Half-S: Halving the Scale for Near-Lossless 4-Bit LLM Training

Jinyang Du[1], Ruihao Gong[1]*, Linghan Ai[1], Zining Wang[1], Yunke Peng[2], Yao Wang[2], Lei Yan[2], Xuefei Wang[2], Yaoyuan Wang[2], Jinyang Guo[1], Dahua Lin[3,4], Xianglong Liu[1]

[1]Beihang University, China
[2]Huawei Technologies Ltd., China
[3]SenseTime Research, China
[4]The Chinese University of Hong Kong, Hong Kong SAR, China

{jinyangdu, gongruihao, wangzining, jinyangguo, xlliu}@buaa.edu.cn, fgailinghan@gmail.com
{pengyunke, wangxuefei10, wangyao123, ray.yanlei, wangyaoyuan1}@huawei.com, dhlin@ie.cuhk.edu.hk

Abstract

Training large language models (LLMs) at 4-bit precision offers substantial efficiency gains but remains challenging due to the limited dynamic range and coarse numerical resolution. Existing 4-bit training pipelines typically rely on max-scaling, which is ill-suited for heavy-tailed LLM tensor distributions and leads to severe under-utilization of the FP4 quantization grid in the low-magnitude region. This effect causes pronounced *representation collapse* and large rounding errors for the values that dominate LLM computation. In this work, we derive the theoretically optimal scaling for FP4 under heavy-tailed inputs, revealing why max-scaling is intrinsically suboptimal. Guided by this analysis, we propose **Half-S**, a simple and efficient scaling strategy that uses half-scaling as a hardware-friendly default and falls back to an MSE-based clipping threshold when needed, yielding a close approximation to the theoretical optimum under real LLM statistics. Extensive experiments on large-scale pretraining and downstream fine-tuning show that Half-S consistently narrows the gap to BF16 in both convergence and final model quality, while preserving the efficiency benefits of 4-bit computation. Under native FP4 support, Half-S is estimated to provide up to **1.8**$\times$ end-to-end training speedup. These results indicate that Half-S provides a simple and effective correction to max-scaling, substantially improving the stability and accuracy of 4-bit LLM training.

1 Introduction

Large Language Models (LLMs) have achieved transformative success across a wide range of applications, from natural language understanding to complex reasoning (Vaswani et al., 2017; Brown et al., 2020; Chowdhery et al., 2022; Wei et al., 2022; Touvron et al., 2023; OpenAI, 2023). However, the rapid scaling of these models imposes immense computational and memory demands during training. To mitigate these costs, modern accelerators have introduced native support for low-precision arithmetic, including 8-bit and even 4-bit integer or floating-point instructions (NVIDIA, 2022; Advanced Micro Devices (AMD), 2023). While 8-bit floating-point (FP8) training has been successfully validated by industry deployments such as DeepSeek (DeepSeek-AI, 2024), stable and near-lossless 4-bit training remains a significant challenge due to the severely constrained dynamic range and numerical resolution.

Recent efforts to improve 4-bit training have explored more fine-grained scaling strategies to map real-valued tensors into the FP4 range, such as block-wise scaling adopted in MXFP4. However, these methods still compute the scale from the absolute block maximum, i.e., max-scaling, which prioritizes avoiding clipping but inflates the scale and induces large rounding errors for most values. Some subsequent works modify the treatment of large values by rescaling the upper representable range (e.g., $[4, 6]$) (Cook et al., 2025), yet they remain focused on large magnitudes and neglect the small- and medium-magnitude values in $[0, 4]$ that dominate LLM tensors, leading to a substantial loss gap in 4-bit training.

In this work, we first identify a fundamental limitation of max-scaling in 4-bit training: under heavy-tailed LLM tensor distributions, max-scaling severely under-utilizes the FP4 quantization grid in the low-magnitude region, leading to pronounced *representation collapse*. In practice, although FP4 provides 15 non-zero representable values, max-scaling often allows only a small subset of low-magnitude levels to represent the ma-

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 4890–4903
July 2-7, 2026

jority of values in a block (e.g., 6 levels), causing many distinct inputs to collapse onto identical representations and resulting in large rounding error.

Motivated by this observation, we propose **Half-S**, a simple yet effective scaling correction that halves the max-scaling factor to alleviate representation collapse in the low-magnitude region. We provide a theoretical analysis that characterizes the error-optimal clipping threshold for heavy-tailed inputs under the non-uniform FP4 quantization grid, showing that the scale implied by the empirical maximum is systematically inflated by outliers. We then bridge theory and practice by demonstrating that halving the max-based scale closely matches the theoretical optimum in real LLM tensors, while admitting an extremely efficient implementation via a single exponent shift and safely reducing to Max-scaling when Half-S is not theoretically appropriate.

We evaluate Half-S through large-scale pretraining on OLMo-7B with 20B tokens and downstream fine-tuning on Llama-2-7B. Experimental results show that Half-S matches BF16 convergence and final model quality while preserving the full performance benefits of 4-bit arithmetic, achieving up to an estimated **1.8**× end-to-end throughput speedup. Together, these results demonstrate that Half-S constitutes a minimal yet fundamental correction for enabling practical, stable, and near-lossless 4-bit LLM training on modern hardware. Our contributions are summarized as follows:

- We identify *representation collapse* in the low-magnitude region as a key problem of max-scaling, leading to large rounding errors.
- A theoretically optimal scaling is derived for FP4 under heavy-tailed inputs, revealing why max-scaling is intrinsically suboptimal and providing guidance for improved strategies.
- We propose **Half-S**, an efficient scaling strategy that bridges theory and practice by appropriately halving the max-based scale, achieving near-optimal scaling theoretically.
- Extensive pretraining and fine-tuning experiments demonstrate that Half-S substantially narrows the gap to BF16 convergence under MXFP4 while delivering an estimated **1.8**× speedup under native FP4 support.

