix E.1 and find no improvement over GPTQ with activation reordering; we therefore report only GPTQ results in the main experiments. For scale initialization and optimization baselines, we compare against AbsMax and 4/6 (Cook et al., 2026). Our ScaleSweep method optimizes the WMSE objective, while ScaleSweep$_{\text{MSE}}$ is its MSE-based variant. Additional implementation details are provided in Appendix D. We further evaluate statistical robustness across different calibration seeds in Appendix E.2 and observe only small variations in performance.

5.2 Results

As shown in Table 3 and Table 11,12 in Appendix F, ScaleSweep and ScaleSweep$_{\text{MSE}}$ achieve competitive performance across different NVFP4 quantization settings and, in most cases, further narrow the gap to the BF16 baseline compared with AbsMax and 4/6 initialization. Moreover, ScaleSweep achieves further improvements over ScaleSweep$_{\text{MSE}}$ in most settings. Under the WA setting, the baseline initialization methods already achieve strong performance, while ScaleSweep still provides additional improvements for both RTN and GPTQ quantization. In particular, on Qwen3-8B, ScaleSweep further improves the recovery rate to 99.50%. Under the WAKV setting, ScaleSweep continues to provide incremental gains on top of already competitive baselines, where the recovery

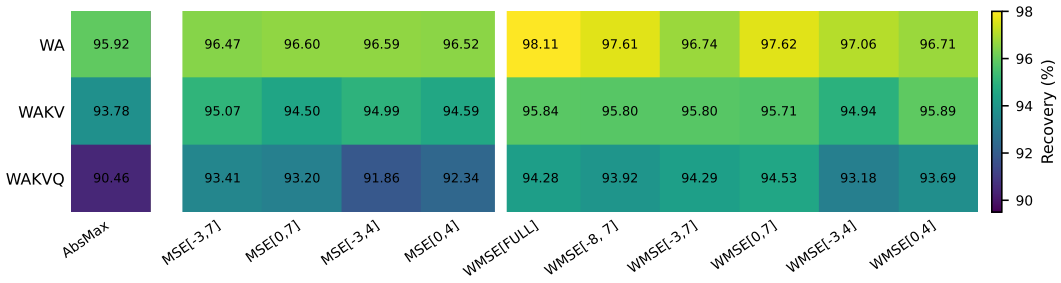


Figure 3: Recovery rate heatmaps of ScaleSweep over different bit-pattern ranges. $\mathrm{MSE}[l, r]$ and $\mathrm{WMSE}[l, r]$ denote ScaleSweep on Llama-3.1-8B-Instruct with the bit-pattern range $[l, r]$. $\mathrm{WMSE}[\mathrm{FULL}]$ denotes enumeration over all positive FP8 bit patterns.

rate is improved by around 1% to 2% across different models and quantization backends in most cases. Under the more challenging WAKVQ setting, ScaleSweep demonstrates stronger robustness as the quantization difficulty increases, while maintaining high recovery rates across different models. In particular, Llama-3.2-3B-Instruct maintains at least 93% recovery rate under both RTN and GPTQ quantization, while Qwen3-4B consistently achieves at least 94.5%. These results demonstrate that ScaleSweep remains effective under increasingly aggressive quantization settings.

Scale Sweep Range We further evaluate the impact of the sweep range in ScaleSweep under both MSE and WMSE objectives. As shown in Figure 3, narrowing the sweep range generally does not degrade performance and can occasionally yield slightly better results. Even for WMSE, expanding the range to enumerate all positive FP8 bit patterns provides marginal gains. Overall, once a moderate sweep range is included, ScaleSweep improves over AbsMax across the evaluated settings. These results also suggest that the local surrogate metric used for scale selection does not perfectly align with downstream performance, leaving further room for improving the selection criterion.

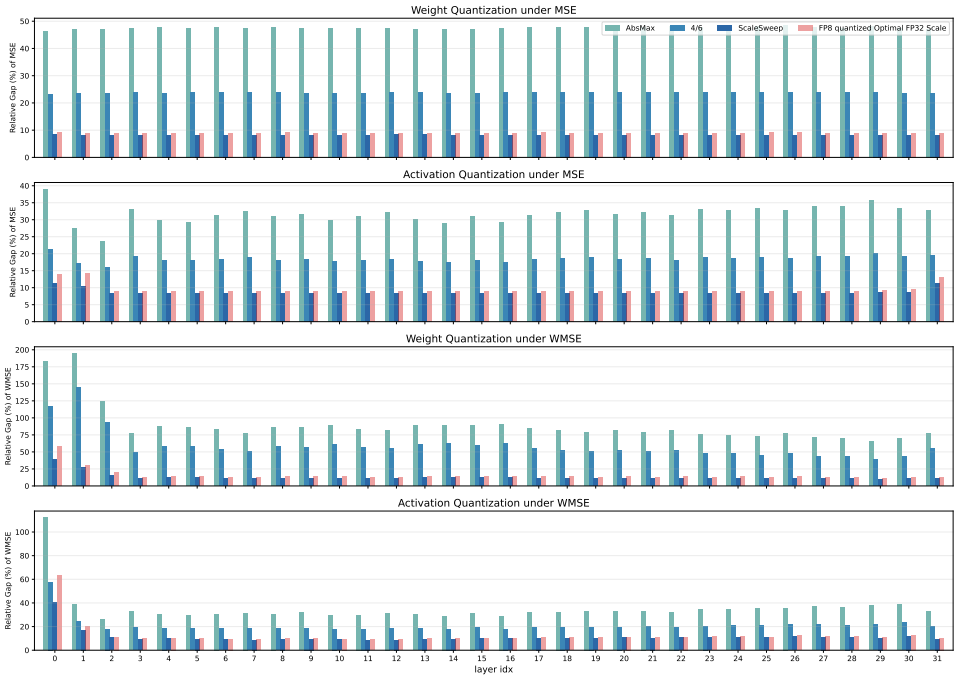


Figure 4: Performance gap (%) of ScaleSweep, initialization baselines, and FP8-quantized optimal FP32 scales relative to the optimal FP32 block scale in terms of MSE and WMSE for weight quantization and activation quantization on Llama-3.1-8B-Instruct. For both weight and activation quantization, results are averaged over all quantized tensors within each Transformer block.

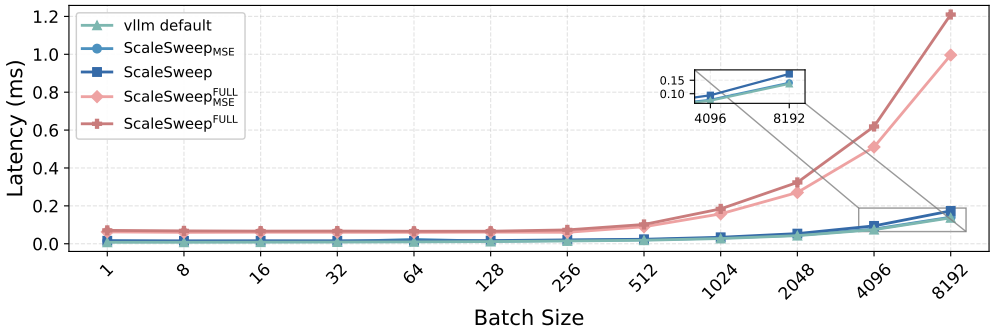


Figure 5: NVFP4 quantization operator latency comparison across different batch sizes with hidden dimension 8192 on NVIDIA RTX PRO 6000 Blackwell GPUs.

Table 4: End-to-end inference throughput of the evaluated methods. We report both total and output token throughput in tokens per second. The superscript "FULL" denotes the full-range variant, which sweeps over all positive FP8 bit patterns.

Method	vLLM Default	ScaleSweep	ScaleSweep$_{\mathrm{MSE}}$	ScaleSweep$^{\mathrm{FULL}}$	ScaleSweep$^{\mathrm{FULL}}_{\mathrm{MSE}}$
Total Tokens/s	23716.96	22267.78	23588.05	10992.82	13487.30
Output Tokens/s	2635.22	2474.20	2620.89	1221.42	1498.59

Quantization Error Analysis We evaluate the quantization error of different initialization methods against the optimal FP32 block scales across layers. As shown in Figure 4 and Figure 6 in Appendix F, ScaleSweep achieves substantial improvements over AbsMax and 4/6, and in most cases even outperforms the FP8-quantized optimal FP32 scale. For the MSE objective, ScaleSweep achieves a relative gap below 10% in nearly all cases. For the WMSE objective, the relative gap also remains below 10% in nearly all cases for weight quantization, activation quantization, and value cache quantization.

Overhead Analysis We further implement preliminary Triton kernels for ScaleSweep and ScaleSweep$_{\mathrm{MSE}}$ and compare their kernel latency with the default NVFP4 quantization operator in vLLM and their full-range variants across different batch sizes with hidden dimension 8192 on NVIDIA RTX PRO 6000 Blackwell GPUs. The results are reported in Figure 5 and Table 10 in Appendix F. ScaleSweep and ScaleSweep$_{\mathrm{MSE}}$ achieve latency close to the highly optimized vLLM operator, with ScaleSweep$_{\mathrm{MSE}}$ nearly matching the baseline and ScaleSweep introducing only moderate additional overhead. In contrast, the full-range variants incur substantially higher latency. The end-to-end throughput results in Table 4 show a similar trend: ScaleSweep and ScaleSweep$_{\mathrm{MSE}}$ retain most of the default vLLM throughput, whereas their full-range variants lead to a much larger throughput reduction. These results validate the effectiveness of restricting the sweep range for practical inference.

6 Conclusion

In this paper, we propose ScaleSweep, a scale sweep method for FP8 block scales in NVFP4 quantization that can be seamlessly integrated with existing PTQ methods such as RTN and GPTQ. To the best of our knowledge, our study is the first to derive lower and upper bounds for block scale optimization under both the MSE and WMSE objectives in NVFP4 quantization. Based on these bounds, ScaleSweep restricts the sweep range to a theoretically justified local neighborhood in the FP8 bit-pattern space. Experimental results show that ScaleSweep improves over existing initialization baselines in weight-activation quantization and under more aggressive settings involving KV cache and query state quantization. Further kernel-level latency and end-to-end throughput experiments show that ScaleSweep maintains inference efficiency close to the baseline.

AI USE STATEMENT

We used generative AI tools to improve the readability of the manuscript and to assist with implementing code based on detailed specifications, including code for figure generation. All AI-assisted text, code, and outputs were reviewed and verified by the authors.

ETHICS STATEMENT

This work focuses on improving the efficiency of large language model quantization. We use publicly available models and datasets in accordance with their respective licenses and do not involve human subjects or collect private or personally identifiable information. We are not aware of any significant ethical concerns arising specifically from this work.

REPRODUCIBILITY STATEMENT

We provide the details necessary to reproduce our experiments in Section 5.1 and Appendix D, with additional resources provided in the supplementary materials.

REFERENCES

Eduardo Alvarez, Omri Almog, Eric Chung, Simon Layton, Dusan Stosic, Ronny Krashinsky, and Kyle Aubrey. Introducing NVFP4 for efficient and accurate low-precision inference, June 2025. URL https://developer.nvidia.com/blog/introducing-nvfp4-for-efficient-and-accurate-low-precision-inference/. NVIDIA Technical Blog. Accessed: 2026-05-18.

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. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M De Sa. Quip: 2-bit quantization of large language models with guarantees. *Advances in Neural Information Processing Systems*, 36: 4396–4429, 2023.

Mengzhao Chen, Meng Wu, Hui Jin, Zhihang Yuan, Jing Liu, Chaoyi Zhang, Yunshui Li, Jie Huang, Jin Ma, Zeyue Xue, Zhiheng Liu, Xingyan Bin, and Ping Luo. Int v.s. fp: A comprehensive study of fine-grained low-bit quantization formats, 2025. URL https://arxiv.org/abs/2510.25602.

Yuxiang Chen, Yifan Liu, Xiaoming Xu, Pengle Zhang, Michael Beyer, Martin Rapp, Jun Zhu, and Jianfei Chen. Tetrajet-v2: Accurate nvfp4 training for large language models with oscillation suppression and outlier control, 2026. URL https://arxiv.org/abs/2510.27527.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Jack Cook, Junxian Guo, Guangxuan Xiao, Yujun Lin, Keith Wyss, Mahdi Nazemi, Asit Mishra, Carlo del Mundo, Tijmen Blankevoort, and Song Han. Four over six: More accurate nvfp4 quantization with adaptive block scaling, 2026. URL https://arxiv.org/abs/2512.02010.

Vage Egiazarian, Roberto L. Castro, Denis Kuznedelev, Andrei Panferov, Eldar Kurtic, Shubhra Pandit, Alexandre Noll Marques, Mark Kurtz, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Bridging the gap between promise and performance for microscaling FP4 quantization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=zCBGe9AqJZ.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers, 2023. URL https://arxiv.org/abs/2210.17323.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Cong Guo, Yuxian Qiu, Jingwen Leng, Xiaotian Gao, Chen Zhang, Yunxin Liu, Fan Yang, Yuhao Zhu, and Minyi Guo. SQuant: On-the-fly data-free quantization via diagonal hessian approximation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=JXhROKNZzOc.

Tanmaey Gupta, Hayden Prairie, Xiaoxia wu, Reyna Abhyankar, Qingyang Wu, Austin Silveria, Pragaash Ponnusamy, Jue Wang, Ben Athiwaratkun, Leon Song, Tri Dao, Daniel Y. Fu, and Chris De. Search your block floating point scales! In A. Chowdhery and Z. Jia (eds.), *Proceedings of Machine Learning and Systems*, volume 8, pp. 1383–1398. MLSys, 2026. URL https://proceedings.mlsys.org/paper_files/paper/2026/file/633b0e871a48d542280c3ad03928e60d-Paper-Conference.pdf.

Xing Hu, Yuan Cheng, Dawei Yang, Zhixuan Chen, Zukang Xu, JiangyongYu, XUCHEN, Zhihang Yuan, Zhe jiang, and Sifan Zhou. OSTQuant: Refining large language model quantization with orthogonal and scaling transformations for better distribution fitting. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=rAcgDBdKnP.

Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper, 2018. URL https://arxiv.org/abs/1806.08342.

Andrey Kuzmin, Mart Van Baalen, Yuwei Ren, Markus Nagel, Jorn Peters, and Tijmen Blankevoort. Fp8 quantization: The power of the exponent. *Advances in Neural Information Processing Systems*, 35:14651–14662, 2022.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

Li Lin, Xinyu Hu, and Xiaojun Wan. Neuqi: Near-optimal uniform quantization parameter initialization for low-bit llms, 2026. URL https://arxiv.org/abs/2505.17595.

Yifei Liu, Jicheng Wen, Yang Wang, Shengyu Ye, Li Lyna Zhang, Ting Cao, Cheng Li, and Mao Yang. Vptq: Extreme low-bit vector post-training quantization for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8181–8196, 2024.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations. In *International Conference on Learning Representations*, volume 2025, pp. 92009–92032, 2025.

Vladimir Malinovskii, Denis Mazur, Ivan Ilin, Denis Kuznedelev, Konstantin Burlachenko, Kai Yi, Dan Alistarh, and Peter Richtarik. Pv-tuning: Beyond straight-through estimation for extreme llm compression. *Advances in Neural Information Processing Systems*, 37:5074–5121, 2024.

Meta AI. Introducing Llama 3.1: Our most capable models to date, July 2024a. URL https://ai.meta.com/blog/meta-llama-3-1/. Accessed: 2026-05-18.