2 Related Work

2.1 Low-precision Training

To circumvent memory and communication bottlenecks, training paradigms have progressively shifted from 32-bit precision to 16-bit formats like FP16 (Micikevicius et al., 2017) and BF16 (Kalamkar et al., 2019; Shoeybi et al., 2019; Rasley et al., 2020), and more recently to 8-bit floating point (FP8) workflows (Micikevicius et al., 2022; Dettmers et al., 2022). While early FP8 systems like Transformer Engine (NVIDIA Corporation, 2025) focused on linear layers, subsequent efforts such as FP8-LM (Peng et al., 2023) and ZeroQuant (Yao et al., 2022) extended quantization to activations and gradients. To further maximize efficiency, recent systems including COAT (Xi et al., 2024) and DeepSeek-V3 (Liu et al., 2024) introduced fine-grained quantization for activations and optimizer states, a direction also explored by 8-bit optimizers (Dettmers et al., 2021) and innovations like QLoRA (Dettmers et al., 2023), which enables 4-bit fine-tuning. However, as noted by MOSS (Zhang et al., 2025), excessive granularity can incur prohibitive dequantization overheads, necessitating designs that balance compression rates with kernel efficiency.

2.2 Block-scaled Quantization

Quantization has evolved from coarse tensor-wise scaling to fine-grained Block-scaled Quantization (Microscaling) to maintain precision at lower bit-widths. The OCP Microscaling standard (Rouhani et al., 2023) formalized this by partitioning tensors into blocks (e.g., $k = 32$) sharing a hardware-efficient E8M0 exponent. This structure decouples dynamic range from local precision. Building on this, NVFP4 further reduced block sizes ($k = 16$) to facilitate direct 4-bit training on next-generation architectures (Abecassis et al., 2025), establishing the foundation for ultra-low precision scaling.

2.3 Scaling Strategies

The standard **Max-scaling** strategy (Rouhani et al., 2023) specifically sets the clipping threshold to the block's absolute maximum to strictly prevent overflow. However, aligning the scale with extreme outliers stretches the quantization grid, which causes severe resolution loss for the dense central peak. While adaptive methods like "Four Over Six" (Cook et al., 2025) attempt to optimize

thresholds, they still focus on large values and ignore the large error in the low-magnitude region.

3 Preliminary

FP4 and MXFP4: FP4 is a low-precision floating-point format with a non-uniform set of representable magnitudes. For the E2M1 layout, the positive representable magnitudes form a non-uniform grid $\mathcal{Q}_+ = \{0, 0.5, 1, 1.5, 2, 3, 4, 6\}$, with sign symmetry for negative values. Due to the limited intrinsic dynamic range of FP4, real-valued tensors must be scaled before quantization, which can be expressed as:

$$\hat{\mathbf{X}} = S \cdot Q\left(\frac{\mathbf{X}}{S}\right), \tag{1}$$

where $\mathbf{X}$ denotes the original tensor, S is the *scaling factor*, and $Q(\cdot)$ denotes rounding to the nearest FP4 representable value. For E2M1, the quantization function $Q(\cdot)$ is given by:

$$Q(x) = \begin{cases} \frac{1}{2}\lceil 2x \rfloor, & |x| < 2, \\ \lceil x \rfloor, & 2 \le |x| < 4, \\ 2\lceil \frac{x}{2} \rfloor, & 4 \le |x| \le 6, \end{cases} \tag{2}$$

where $\lceil \cdot \rfloor$ denotes rounding to the nearest integer. This piecewise definition reflects the non-uniform spacing of the FP4 grid, with denser levels near zero and sparser levels at larger magnitudes.

MXFP4 is a practical FP4 variant that applies block-wise scaling. A block of n values (e.g., $n = 32$) shares a common scaling factor S, which is stored in E8M0 format and therefore restricted to powers of two. As a result, the scale is selected per block as the largest power-of-two value that accommodates the block maximum.

Max-scaling: A commonly used choice for the scaling factor is *max-scaling*, which aligns the largest magnitude in the tensor with the maximum representable FP4 value:

$$S_{\max} = \frac{\max(|\mathbf{X}|)}{V_{\max}}. \tag{3}$$

In MXFP4, this value is further quantized to the nearest power of two to satisfy the E8M0 constraint. Max-scaling strategy strictly avoids overflow and is widely adopted due to its simplicity. While computationally efficient, this greedy strategy stretches the quantization grid to accommodate rare outliers, significantly inflating rounding noise for the dense majority of values near zero.

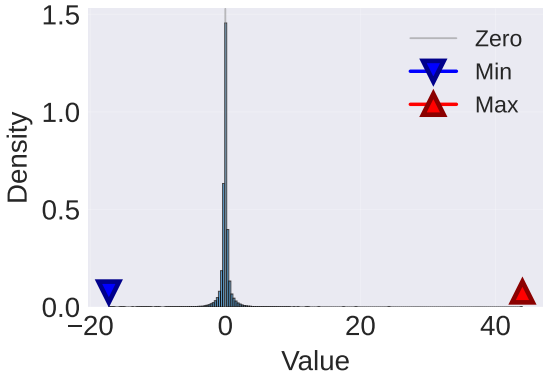


Figure 1: Tensors in LLMs exhibit heavy-tailed value distributions. The majority of values concentrate near zero, while rare outliers extend far into the tails.