Meta AI. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models, September 2024b. URL https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices. Accessed: 2026-05-18.

Paulius Micikevicius, Dusan Stosic, Neil Burgess, Marius Cornea, Pradeep Dubey, Richard Grisenthwaite, Sangwon Ha, Alexander Heinecke, Patrick Judd, John Kamalu, Naveen Mellempudi, Stuart Oberman, Mohammad Shoeybi, Michael Siu, and Hao Wu. Fp8 formats for deep learning, 2022. URL https://arxiv.org/abs/2209.05433.

NVIDIA, Felix Abecassis, Anjulie Agrusa, Dong Ahn, Jonah Alben, Stefania Alborghetti, Michael Andersch, Sivakumar Arayandi, Alexis Bjorlin, Aaron Blakeman, Evan Briones, Ian Buck, Bryan Catanzaro, Muya Chang, Jinhang Choi, Mike Chrzanowski, Eric Chung, Victor Cui, Steve Dai, Bita Darvish Rouhani, Carlo del Mundo, Deena Donia, Burc Eryilmaz, Henry Estela, Abhinav Goel, Oleg Goncharov, Yugi Guvvala, Robert Hesse, Russell Hewett, Herbert Hum, Ujval Kapasi, Brucek Khailany, Mikail Khona, Nick Knight, Alex Kondratenko, Ronny Krashinsky, Ben Lanir, Simon Layton, Michael Lightstone, Daniel Lo, Paulius Micikevicius, Asit Mishra, Tim Moon, Deepak Narayanan, Chao Ni, Abhijit Paithankar, Satish Pasumarthi, Ankit Patel, Mostofa Patwary, Ashwin Poojary, Gargi Prasad, Sweta Priyadarshi, Yigong Qin, Xiaowei Ren, Oleg Rybakov, Charbel Sakr, Sanjeev Satheesh, Stas Sergienko, Pasha Shamis, Kirthi Shankar, Nishant Sharma, Mohammad Shoeybi, Michael Siu, Misha Smelyanskiy, Darko Stosic, Dusan Stosic, Bor-Yiing Su, Frank Sun, Nima Tajbakhsh, Shelby Thomas, Przemek Tredak, Evgeny Tsykunov, Gandhi Vaithilingam, Aditya Vavre, Rangharajan Venkatesan, Roger Waleffe, Qiyu Wan, Hexin Wang, Mengdi Wang, Lizzie Wei, Hao Wu, Evan Wu, Keith Wyss, Ning Xu, Jinze Xue, Charlene Yang, Yujia Zhai, Ruoxi Zhang, Jingyang Zhu, and Zhongbo Zhu. Pretraining large language models with nvfp4, 2026. URL https://arxiv.org/abs/2509.25149.

Open Compute Project. *OCP Microscaling Formats (MX) Specification*, September 2023. URL https://www.opencompute.org/documents/ocp-microscaling-formats-mx-v1-0-spec-final-pdf. Version 1.0; contributed by AMD, Arm, Intel, Meta, Microsoft, NVIDIA, and Qualcomm.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, August 2021. ISSN 0001-0782. doi: 10.1145/3474381. URL https://doi.org/10.1145/3474381.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024.

Maurice Weber, Daniel Y Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, et al. Redpajama: an open dataset for training large language models. *Advances in neural information processing systems*, 37: 116462–116492, 2024.

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 *International conference on machine learning*, pp. 38087–38099. PMLR, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Hao Yu, Zheng Li, Dayiheng Liu, and Jianwei Zhang. H-scale: Hessian-guided scale refinement for nvfp4 sub-byte llm inference, 2026. URL https://arxiv.org/abs/2608.28113.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Tianyi Zhang and Anshumali Shrivastava. Leanquant: Accurate and scalable large language model quantization with loss-error-aware grid. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ISqx8giekS.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

A Floating-Point Representation

EeMm denotes a floating-point format consisting of one sign bit, e exponent bits, and m mantissa bits. Unless otherwise specified, the exponent bias is defined as

$$B = 2^{e-1} - 1. \tag{18}$$

Only finite floating-point numbers are considered throughout this work. Therefore, the special encodings corresponding to infinity and NaN are excluded.

For a floating-point number with sign bit c, exponent field a, and mantissa field b, the represented finite value is denoted by $v(c, a, b)$ and defined as

$$v(c,a,b) = \begin{cases} (-1)^c 2^{a-B}\left(1 + \dfrac{b}{2^m}\right), & a \neq 0, \\ (-1)^c 2^{1-B}\dfrac{b}{2^m}, & a = 0, \end{cases} \tag{19}$$

where $c \in \{0, 1\}$, $a \in [2^e]$, and $b \in [2^m]$.

The set of all representable finite values in the EeMm format is denoted by

$$\mathcal{G}_{\mathrm{EeMm}} = \{v(c,a,b) \mid c \in \{0,1\},\ a \in [2^e],\ b \in [2^m]\}. \tag{20}$$

The corresponding integer interpretation of the bit pattern of the value is defined as

$$I_{\mathrm{EeMm}}(v(c,a,b)) = c \cdot 2^{e+m} + a \cdot 2^m + b. \tag{21}$$

B Proofs

B.1 Proof of Lemma 4.1

Proof. Since $s > \max_i |x_i|/3.5$, every element satisfies $|x_i| < 3.5s$. Under nearest-point FP4 quantization with scale s, no element can therefore be rounded to $\pm 4s$, whose nearest decision boundary begins at $3.5s$. Consequently, every activated quantized value under $\mathrm{FP4}(s)$ belongs to

$$\{0, \pm 0.5s, \pm s, \pm 1.5s, \pm 2s, \pm 3s\}. \tag{22}$$

This set is entirely contained in $\mathcal{G}_{\mathrm{FP4}(s/2)}$. Hence, for each x_i, the quantized value $\lfloor x_i \rceil_{\mathrm{FP4}(s)}$ is also feasible under $\mathrm{FP4}(s/2)$. By the optimality of nearest-point quantization,

$$\left| x_i - \lfloor x_i \rceil_{\mathrm{FP4}(s/2)} \right|^2 \leq \left| x_i - \lfloor x_i \rceil_{\mathrm{FP4}(s)} \right|^2. \tag{23}$$

Multiplying both sides by $w_i \geq 0$ and summing over all elements yields

$$\mathcal{L}(s/2; \mathbf{x}, \mathbf{w}) \leq \mathcal{L}(s; \mathbf{x}, \mathbf{w}). \tag{24}$$

□

B.2 Proof of Lemma 4.2

Proof. Let the exponent field have e bits and the mantissa field have m bits. Define $M = 2^m$.

For any positive $EeMm$ number, let a denote the exponent field and let b denote the mantissa field, where $0 \leq b < M$. If the number is exactly representable in $EeMm$, then b is an integer. Since the proof also considers arbitrary positive real numbers x, this notation is extended to non-representable values by allowing b to be real.