4 Methodology

In this section, we show that max-scaling causes severe *representation collapse*, and introduce **Half-S**, a theory-guided half-scaling correction with safe reduction to Max-scaling.

4.1 Motivation

We show that the severe *representation collapse* of max-scaling in FP4 is induced by quantization grid under-utilization, mapping many distinct values onto a small set of representations and degrading training dynamics.

4.1.1 Heavy-tailed LLM Tensors

Tensors in modern large language models exhibit pronounced heavy-tailed distributions, where most values concentrate near zero while rare outliers extend far into the tails, as shown in Figure 1. Although the majority of elements lie in a narrow central range, these rare large-magnitude values dominate the maximum statistics.

4.1.2 Representation Collapse of Max-scaling

Under heavy-tailed distributions, quantization grid under-utilization directly leads to representation collapse. We analyze this phenomenon at both the single-block and cross-block levels.

Single block analysis. We first illustrate this effect at the block level using Figure 2. In this example, a block contains 32 values, among which 30 values lie within half of the maximum magnitude. However, under Max-scaling, these values can only be represented by a limited subset of the MXFP4 quantization grid. Specifically, although MXFP4 provides 15 non-zero representable levels, only 6 low-magnitude levels are available to

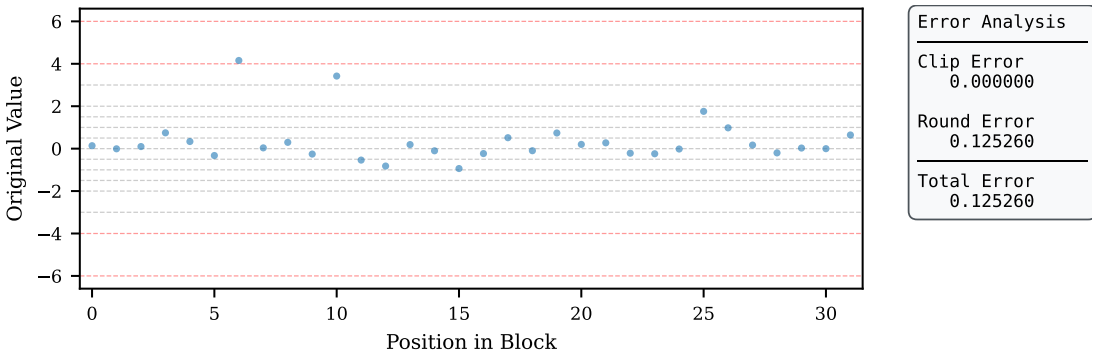


Figure 2: Quantization grid under-utilization and representation collapse at the block level under Max-scaling. In this block of 32 values, 30 values lie within half of the maximum magnitude, yet they can only be represented by 6 available MXFP4 quantization levels. As a result, many distinct inputs collapse onto identical representations, leaving much of the quantization grid unused and leading to large rounding error.

represent the majority of values in this block. As a result, many distinct inputs are forced to collapse onto identical discrete representations, leaving much of the quantization grid unused. This severe under-utilization leads to substantial rounding distortion, yielding a block-wise mean squared error (MSE) of 0.50 despite the absence of clipping.

Cross blocks analysis. The representation collapse problem is not an isolated case for block level, but persists across the model at scale. As shown in Figure 3, after applying Max-scaling, 93.12% of quantized values across multiple blocks fall within the low-magnitude interval $[0, 4]$. Yet, due to representation collapse, a substantial fraction of these values are mapped to zero or to a small number of identical low-magnitude levels. This widespread collapse sharply reduces the effective number of distinct responses during training, diminishing signal diversity and leading to degraded optimization dynamics.

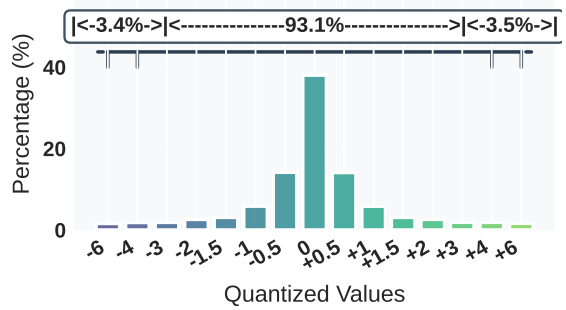


Figure 3: Representation collapse under Max-scaling is pervasive across blocks. We have statistics over 10k tensors and over 93% of values lie within the low-magnitude range after Max-scaling, and many of them collapse to zero or to a small set of identical levels, indicating a widespread loss of representational diversity.

Together, Figure 2 and Figure 3 show that Max-scaling severely under-utilizes the quantization grid, causing representation collapse and large rounding error in the low-magnitude region that dominates LLM computation, which directly motivates the need for a corrected scaling strategy.

4.2 Efficient Half-Scaling Framework

Motivated by the representation collapse induced by Max-scaling, We reformulate scale selection via an explicit clipping threshold α and derive a hardware-efficient correction termed **Half-S**.

4.2.1 Theoretical Basis

Max-scaling sets the shared scale using the observed maximum $\max(|\mathbf{X}|)$, which is dominated by rare outliers under heavy-tailed LLM tensors. To analyze the MSE-optimal scaling from first principles, we instead parameterize the scale by an explicit clipping threshold α:

$$S = \frac{\alpha}{V_{\max}}, \qquad V_{\max} = 6, \tag{4}$$

where $V_{\max}$ is the maximum value of FP4 (E2M1).