Subnormal case. Assume that x lies in the subnormal range. Then

$$I_{\mathrm{EeMm}}(\lfloor x \rfloor_{\mathrm{EeMm}}) = \lfloor b \rfloor. \tag{25}$$

Since $\alpha \leq 2$, we have $\alpha b < 2M$, and therefore

$$I_{\mathrm{EeMm}}(\lfloor \alpha x \rfloor_{\mathrm{EeMm}}) \leq \lfloor \alpha b \rfloor. \tag{26}$$

Hence,

$$I_{\mathrm{EeMm}}(\lfloor \alpha x \rfloor_{\mathrm{EeMm}}) - I_{\mathrm{EeMm}}(\lfloor x \rfloor_{\mathrm{EeMm}}) \leq \lfloor \alpha b \rfloor - \lfloor b \rfloor. \tag{27}$$

Using

$$\lfloor \alpha b \rfloor - \lfloor b \rfloor \leq \lceil (\alpha - 1) b \rceil, \tag{28}$$

together with $b < M$, we obtain

$$\lfloor \alpha b \rfloor - \lfloor b \rfloor \leq \lceil (\alpha - 1) M \rceil . \tag{29}$$

Since $1 \leq \alpha \leq 2$,

$$(\alpha - 1) M \leq \left(1 - \frac{1}{\alpha}\right) 2M. \tag{30}$$

Therefore,

$$I_{\mathrm{EeMm}}(\lfloor \alpha x \rfloor_{\mathrm{EeMm}}) - I_{\mathrm{EeMm}}(\lfloor x \rfloor_{\mathrm{EeMm}}) \leq \left\lceil \left(1 - \frac{1}{\alpha}\right) 2M \right\rceil . \tag{31}$$

Normal case below the binade boundary. Assume that x lies in the normal range. Then

$$I_{\mathrm{EeMm}}(\lfloor x \rfloor_{\mathrm{EeMm}}) = aM + \lfloor b \rfloor. \tag{32}$$

Suppose that multiplying by α does not cross into the next exponent interval:

$$\alpha(M + b) \leq 2M. \tag{33}$$

Then

$$I_{\mathrm{EeMm}}(\lfloor \alpha x \rfloor_{\mathrm{EeMm}}) \leq aM + \lfloor \alpha(M + b) - M \rfloor . \tag{34}$$

Therefore,

$$I_{\mathrm{EeMm}}(\lfloor \alpha x \rfloor_{\mathrm{EeMm}}) - I_{\mathrm{EeMm}}(\lfloor x \rfloor_{\mathrm{EeMm}}) \leq \lfloor b + (\alpha - 1)(M + b) \rfloor - \lfloor b \rfloor. \tag{35}$$

Using

$$\begin{aligned} &\lfloor b + (\alpha - 1)(M + b) \rfloor - \lfloor b \rfloor \\ &\leq \lceil (\alpha - 1)(M + b) \rceil , \end{aligned} \tag{36}$$

together with $\alpha(M + b) \leq 2M$, we obtain

$$\lfloor b + (\alpha - 1)(M + b) \rfloor - \lfloor b \rfloor \leq \left(1 - \frac{1}{\alpha}\right) 2M. \tag{37}$$

Hence,

$$I_{\mathrm{EeMm}}(\lfloor \alpha x \rfloor_{\mathrm{EeMm}}) - I_{\mathrm{EeMm}}(\lfloor x \rfloor_{\mathrm{EeMm}}) \leq \left\lceil \left(1 - \frac{1}{\alpha}\right) 2M \right\rceil . \tag{38}$$

Normal case beyond the binade boundary. Now assume $\alpha(M + b) > 2M$. At the binade boundary $\alpha(M + b) = 2M$, the desired upper bound has already been obtained in the previous case. Moreover, $\lfloor \alpha x \rfloor_{\mathrm{EeMm}}$ is exactly the first representable value on this boundary. After this point, increasing $I_{\mathrm{EeMm}}(\lfloor \alpha x \rfloor_{\mathrm{EeMm}})$ by one requires an increase of $2/\alpha$ in the mantissa field of x, while increasing $I_{\mathrm{EeMm}}(\lfloor x \rfloor_{\mathrm{EeMm}})$ by one requires an increase of 1. Since $2/\alpha \geq 1$, the difference cannot exceed its value at the boundary. Hence,

$$I_{\mathrm{EeMm}}(\lfloor \alpha x \rfloor_{\mathrm{EeMm}}) - I_{\mathrm{EeMm}}(\lfloor x \rfloor_{\mathrm{EeMm}}) \leq \left\lceil \left(1 - \frac{1}{\alpha}\right) 2M \right\rceil . \tag{39}$$

□

B.3 Proof of Lemma 4.3

Proof. The proof consists of first excluding all ratios satisfying $\leq \frac{3}{4}$, and then eliminating the remaining ratios in the interval $(\frac{3}{4}, \frac{4}{5})$ one by one.

Notation Let

$$s_0 = s_{\text{base}}^{\text{FP8}} = \left\lfloor \frac{\max_i |x_i|}{6} \right\rfloor_{\text{FP8}}. \tag{40}$$

By symmetry of the FP4 value set, it suffices to consider only nonnegative values. Define the nonnegative FP4 E2M1 grids as

$$\mathcal{G} = \left\{0, \tfrac{1}{2}, 1, \tfrac{3}{2}, 2, 3, 4, 6\right\}. \tag{41}$$

The normalized single-coordinate quantization loss is defined by

$$d_r(y) = \min_{g \in \mathcal{G}} (y - rg)^2. \tag{42}$$

Let $y_i = |x_i|/s_0$ and define the scale ratio $r = s/s_0$. Then minimizing $\mathcal{L}(s; \mathbf{x})$ over FP8 scales s is equivalent to minimizing

$$\bar{\mathcal{L}}(r; \mathbf{y}) = \sum_{i=0}^{15} d_r(y_i) \tag{43}$$

over the corresponding FP8 ratios r. Furthermore, suppose that

$$s_0 = \frac{m}{8} 2^a, \qquad m \in \{8, \ldots, 15\}. \tag{44}$$

Then the normalized coordinates satisfy

$$\max_i y_i = 6\alpha, \qquad 1 \leq \alpha < \rho_m, \tag{45}$$

where $\rho_m = \frac{m+1}{m}$.

Table 5: Convex certificates used to exclude the remaining FP8 E4M3 scale ratios in $(3/4, 4/5)$. For each candidate ratio r, the listed FP8 ratios $\{u_j\}$ and weights $\{\lambda_j\}$ satisfy $\sum_j \lambda_j = 1$. The positive margin $A_r - 15B_r$ proves that the weighted average loss over $\{u_j\}$ is strictly smaller than the loss at r for any block of size 16.

r	$\{u_j\}$	$\{\lambda_j\}$	B	A	$A + 15B$
$\frac{7}{9}$	$\{\frac{5}{6}, \frac{10}{9}, \frac{14}{9}\}$	$\{\frac{1}{4}, \frac{1}{4}, \frac{1}{2}\}$	$\frac{143}{1728}$	$-\frac{451}{324}$	$-\frac{781}{5184}$
$\frac{10}{13}$	$\{\frac{11}{13}, \frac{14}{13}, \frac{20}{13}\}$	$\{\frac{1}{4}, \frac{1}{4}, \frac{1}{2}\}$	$\frac{18}{169}$	$-\frac{277}{169}$	$-\frac{7}{169}$
$\frac{11}{14}$	$\{\frac{6}{7}, \frac{15}{14}, \frac{11}{7}\}$	$\{\frac{1}{4}, \frac{1}{4}, \frac{1}{2}\}$	$\frac{9}{98}$	$-\frac{271}{196}$	$-\frac{1}{196}$

Proof sketch. For each candidate ratio $r < \frac{5}{4}$, we find another valid ratio r' satisfying

$$r < r' \leq \tfrac{11}{4}, \tag{46}$$

and show that

$$\bar{\mathcal{L}}(r'; \mathbf{y}) \leq \bar{\mathcal{L}}(r; \mathbf{y}) \tag{47}$$

for all admissible normalized coordinates $\mathbf{y}$. The worst-case increase of

$$\bar{\mathcal{L}}(r'; \mathbf{y}) - \bar{\mathcal{L}}(r; \mathbf{y}) \tag{48}$$

is obtained by combining the largest possible contribution from the maximal coordinate with the largest possible contribution from each of the remaining 15 coordinates. Therefore, it suffices to verify

$$\begin{aligned} A_{r',r} &= \sup_{6 \leq y \leq 6\rho_m} \left(d_{r'}(y) - d_r(y)\right), \\ B_{r',r} &= \sup_{0 \leq y \leq 6\rho_m} \left(d_{r'}(y) - d_r(y)\right), \\ A_{r',r} &+ 15B_{r',r} \leq 0. \end{aligned} \tag{49}$$

Excluding $r \le \frac{3}{4}$. We first consider ratios satisfying $r \le \frac{3}{4}$, for which we choose $r' = 2r$. The grids induced by r and $2r$ share most reconstruction points, while the grid associated with r additionally contains the intermediate points $\frac{1}{2}r$ and $\frac{3}{2}r$. Therefore,

$$d_{2r}(y) - d_r(y) \le \frac{r^2}{4}, \qquad 0 \le y \le 6\rho_m, \tag{50}$$

where equality is attained at $y = \frac{1}{2}r$ and $y = \frac{3}{2}r$. Consequently,

$$B_{2r,r} = \sup_{0 \le y < 6\rho_m} (d_{2r}(y) - d_r(y)) = \frac{r^2}{4}. \tag{51}$$

Therefore, replacing r with $2r$ increases the total loss on the remaining 15 coordinates by at most $15r^2/4$.

Now consider the maximal coordinate $y_{\max} = 6\alpha$. Since $r \le \frac{3}{4}$ and $\alpha \ge 1$, the maximal reconstruction value achievable under scale r equals $6r$, which gives

$$d_r(6\alpha) = (6\alpha - 6r)^2. \tag{52}$$

Under scale $2r$, selecting the FP4 grid point 4 yields reconstruction value $8r$, and hence

$$d_{2r}(6\alpha) \le (6\alpha - 8r)^2. \tag{53}$$

Therefore,

$$d_{2r}(6\alpha) - d_r(6\alpha) \le (6\alpha - 8r)^2 - (6\alpha - 6r)^2 = -24\alpha r + 28r^2. \tag{54}$$

Using $\alpha \ge 1$ and $r \le \frac{3}{4}$ yields

$$\begin{aligned} A_{2r,r} &= \sup_{6 \le y \le 6\rho_m} (d_{2r}(y) - d_r(y)) \\ &\le -24\alpha r + 28r^2 \\ &\le -24r + 28r^2. \end{aligned} \tag{55}$$

Combining this bound with

$$B_{2r,r} = \frac{r^2}{4} \tag{56}$$

gives

$$A_{2r,r} + 15B_{2r,r} \le -24r + 28r^2 + \frac{15r^2}{4} = -r\left(24 - \frac{127}{4}r\right). \tag{57}$$

Since $r \le \frac{3}{4}$,

$$24 - \frac{127}{4}r \ge 24 - \frac{127}{4} \cdot \frac{3}{4} = \frac{3}{16} > 0. \tag{58}$$

Hence,

$$A_{2r,r} + 15B_{2r,r} < 0, \tag{59}$$

which implies

$$\bar{\mathcal{L}}(2r; \mathbf{y}) < \bar{\mathcal{L}}(r; \mathbf{y}). \tag{60}$$

Therefore, no ratio $r \le \frac{3}{4}$ can be optimal.

Excluding $r \in (3/4, 4/5)$ It remains to consider FP8 ratios in the interval $(3/4, 4/5)$. Since a positive normal E4M3 value has significand $m/8$ with $m \in \{8, \dots, 15\}$, every ratio between two such FP8 values has the form

$$r = \frac{j}{m} 2^{\Delta a}, \qquad j, m \in \{8, \dots, 15\}. \tag{61}$$

A finite enumeration shows that the only legal ratios in $(3/4, 4/5)$ are

$$\frac{7}{9}, \qquad \frac{10}{13}, \qquad \frac{11}{14}.$$

We exclude the remaining cases using convex certificates. For a fixed candidate ratio r, consider valid FP8 ratios u_j and nonnegative weights λ_j satisfying $\sum_j \lambda_j = 1$, and define

$$\Delta_{\{u_j\},r}(y) = \sum_j \lambda_j d_{u_j}(y) - d_r(y). \tag{62}$$

This construction extends the previous argument by allowing comparison against a convex combination of multiple ratios instead of a single ratio r'. Define

$$\begin{aligned} A_{\{u_j\},r} &= \sup_{6 \le y \le 6\rho_m} \Delta_{\{u_j\},r}(y), \\ B_{\{u_j\},r} &= \sup_{0 \le y \le 6\rho_m} \Delta_{\{u_j\},r}(y). \end{aligned} \tag{63}$$

If

$$A_{\{u_j\},r} + 15 B_{\{u_j\},r} \le 0, \tag{64}$$

then for any block containing a maximal coordinate,

$$\begin{aligned} &\sum_j \lambda_j \bar{\mathcal{L}}(u_j; \mathbf{y}) - \bar{\mathcal{L}}(r; \mathbf{y}) && (65) \\ &= \sum_{i=0}^{15} \Delta_{\{u_j\},r}(y_i) \\ &\le A_{\{u_j\},r} + 15 B_{\{u_j\},r} && (66) \\ &\le 0. && (67) \end{aligned}$$

Therefore, at least one ratio u_j achieves strictly smaller loss than r, implying that r cannot be optimal.

The required certificates are summarized in Table 5. All listed certificates satisfy

$$A_{\{u_j\},r} + 15 B_{\{u_j\},r} \le 0. \tag{68}$$

Since $\sum_j \lambda_j = 1$, the quadratic y^2 terms cancel in $\Delta_{\{u_j\},r}(y)$. Therefore, $\Delta_{\{u_j\},r}(y)$ is piecewise affine in y, and all extrema are attained at breakpoints. Consequently, both $A_{\{u_j\},r}$ and $B_{\{u_j\},r}$ can be computed by enumerating finitely many breakpoints. The coefficients λ_j are obtained either by exhaustive search or by solving a linear program.

Summary In summary, for every ratio satisfying $r \le \frac{4}{5}$, there exists a valid FP8 ratio $r' \le \frac{11}{4}$, or more generally a convex combination of valid FP8 ratios $\{u_j\}$ with $u_j \le \frac{11}{4}$, such that

$$\bar{\mathcal{L}}(r'; \mathbf{y}) \le \bar{\mathcal{L}}(r; \mathbf{y}), \tag{69}$$

or

$$\min_j \bar{\mathcal{L}}(u_j; \mathbf{y}) \le \sum_j \lambda_j \bar{\mathcal{L}}(u_j; \mathbf{y}) \le \bar{\mathcal{L}}(r; \mathbf{y}), \tag{70}$$

for all admissible normalized coordinates $\mathbf{y}$. Therefore, no ratio $r \le \frac{4}{5}$ can be optimal. □

Table 6: Dataset size and domain for MMLU Pro, GSM8K, IFEval, HellaSwag, and WinoGrande.

Dataset	Size	Evaluation Domain
MMLU-Pro	12,032	Broad domain knowledge and reasoning.
GSM8K	1,319	Grade school mathematical reasoning.
IFEval	541	Verifiable instruction following.
HellaSwag	10,042	Commonsense sentence completion.
WinoGrande	1,267	Winograd style commonsense reasoning.

C Optimal FP32 Scale for FP4 Quantization

Optimal FP32 Block Scale The WMSE objective in Eq. equation 2 can be equivalently written as

$$\mathcal{L}(s; \mathbf{x}, \mathbf{w}) = \sum_{i=0}^{n-1} w_i \left(x_i - \lfloor x_i/s \rceil_{\mathrm{FP4}} \cdot s\right)^2. \tag{71}$$

Each term $w_i \left(x_i - \lfloor x_i/s \rceil_{\mathrm{FP4}} \cdot s\right)^2$ constitutes a piecewise quadratic function of s, with 15 segments corresponding to the breakpoints at which x_i/s aligns with the midpoints between consecutive FP4 representable values. Consequently, $\mathcal{L}(s; \mathbf{x}, \mathbf{w})$ is a piecewise quadratic function of s, with breakpoints given by the union of all individual breakpoints, yielding at most $14n + 1$ segments. The exact global minimum of $\mathcal{L}(s; \mathbf{x}, \mathbf{w})$ can be computed efficiently by enumerating all segments, determining the quadratic coefficients within each, and evaluating the minimum of each quadratic segment.