Objective. With the above parameterization, the total error decomposes into clipping and rounding:

$$\mathrm{MSE}(\alpha) = E_c(\alpha) + E_r(\alpha), \tag{5}$$

where $E_c(\alpha) = \mathbb{E}[(X - X_c)^2]$ is the clipping error with $X_c = \mathrm{clip}(X, -\alpha, \alpha)$, and $E_r(\alpha) = \mathbb{E}[(X_c - Q(X, S))^2]$ is the rounding error within the clipping range. This decomposition holds because the two error terms have disjoint supports.

Part I: Clipping error. Assume a zero-mean Laplace input $X \sim \text{Laplace}(0, b)$ (evidence in Appendix A) with density $p(x) = \frac{1}{2b}\exp(-|x|/b)$. For symmetric clipping to $[-\alpha, \alpha]$, the clipping MSE admits a closed form:

$$E_c(\alpha) = \mathbb{E}\big[(|X| - \alpha)_+^2\big] = 2b^2 \exp\Big(-\frac{\alpha}{b}\Big). \quad (6)$$

Part II: Rounding error. As introduced in Section 3, FP4 (E2M1) employs a non-uniform quantization grid with positive representable magnitudes $\mathcal{Q}_+ = \{0, 0.5, 1, 1.5, 2, 3, 4, 6\}$. Let $\{q_i\}_{i=0}^7$ denote the ordered elements of this grid. The corresponding decision boundaries are defined as $m_0 = 0$, $m_{i+1} = \frac{q_i + q_{i+1}}{2}$ for $i = 0, \ldots, 6$, and $m_8 = 6$. Within the clipping range, the rounding mean squared error (MSE) can be expressed as a piecewise integral over the FP4 quantization bins:

$$E_r(\alpha) = \sum_{i=0}^{7} \int_{m_i S}^{m_{i+1} S} (x - q_i S)^2 \, \frac{1}{b} \exp\Big(-\frac{x}{b}\Big) \, dx, \quad (7)$$

which explicitly captures the effect of non-uniform FP4 grid geometry on rounding distortion.

Optimal solution. Based on the MSE objective Equation 5 derived above, the optimal clipping threshold is obtained by solving the first-order condition $\frac{d}{d\alpha}\text{MSE}(\alpha) = 0$. We normalize by the Laplace scale b and define $\hat{\alpha} = \alpha/b$ and $\hat{S} = S/b = \hat{\alpha}/6$. Let $\Phi(\hat{\alpha}) \triangleq E_r(\alpha)/b^2$ denote the normalized rounding error, which depends on α only through $\hat{\alpha}$. The normalized objective becomes:

$$\frac{\text{MSE}(\alpha)}{b^2} = 2e^{-\hat{\alpha}} + \Phi(\hat{\alpha}), \quad (8)$$

yielding the optimality condition:

$$\Phi'(\hat{\alpha}^*) = 2e^{-\hat{\alpha}^*}. \quad (9)$$

Solving this one-dimensional equation numerically gives $\hat{\alpha}^* \approx 5.86453$ (more details in Appendix B), hence $\alpha^* \approx 5.86b$. For a Laplace distribution, $\sigma = \sqrt{2}b$, leading to:

$$\alpha^* \approx \frac{5.86}{\sqrt{2}}\sigma \approx 4.14\sigma. \quad (10)$$

4.2.2 Half-S: Bridging Theory and Practice

Given the optimal threshold scales with σ, we next connect this result to empirical LLM statistics and derive a hardware-friendly solution.

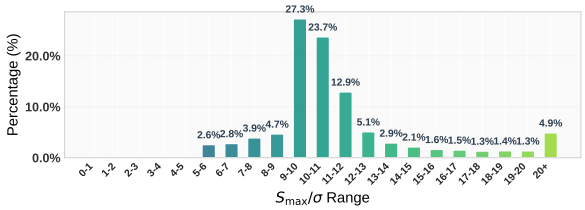


Figure 4: Distribution of the empirical maximum-to-σ ratio. Most blocks exhibit $\max(|\mathbf{X}|)/\sigma \approx 10$, reflecting outlier-dominated maxima.

Empirical maximum-to-σ ratio. To quantify the empirical relationship between the block maximum and σ, we compute the ratio $\max(|\mathbf{X}|)/\sigma$ over 10k tensor blocks. As shown in Figure 4, the block maximum is frequently dominated by rare outliers, with $\max(|\mathbf{X}|)$ commonly reaching $\sim 10\sigma$. This systematic inflation implies that the default Max-scaling strategy selects a clipping threshold far into the tail of the distribution, substantially exceeding the theoretical optimum.

Half-scaling: Bridging theory and practice. Motivated by the gap between the empirical block maximum ($\sim 10\sigma$) and the theoretical optimum ($\alpha^* \approx 4.14\sigma$), we propose **Half-S**, which applies a single half-scaling operation:

$$\textbf{Half-S:} \qquad S \leftarrow \frac{1}{2} S_{\max}. \quad (11)$$

Equivalently, this corresponds to setting the clipping threshold as $\alpha \leftarrow \max(|\mathbf{X}|)/2$. This Half-S rule is a simple and effective default that already yields clear improvements over standard Max-scaling. For improved robustness and accuracy, we further use a per-block MSE-based threshold selection strategy, which directly chooses the clipping threshold that minimizes the quantization error for each block. Therefore, Half-S can be viewed as a simple and hardware-friendly approximation, while the MSE-based strategy provides a more robust instantiation of the same principle.