FP8-quantized Optimal FP32 Scale The FP8-quantized optimal FP32 scale s_{fp8} is obtained by quantizing the optimal FP32 scale s with the global scale S, where s_{fp8} is selected between $\lfloor s/S \rfloor_{\mathrm{FP8}}$ and $\lceil s/S \rceil_{\mathrm{FP8}}$ according to the minimization of $\mathcal{L}(s_{\mathrm{fp8}} \cdot S; \mathbf{x}, \mathbf{w})$.

D Additional Implementation Details

D.1 Calibration Details

Calibration We follow PV-Tuning (Malinovskii et al., 2024) and sample 128 sequences of length 2048 from the 1B samples released in the official RedPajama dataset as calibration samples. Calibration is performed only once using a fixed seed of 0.

Quantization For activation quantization, KV cache quantization, and query states quantization in NVFP4, we follow the practices presented in LLM-Compressor[2] and TensorRT[3], employing a static global scale. The global scale is determined during the calibration stage and remains fixed throughout the inference stage.

GPTQ For GPTQ (Frantar et al., 2023), we employ LDLQ (Chee et al., 2023), which has been shown to be algorithmically equivalent to GPTQ. The damping factor is set to 0.01 times the mean of the Hessian diagonal. Following MR-GPTQ (Egiazarian et al., 2026), we apply static reordering: the block scales is determined prior to GPTQ, and the input channels for GPTQ quantization are ordered according to the Hessian diagonal in descending order.

D.2 Evaluation Details

For MMLU Pro, GSM8K, IFEval, HellaSwag, and WinoGrande, we use lm-evaluation-harness (Gao et al., 2024) for evaluation, corresponding to the tasks `mmlu_pro_llama`, `gsm8k_llama`, `ifeval`, `hellaswag`, and `winogrande`, respectively. Among these, `apply_chat_template` is enabled for MMLU Pro, GSM8K, and IFEval. The size and evaluation domain of the five benchmarks are summarized in Table 6.

D.3 Kernel and Throughput Benchmark Details

We develop and evaluate our implementation based on vLLM 0.29.0 on NVIDIA RTX PRO 6000 Blackwell GPUs. For kernel-level latency evaluation, we benchmark batch sizes from 1 to 8192 with hidden dimension 8192. For end-to-end throughput evaluation, we use 512 input sequences of length 2048, generate 512 output tokens per sequence, and enable vLLM's piecewise CUDA Graphs.

[2] LLM-Compressor W4A4 FP4 quantization example

[3] TensorRT dynamic double quantization guide

E Additional Evaluations

E.1 Comparison of MR-GPTQ and GPTQ

We compare MR-GPTQ with GPTQ using static activation reordering under NVFP4 quantization on Llama-3.1-8B-Instruct and Qwen3-8B. As shown in Table 7, MR-GPTQ provides no consistent improvement over GPTQ with static activation reordering. While MR-GPTQ achieves slightly better results under the WA setting, GPTQ with static activation reordering performs better under the WAKV and WAKVQ settings.

Table 7: Comparison of GPTQ with activation reordering and MR-GPTQ under NVFP4 quantization.

Setting	Init Method	Method	MMLU Pro	GSM8k	IFEval	HellaS	WinoG	Avg	Recovery (%)
Llama-3.1-8B-Instruct									
-	-	BF16	46.59	84.69	73.94	79.53	73.80	71.71	100
WA	AbsMax	GPTQ	41.94	**81.58**	69.13	**77.94**	**73.32**	68.78	95.92
		MR-GPTQ	**42.23**	79.98	**72.83**	77.60	72.14	**68.96**	**96.16**
WAKV	AbsMax	GPTQ	38.40	**77.94**	**70.61**	**77.95**	**71.35**	**67.25**	**93.78**
		MR-GPTQ	**39.18**	76.95	67.65	77.19	70.72	66.34	92.51
WAKVQ	AbsMax	GPTQ	34.53	**73.16**	68.21	**76.92**	**71.51**	**64.87**	**90.46**
		MR-GPTQ	**35.25**	70.28	**70.06**	76.26	70.01	64.37	89.77
Qwen3-8B									
-	-	BF16	61.99	91.28	81.70	74.91	68.11	75.60	100
WA	AbsMax	GPTQ	58.29	**88.78**	80.41	**73.50**	66.77	73.55	97.29
		MR-GPTQ	**58.59**	88.40	**80.96**	73.20	**67.09**	**73.65**	**97.42**
WAKV	AbsMax	GPTQ	54.90	**88.55**	**83.92**	**72.97**	**67.40**	**73.55**	**97.29**
		MR-GPTQ	**55.09**	87.95	80.41	**72.97**	67.32	72.75	96.23
WAKVQ	AbsMax	GPTQ	**51.19**	**88.32**	**82.07**	**72.58**	**67.56**	**72.34**	**95.69**
		MR-GPTQ	49.30	88.17	78.93	72.33	65.98	70.94	93.84

E.2 Statistical Robustness

To evaluate statistical robustness, we conduct experiments with five calibration seeds (0–4) and report the corresponding 95% confidence intervals. As shown in Table 8, the results remain stable across different calibration seeds.

Table 8: Results over five calibration seeds (0–4), reported as the mean and 95% confidence interval.

Model	Setting	Seed 0	Seed 1	Seed 2	Seed 3	Seed 4	95% CI
Llama-3.1-8B-Instruct	WA	69.99	69.86	69.99	70.26	69.99	70.02 ± 0.18
	WAKV	68.69	68.79	69.12	68.48	68.85	68.79 ± 0.29
	WAKVQ	67.35	67.77	67.92	67.96	68.31	67.86 ± 0.43
Qwen3-8B	WA	74.11	74.01	74.63	74.59	73.21	74.11 ± 0.71
	WAKV	72.61	73.97	72.84	73.54	73.27	73.25 ± 0.67
	WAKVQ	73.01	72.86	73.10	73.66	73.01	73.13 ± 0.38

E.3 Calibration Data

We further assess the impact of using the GSM8K training split rather than RedPajama as the calibration dataset for GPTQ with ScaleSweep. As reported in Table 9, the performance advantage on GSM8K becomes more pronounced under increasingly aggressive quantization. Across the five evaluated benchmarks, the average performance also exhibits a modest improvement, primarily

Table 9: Comparison results of ScaleSweep under different settings on Llama-3.1-8B-Instruct and Qwen3-8B using RedPajama and the GSM8K training split as calibration datasets.

			Llama-3.1-8B-Instruct		Qwen3-8B	
Setting	**Method**	**Dataset**	**GSM8K**	**Avg**	**GSM8K**	**Avg**
WA	GPTQ	RedPajama	81.50	69.99	89.31	74.11
		GSM8K	**83.47**	**70.78**	**90.60**	**74.64**
WAKV	GPTQ	RedPajama	79.98	68.69	89.31	72.61
		GSM8K	**81.88**	**68.84**	**89.39**	**73.43**
WAKVQ	GPTQ	RedPajama	77.33	67.35	88.48	73.01
		GSM8K	**80.21**	**67.79**	**89.39**	**73.78**

attributable to gains on GSM8K, while the results on the remaining four benchmarks remain largely consistent.

F Full Result

Table 10 presents the latency comparison and relative latency ratios under different batch sizes with hidden dimension 8192 on NVIDIA RTX PRO 6000 Blackwell GPUs. Figure 6 presents the MSE and WMSE gap (%) of different initialization methods relative to the optimal FP32 block scale for key cache, value cache, and query state quantization on all layers of Llama-3.1-8B-Instruct. Table 11 presents the results on Llama-3.2-3B-Instruct, while Table 12 presents the results on Qwen3-4B and Qwen3-8B.

Table 10: Comparison of latency and relative latency ratio of the NVFP4 quantization operator across different batch sizes with hidden dimension 8192 on NVIDIA RTX PRO 6000 Blackwell GPUs. "Rel. Latency" refers to relative latency ratio, which denotes the latency ratio relative to the default vLLM operator.

Method	**Metric**	1	8	16	32	64	128
vllm default	Latency(ms)	0.008	0.008	0.008	0.009	0.010	0.011
ScaleSweep$_{\text{MSE}}$	Latency(ms)	0.015	0.012	0.012	0.012	0.023	0.014
ScaleSweep$_{\text{MSE}}$	Rel. Latency	2.05×	1.50×	1.40×	1.31×	2.30×	1.19×
ScaleSweep$_{\text{MSE}}^{\text{FULL}}$	Latency(ms)	0.063	0.060	0.061	0.061	0.060	0.061
ScaleSweep$_{\text{MSE}}^{\text{FULL}}$	Rel. Latency	8.29×	7.77×	7.32×	6.80×	6.06×	5.36×
ScaleSweep	Latency(ms)	0.016	0.015	0.016	0.016	0.015	0.017
ScaleSweep	Rel. Latency	2.07×	2.00×	1.90×	1.76×	1.55×	1.45×
ScaleSweep$^{\text{FULL}}$	Latency(ms)	0.071	0.068	0.067	0.067	0.066	0.066
ScaleSweep$^{\text{FULL}}$	Rel. Latency	9.35×	8.76×	8.01×	7.52×	6.64×	5.79×
Method	**Metric**	256	512	1024	2048	4096	8192
vllm default	Latency(ms)	0.015	0.018	0.028	0.043	0.074	0.137
ScaleSweep$_{\text{MSE}}$	Latency(ms)	0.018	0.021	0.030	0.045	0.077	0.140
ScaleSweep$_{\text{MSE}}$	Rel. Latency	1.20×	1.13×	1.10×	1.05×	1.04×	1.03×
ScaleSweep$_{\text{MSE}}^{\text{FULL}}$	Latency(ms)	0.060	0.090	0.158	0.271	0.511	0.996
ScaleSweep$_{\text{MSE}}^{\text{FULL}}$	Rel. Latency	4.12×	4.85×	5.73×	6.32×	6.87×	7.28×
ScaleSweep	Latency(ms)	0.020	0.024	0.034	0.053	0.094	0.173
ScaleSweep	Rel. Latency	1.39×	1.28×	1.24×	1.25×	1.27×	1.27×
ScaleSweep$^{\text{FULL}}$	Latency(ms)	0.073	0.102	0.184	0.324	0.619	1.210
ScaleSweep$^{\text{FULL}}$	Rel. Latency	4.98×	5.51×	6.68×	7.57×	8.33×	8.85×

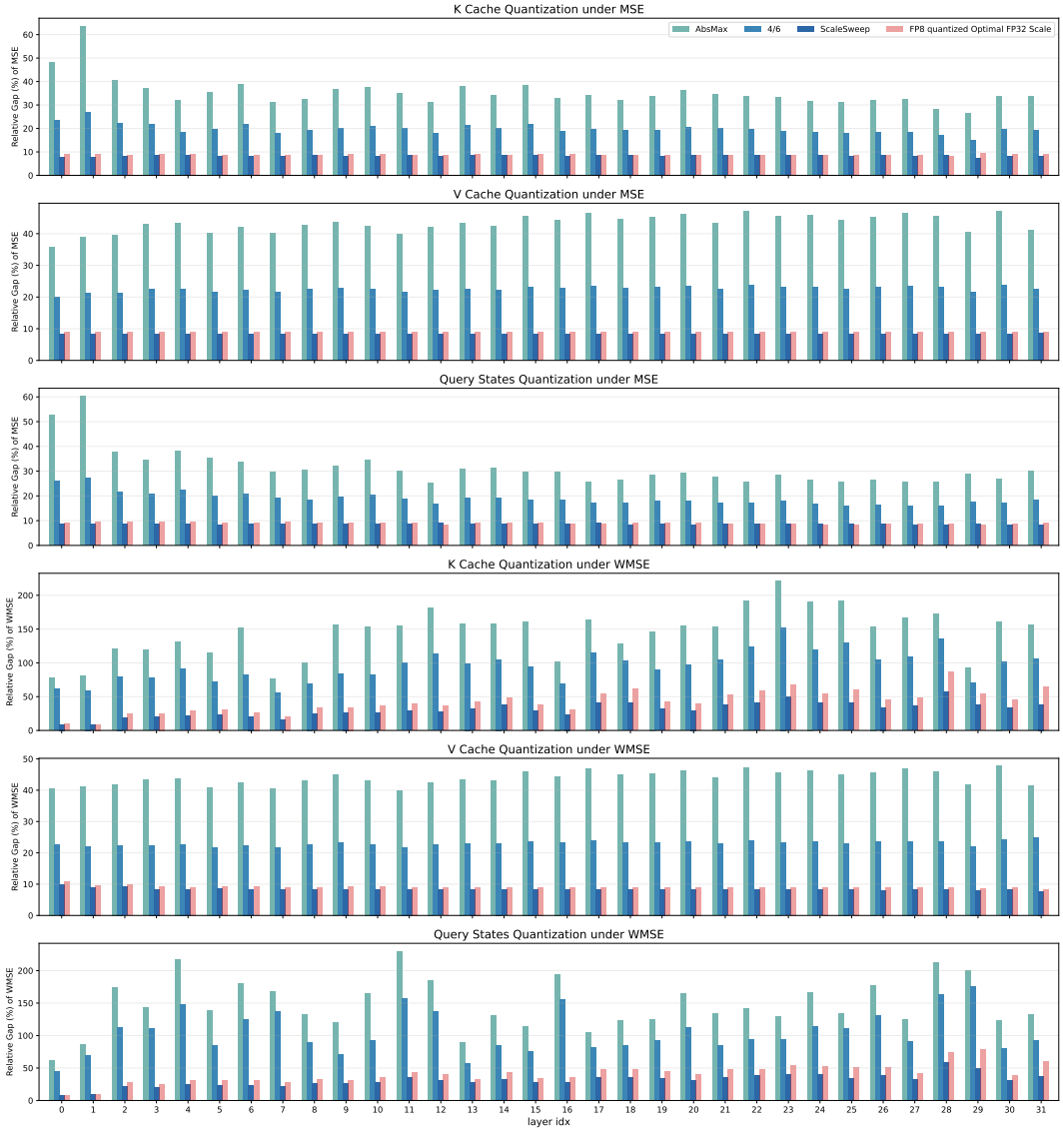


Figure 6: Performance gap (%) of ScaleSweep, initialization baselines, and FP8-quantized optimal FP32 scales relative to the optimal FP32 block scale in terms of MSE and WMSE for KV cache quantization and query states quantization on Llama-3.1-8B-Instruct.

Table 11: Comparison results of ScaleSweep and baseline initialization methods under RTN and GPTQ across different settings on Llama-3.2-3B-Instruct.