Improved grid utilization and reduced error. Half-S directly alleviates the representation collapse identified in Section 4.1 by reallocating quantization resolution toward the dense low-magnitude region. At the block level, as shown in Figure 5, Max-scaling maps the majority of values (e.g., 30 dominant values) onto only 6 FP4 levels, whereas Half-S increases the number of actively utilized levels to 10, substantially improving grid utilization and preserving representational

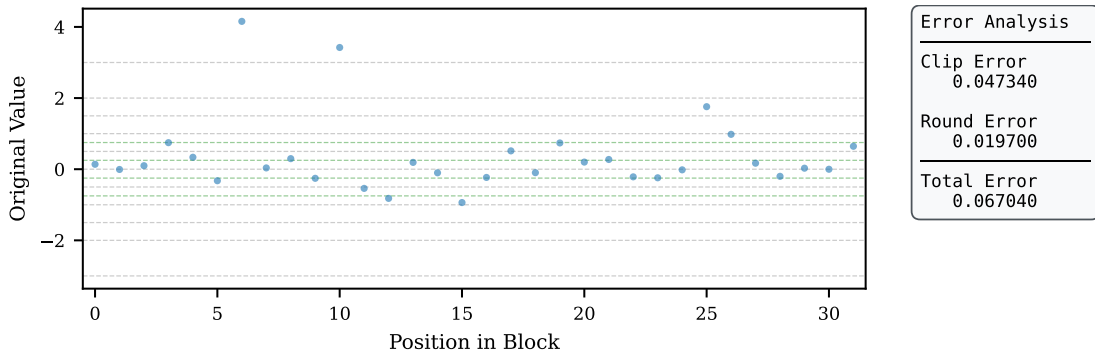


Figure 5: Improved quantization grid utilization at the block level under Half-S. In the same setting as Figure 2, Half-S enables 10 active MXFP4 quantization levels to represent the dominant values, substantially alleviating the representation collapse observed under Max-scaling. As a result, the rounding error is significantly reduced from 0.12526 to 0.0197, reflecting more effective use of the quantization grid in the dense low-magnitude region.

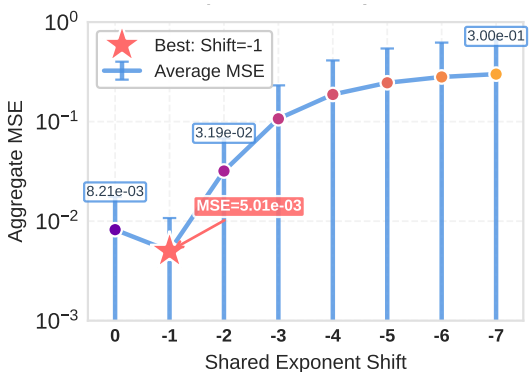


Figure 6: $S_{\max}/2$ (exponent shift=-1) achieves the global minimum error, empirically validating our theoretical analysis established in over 10k tensor blocks.

diversity. As a result, the rounding error is reduced from 0.12526 under Max-scaling to 0.0197 with Half-S. At scale, Figure 6 shows that Half-S ($S_{\max}/2$) achieves the global minimum aggregate MSE across shared-exponent shifts, while more aggressive scaling ($S_{\max}/4$) incurs rapidly increasing error due to dominant clipping.

Hardware efficiency. Half-S incurs negligible overhead by implementing division by two via bit shifting; under MXFP4 with E8M0 scaling, this reduces to a single shared-exponent decrement. As a result, Half-S preserves high efficiency while improving accuracy.

5 Experiments

In this section, we evaluate the practical effectiveness of Half-S in LLMs training, covering large-scale pretraining, instruction fine-tuning, ablation studies, and performance analysis.

5.1 Experimental Setup

We evaluate Half-S across two primary regimes: large-scale pretraining using the OLMo-7B on the Dolma dataset (Soldaini et al., 2024), and mathematical instruction tuning using Llama-2-7B on the MAmmoTH corpus (Yue et al., 2023). All experiments are conducted using a custom simulated MXFP4 operator. We apply Half-S to weights and activations while maintaining optimizer states in high precision. We compare our approach against **BF16**, COAT* (Xi et al., 2024)(COAT with MXFP4 adaptation, excluding optimizer quantization), and the search-based **Four Over Six** (Cook et al., 2025) baseline(with MXFP4 adaptation).Detailed hyperparameters and evaluation metrics are provided in Appendix C.

5.2 Accuracy Experiments

5.2.1 LLM Pretraining

We evaluate pretraining stability on OLMo-7B and OLMo-1B. As presented in Figure 7 and Table 1, Half-S provides a strong 4-bit training baseline and substantially narrows the gap to BF16.

Closing the gap to BF16 convergence. The most critical finding is that Half-S consistently tracks the BF16 baseline more closely than prior 4-bit methods in both convergence behavior and final accuracy. On OLMo-7B, Half-S achieves 46.81% zero-shot accuracy compared to BF16's 48.25% a gap of only 1.44 percentage points, recovering 97% of BF16 accuracy. This advantage scales consistently: on OLMo-1B, Half-S achieves 49.70% versus BF16's 52.00%, demonstrating scale-invariant robustness.

Resolving representation collapse. In stark contrast, COAT* and the search-based Four Over Six suffer from significant degradation (e.g., trailing BF16 by > 20 points in PPL on 7B). These baselines fail because they prioritize outlier coverage at the expense of the dense central mass. By correcting this bias, Half-S restores the representational fidelity of small-magnitude gradients, ensuring that the training loss trajectory remains tightly coupled with the high-precision baseline throughout the optimization process.

Further validation experiments. We further evaluate Half-S on larger-scale and different architectural settings, including OLMo-30B (dense) and DeepSeek-23B (MoE). On OLMo-30B trained for 8B tokens, Half-S achieves a loss of 3.387, compared with 3.697 for MXFP4 and 3.232 for BF16. On DeepSeek-23B trained for 2B tokens, Half-S achieves a loss of 3.384, compared with 3.400 for MXFP4 and 3.373 for BF16. These results indicate that the advantage of Half-S extends beyond smaller dense models to both larger models and MoE architectures.

5.2.2 LLM Fine-tuning

To evaluate the robustness of Half-S under instruction tuning, we fine-tune Llama-2-7B on the MAmmoTH corpus and assess performance on mathematical reasoning benchmarks. As detailed in Table 2, Half-S achieves an average accuracy of 24.19%, with only a 0.40-point gap from the BF16 baseline upper bound (24.59%).

Preserving fidelity in reasoning chains. Unlike general pretraining, mathematical reasoning requires preserving subtle signal differences to maintain the integrity of multi-step logical chains. The degradation observed in Four Over Six (Avg 19.77%) reveals a critical *misalignment*: by optimizing for the upper tail range ($[4, 6]$), it wastes representational capacity on values that rarely appear in fine-tuning gradients. Similarly, COAT* introduces excessive quantization noise by stretching the grid to cover outliers, effectively drowning out the fine-grained updates required for correcting logic errors. Half-S addresses this by explicitly prioritizing the signal-to-noise ratio of the central distribution. By doubling the effective resolution for the majority of updates, it minimizes error propagation in chain-of-thought generation, recovering the reasoning capabilities of the BF16 baseline.

5.2.3 Ablation Study on Scaling Factors

To validate the optimality of the half-scaling strategy, we compare Half-S against the standard baseline and an aggressive scaling variant. As shown in Table 3, the choice of scaling factor involves a critical trade-off between grid utilization (rounding error) and dynamic range coverage (clipping error).

The trade-off zone of scaling. Standard MXFP4 (S_{max}) ensures zero clipping but suffers from coarse granularity, resulting in a degraded PPL of 63.32 due to representation collapse in the dense central region. Conversely, the aggressive MXFP4 ($S_{\text{max}}/4$) strategy pushes the quantization threshold too low ($\approx 2.5\sigma$). This triggers catastrophic clipping of activation outliers, causing the training to diverge with a PPL of 252.38.

Optimality of Half-S. Half-S effectively identifies the operational sweet spot ($\approx 5\sigma$). By halving the scale, it doubles the representational density for the majority of values without crossing the critical clipping boundary that destroys model convergence. This balance allows Half-S to achieve a PPL of 39.81, recovering 99% of the BF16 baseline performance (34.68) and significantly outperforming both the conservative max-scaling and the overly aggressive MXFP4 ($S_{\text{max}}/4$) approach.

Effect of the fallback mechanism. Although fixed Half-S already improves over standard Max-scaling, its performance can still degrade on blocks whose statistics are not well matched to a fixed half-scaling rule. The fallback mechanism mitigates this issue by providing a more suitable threshold for such cases. As shown in Table 3, adding fallback improves the average perplexity from 47.35 to 39.81 and the average zero-shot accuracy from 42.08% to 43.99%. This suggests that the fallback mechanism mainly contributes additional robustness, while the main improvement still comes from the Half-S scaling principle itself.

5.2.4 Verification of Half-S on Attention

Attention layers are notoriously sensitive to quantization noise due to the heavy-tailed nature of activation outliers. Hence, we extend our evaluation to the attention modules (Q, K, V projections) of OLMo-1B to verify generalizability in Table 4 and Figure 8.

Recovery of attention fidelity. Quantizing attention heads typically triggers severe perfor-

Table 1: Pretraining performance comparison on OLMo-7B (left) and OLMo-1B (right) . Metrics are divided into Perplexity (top) and Zero-shot Accuracy (bottom).

	Avg Loss ↓	WikiText ↓	C4 ↓	Pile ↓	Avg PPL ↓
BF16	2.879	33.785	27.350	20.224	27.120
COAT*	3.412	70.182	46.350	34.899	50.477 (+23.357)
Four Over Six	3.404	64.174	45.184	33.016	47.458 (+20.338)
Half-S	**3.053**	**41.040**	**31.986**	**22.856**	**31.961 (+4.841)**

	COPA ↑	ARC(E) ↑	SciQ ↑	HellaSwag ↑	Avg Acc(%) ↑
BF16	54.00	40.53	65.20	33.29	48.25
COAT*	50.90	35.61	48.50	27.14	40.54 (-7.72)
Four Over Six	58.90	35.61	50.80	27.60	43.23 (-5.02)
Half-S	**56.00**	**40.00**	**61.00**	**30.24**	**46.81 (-1.44)**

Method	Avg Loss ↓	WikiText ↓	C4 ↓	Pile ↓	Avg PPL ↓
BF16	2.920	33.874	24.986	20.547	26.469
COAT*	3.214	47.005	33.703	26.433	35.714 (+9.245)
Four Over Six	3.330	58.648	37.769	31.276	42.564 (+16.095)
Half-S	**3.025**	**38.502**	**28.352**	**23.073**	**29.976 (+3.507)**

Method	COPA ↑	ARC(E) ↑	SciQ ↑	HellaSwag ↑	Avg Acc(%) ↑
BF16	65.00	44.39	66.30	32.40	52.00
COAT*	62.00	39.20	56.10	29.00	46.60 (-5.40)
Four Over Six	**56.00**	34.91	48.20	27.81	41.70 (-10.30)
Half-S	65.00	**42.98**	**60.80**	**30.31**	**49.70 (-2.30)**