Setting	Method	Init Method	MMLU Pro	GSM8k	IFEval	HellaS	WinoG	Avg	Recovery (%)
Llama-3.2-3B-Instruct									
-	BF16	-	37.30	78.54	70.24	71.75	68.51	65.27	100
WA	RTN	AbsMax	33.43	71.49	63.96	70.30	68.27	61.49	94.21
		4/6	34.10	71.95	67.28	70.11	65.51	61.79	94.67
		$\text{ScaleSweep}_{\text{MSE}}$	33.96	**73.46**	68.02	**70.39**	67.09	62.59	95.89
		ScaleSweep	**34.12**	73.16	**68.76**	70.18	**68.98**	**63.04**	**96.58**
	GPTQ	AbsMax	33.29	71.57	65.43	69.85	**68.75**	61.78	94.65
		4/6	34.23	72.78	69.69	**70.53**	66.06	62.66	96.00
		$\text{ScaleSweep}_{\text{MSE}}$	**35.09**	**74.68**	65.62	70.26	**68.75**	62.88	96.34
		ScaleSweep	34.97	73.69	**70.61**	70.37	66.46	**63.22**	**96.86**
WAKV	RTN	AbsMax	30.93	67.17	66.36	69.74	**68.03**	60.45	92.61
		4/6	31.99	69.83	**67.84**	69.63	66.69	61.19	93.76
		$\text{ScaleSweep}_{\text{MSE}}$	33.08	**72.33**	67.47	69.79	66.30	61.79	94.67
		ScaleSweep	**33.10**	**72.33**	**67.84**	**70.08**	67.80	**62.23**	**95.34**
	GPTQ	AbsMax	31.26	67.63	66.73	69.81	66.85	60.45	92.62
		4/6	32.41	71.11	66.17	69.50	**67.40**	61.32	93.95
		$\text{ScaleSweep}_{\text{MSE}}$	33.44	**72.33**	65.62	**70.22**	66.93	61.71	94.54
		ScaleSweep	**33.75**	71.80	**67.65**	70.07	67.25	**62.10**	**95.15**
WAKVQ	RTN	AbsMax	28.00	61.87	63.96	68.71	65.43	57.59	88.24
		4/6	30.44	67.32	60.44	69.46	**67.32**	59.00	90.39
		$\text{ScaleSweep}_{\text{MSE}}$	31.33	68.31	66.17	69.13	66.69	60.33	92.43
		ScaleSweep	**31.93**	**69.90**	**66.91**	**69.73**	66.38	**60.97**	**93.41**
	GPTQ	AbsMax	29.81	62.62	61.74	69.05	65.35	57.71	88.43
		4/6	30.73	66.11	65.80	69.01	64.80	59.29	90.84
		$\text{ScaleSweep}_{\text{MSE}}$	32.21	68.76	62.66	69.25	65.82	59.74	91.53
		ScaleSweep	**33.02**	**71.11**	**65.99**	**69.43**	**66.30**	**61.17**	**93.72**

Table 12: Comparison results of ScaleSweep and baseline initialization methods under RTN and GPTQ across different settings on Qwen3-8B and Qwen3-4B in the non-thinking mode.

Setting	Method	Init Method	MMLU Pro	GSM8k	IFEval	HellaS	WinoG	Avg	Recovery (%)
Qwen3-8B									
-	BF16	-	61.99	91.28	81.70	74.91	68.11	75.60	100
WA	RTN	AbsMax	58.78	89.31	81.33	73.20	68.19	74.16	98.10
		4/6	59.38	89.61	81.15	73.37	68.03	74.31	98.30
		ScaleSweep$_{\text{MSE}}$	**59.67**	89.23	82.62	73.66	67.64	74.57	98.63
		ScaleSweep	59.57	**90.37**	**83.18**	**73.68**	**69.30**	**75.22**	**99.50**
	GPTQ	AbsMax	**58.29**	88.78	80.41	73.50	66.77	73.55	97.29
		4/6	52.24	89.23	81.52	73.31	67.25	72.71	96.18
		ScaleSweep$_{\text{MSE}}$	56.91	**89.99**	**82.44**	**73.97**	**67.96**	**74.25**	**98.22**
		ScaleSweep	57.95	89.31	82.07	73.41	67.80	74.11	98.03
WAKV	RTN	AbsMax	56.48	86.81	80.59	72.34	**68.59**	72.96	96.51
		4/6	56.09	89.46	80.59	73.54	66.14	73.17	96.78
		ScaleSweep$_{\text{MSE}}$	57.50	89.16	81.70	73.50	65.98	73.57	97.31
		ScaleSweep	**58.05**	**89.61**	**83.73**	**73.79**	65.90	**74.22**	**98.17**
	GPTQ	AbsMax	54.90	88.55	**83.92**	72.97	**67.40**	**73.55**	**97.29**
		4/6	47.91	88.55	81.33	73.04	67.25	71.62	94.73
		ScaleSweep$_{\text{MSE}}$	**55.49**	88.40	79.48	73.56	66.30	72.65	96.09
		ScaleSweep	51.95	**89.31**	80.96	**73.88**	66.93	72.61	96.04
WAKVQ	RTN	AbsMax	52.98	87.11	**82.62**	71.83	**67.48**	72.40	95.77
		4/6	53.71	87.64	80.41	**72.92**	66.46	72.23	95.54
		ScaleSweep$_{\text{MSE}}$	56.81	88.70	81.70	72.84	66.69	**73.35**	**97.02**
		ScaleSweep	**57.34**	**88.86**	80.96	72.66	65.98	73.16	96.77
	GPTQ	AbsMax	51.19	88.32	**82.07**	72.58	**67.56**	72.34	95.69
		4/6	48.09	88.25	80.59	72.78	67.17	71.38	94.41
		ScaleSweep$_{\text{MSE}}$	53.55	**88.48**	80.22	73.18	65.98	72.28	95.61
		ScaleSweep	**56.08**	**88.48**	80.59	**73.20**	66.69	**73.01**	**96.57**
Qwen3-4B									
-	BF16	-	57.86	88.25	79.85	68.32	66.06	72.07	100
WA	RTN	AbsMax	52.86	86.58	**79.67**	65.77	62.51	69.48	96.40
		4/6	54.21	86.73	78.56	66.02	**63.85**	**69.87**	**96.95**
		ScaleSweep$_{\text{MSE}}$	53.37	86.05	78.93	**66.67**	63.30	69.66	96.66
		ScaleSweep	**54.69**	**88.02**	76.52	66.61	61.25	69.42	96.32
	GPTQ	AbsMax	53.74	85.82	77.82	66.48	64.09	69.59	96.56
		4/6	54.16	86.88	77.82	66.37	65.43	70.13	97.31
		ScaleSweep$_{\text{MSE}}$	54.75	86.28	**79.11**	66.34	**65.51**	**70.40**	**97.68**
		ScaleSweep	**54.95**	**87.04**	77.26	**66.98**	62.98	69.84	96.91
WAKV	RTN	AbsMax	49.33	84.23	76.52	65.12	62.43	67.53	93.70
		4/6	51.38	83.78	**79.85**	65.46	61.72	68.44	94.96
		ScaleSweep$_{\text{MSE}}$	51.02	85.37	78.19	65.64	62.12	68.47	95.00
		ScaleSweep	**52.79**	**87.11**	77.45	**66.32**	**62.51**	**69.24**	**96.07**
	GPTQ	AbsMax	51.18	84.31	74.68	65.52	64.56	68.05	94.42
		4/6	51.67	85.06	**80.22**	65.68	62.27	68.98	95.72
		ScaleSweep$_{\text{MSE}}$	52.35	84.99	79.48	65.65	63.30	69.16	95.96
		ScaleSweep	**53.25**	**86.35**	78.19	**66.73**	**65.04**	**69.91**	**97.01**
WAKVQ	RTN	AbsMax	47.44	82.41	76.34	64.38	61.17	66.35	92.06
		4/6	46.88	83.02	77.45	65.21	**63.06**	67.12	93.14
		ScaleSweep$_{\text{MSE}}$	49.29	**84.08**	77.63	65.07	62.04	67.62	93.83
		ScaleSweep	**51.70**	84.00	**78.37**	**65.76**	60.85	**68.14**	**94.54**
	GPTQ	AbsMax	46.96	81.12	77.82	65.24	61.48	66.52	92.30
		4/6	49.17	81.88	76.71	65.23	62.51	67.10	93.10
		ScaleSweep$_{\text{MSE}}$	50.87	**84.76**	**78.93**	65.46	64.88	**68.98**	**95.71**
		ScaleSweep	**51.53**	83.70	76.34	**65.91**	**64.96**	68.49	95.03