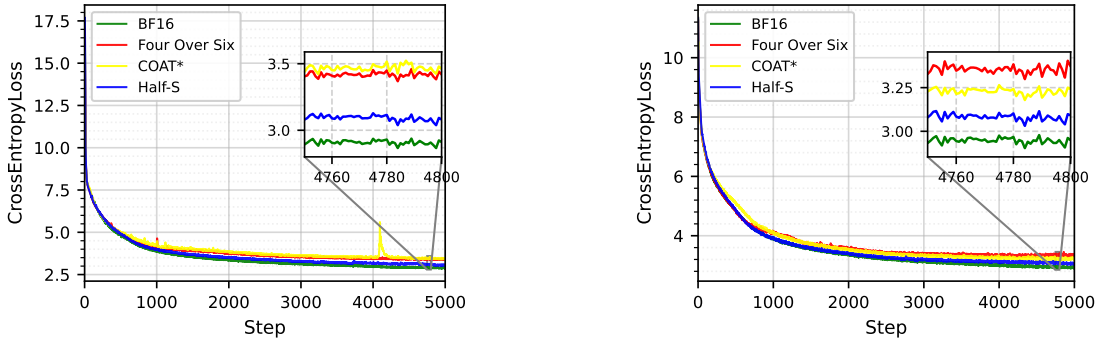


Figure 7: Pretraining loss curves for OLMo-7B (left) and OLMo-1B (right) on the Dolma dataset. Half-S (blue) remains close to the BF16 (green) baseline, with an accuracy gap of 1.44 points on OLMo-7B and 2.30 points on OLMo-1B, while others suffer from >4% degradation on OLMo-7B.

Table 2: Llama-2-7B fine-tuning performance on math and reasoning benchmarks. Half-S maintains high accuracy across complex tasks, significantly outperforming aggressive clipping strategies.

	GSM8K↑	Minerva Math↑	Avg Acc(%)↑
Origin	15.01	4.76	9.89
BF16	41.17	8.00	24.59
COAT*	38.89	**7.38**	23.14(-1.45)
Four Over Six	34.50	5.04	19.77(-4.82)
Half-S (Ours)	**41.02**	7.36	**24.19(-0.40)**

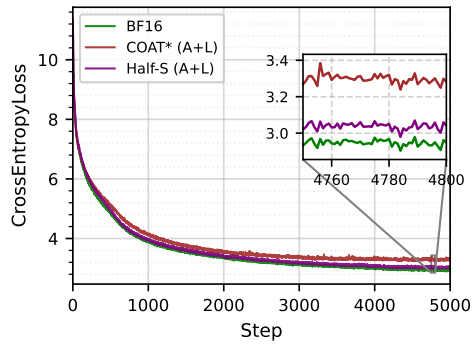


Figure 8: Stability of Attention Layer quantization. Half-S (purple) effectively eliminates the significant gap of 7.5% observed in COAT* (red), reducing it to below 4.3% and achieving BF16-level convergence.

mance collapse due to the inability of standard metrics to handle outlier-dominated distributions.

Table 3: Ablation study of scaling factors on OLMo-7B. Aggressive scaling ($S_{\max}/4$) leads to divergence due to excessive clipping, while Half-S identifies the optimal trade-off between grid resolution and range coverage.

	Avg Loss ↓	WikiText ↓	C4 ↓	Pile ↓	Avg PPL ↓
BF16	4.0121	46.10	31.08	26.86	34.68
MXFP4 ($S_{\max}$)	4.3805	94.01	52.03	43.92	63.32 (+28.64)
MXFP4 ($S_{\max}/4$)	4.7102	406.83	211.47	138.85	252.38 (+217.70)
MXFP4* ($S_{\max}/4$)	3.5235	74.86	45.55	37.55	52.66 (+17.98)
Half-S (Ours)	3.4371	66.42	41.67	33.95	47.35 (+12.67)
Half-S* (Ours)	**4.1314**	**54.28**	**35.68**	**29.47**	**39.81 (+5.13)**

	COPA ↑	ARC(E) ↑	SciQ ↑	HellaSwag ↑	Avg Acc(%) ↑
BF16	54.00	40.53	55.30	29.61	44.86
MXFP4 ($S_{\max}$)	55.00	34.03	46.90	27.05	40.74 (-4.12)
MXFP4 ($S_{\max}/4$)	**59.00**	27.37	38.60	26.69	37.91 (-6.95)
MXFP4* ($S_{\max}/4$)	**59.00**	34.39	48.00	27.30	42.17 (-2.69)
Half-S (Ours)	54.00	36.84	50.10	27.39	42.08 (-2.78)
Half-S* (Ours)	57.00	**38.07**	**52.60**	**28.30**	**43.99 (-0.87)**

* denotes methods using the fallback mechanism.

Table 4: OLMo-1B attention verification. We compare PPL and Accuracy across different implementations of Half-S. (A+L) refers to Attention and Linear layers.

Method	Train Loss ↓	WikiText ↓	C4 ↓	Pile ↓	Avg PPL ↓
BF16	2.920	33.874	24.986	20.547	26.469
COAT*(A+L)	3.258	53.942	40.878	28.484	41.102 (+14.633)
Half-S (A+L)	**3.005**	**37.193**	**30.876**	**21.772**	**29.947 (+3.478)**

Method	COPA ↑	ARC(E) ↑	SciQ ↑	HellaSwag ↑	Avg Acc(%) ↑
BF16	65.00	44.39	66.30	32.40	52.00
COAT*(A+L)	56.50	37.28	55.85	28.44	44.52 (-7.48)
Half-S (A+L)	55.00	**41.93**	**63.50**	**30.20**	**47.66 (-4.34)**

Table 4 reveals that while standard MXFP4 causes PPL to spike to 41.102, Half-S (A+L) achieves **near-lossless recovery**, attaining an Average PPL

Table 5: Comparison with MXFP8 and NVFP4 on OLMo-1B pretraining. Half-S remains competitive with MXFP8 while consistently outperforming NVFP4 on both perplexity and zero-shot accuracy.

Method	Train Loss ↓	WikiText ↓	C4 ↓	Pile ↓	Avg PPL ↓
BF16	2.920	33.874	24.986	20.547	26.469
MXFP8	2.933	33.477	30.157	20.951	28.195 (+1.726)
NVFP4	3.072	41.397	35.115	23.967	33.493 (+7.024)
Half-S	**3.025**	**38.502**	**28.352**	**23.073**	**29.976 (+3.507)**
Method	**COPA ↑**	**ARC(E) ↑**	**SciQ ↑**	**HellaSwag ↑**	**Avg Acc(%) ↑**
BF16	64.90	44.39	66.30	32.40	52.00
MXFP8	66.00	46.40	64.60	32.00	52.30 (+0.30)
NVFP4	56.90	42.20	60.60	29.70	47.40 (-4.60)
Half-S	65.00	**42.98**	**60.80**	**30.31**	**49.70 (-2.30)**

of 29.947 and Zero-shot Accuracy of 47.66%. These metrics are within 4.5% of the BF16 baseline, demonstrating that Half-S effectively preserves signal fidelity in outlier-heavy projections without requiring higher-precision retention.

Convergence stability. As shown in Figure 8, Half-S exhibits remarkably stable convergence behavior. It achieves a final training loss of 3.4151, significantly outperforming the standard 4-bit configuration (3.5620) and matching the BF16 trajectory. By preventing the under-utilization of the quantization grid in Q, K, V projections, Half-S minimizes the quantization noise that typically disrupts the attention mechanism, thereby ensuring stable end-to-end 4-bit optimization.

5.2.5 Comparison with MXFP8 and NVFP4

We compare Half-S against MXFP8 and NVFP4 on OLMo-1B pretraining to evaluate its performance across different precision formats, as shown in Table 5.

Comparison across different precision formats. MXFP8 remains very close to the BF16 baseline, achieving 52.30% average zero-shot accuracy. By contrast, NVFP4 shows a noticeable degradation, increasing average perplexity by 7.024 and reducing average zero-shot accuracy by 4.60 points relative to BF16. This highlights the difficulty of maintaining training quality under direct 4-bit quantization without distribution-aware scaling.

Effect of Half-S. Half-S significantly alleviates this degradation. Relative to NVFP4, it reduces the perplexity increase by about 50% and improves the average zero-shot accuracy to 49.70%. These results indicate that Half-S can substantially narrow the gap between 4-bit training and higher-precision baselines.

5.3 Efficiency Analysis

We analyze the computational efficiency assuming native FP4 support. For memory-bound 7B models, 4-bit quantization effectively quadruples the available bandwidth. Concretely, the BF16 baseline requires 10 seconds, whereas MXFP8 reduces the runtime to 7.1 seconds (yielding a $\sim$1.4$\times$ speedup), and MXFP4 further reduces it to 5.5 seconds, corresponding to an estimated **1.8$\times$ speedup** over BF16. Crucially, the simple bit-shift correction of Half-S incurs negligible computational overhead, allowing these theoretical bandwidth gains to be fully realized in practice.

6 Conclusion

In this paper, we identify *representation collapse* as the core limitation of max-scaling in 4-bit LLM training. Guided by theoretical analysis, we propose **Half-S**, an efficient correction that halves the max-based scale. Extensive experiments show that Half-S enables more stable and accurate 4-bit training, substantially narrowing the gap to BF16 while preserving the efficiency benefits of low-precision computation.

7 Limitations

Although Half-S is a general scaling strategy, our experiments in this work are mainly conducted on MXFP4, and its effectiveness on other 4-bit formats like HiFloat4 (Luo et al., 2026) remains to be further validated. In addition, Half-S is most beneficial under heavy-tailed tensor distributions; its behavior under less heavy-tailed regimes deserves further study. Finally, our end-to-end results are based on a simulated MXFP4 operator, so validation on a fully native FP4 hardware-software stack is still needed to confirm practical deployment efficiency.

Acknowledgement This work was supported by the National Natural Science Foundation of China (Nos. 62525601, 62476018), and the Postdoctoral Fellowship Program of CPSF (No. BX20250487). We would also like to thank Yuanyong Luo from Huawei for his valuable support and insightful discussions.

References

Felix Abecassis, Anjulie Agrusa, Dong Ahn, Jonah Alben, Stefania Alborghetti, and 1 others. 2025. Pre-

training large language models with NVFP4. *arXiv preprint arXiv:2509.25149*.

Advanced Micro Devices (AMD). 2023. AMD Instinct MI300 Series Accelerators. Advancing AI Event Presentation. Available at: https://www.amd.com/en/events/advancing-ai.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2022. PaLM: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jack Cook, Junxian Guo, Guangxuan Xiao, Yujun Lin, and Song Han. 2025. Four over six: More accurate NVFP4 quantization with adaptive block scaling. *Preprint*, arXiv:2512.02010.

DeepSeek-AI. 2024. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. *arXiv preprint arXiv:2208.07339*.

Tim Dettmers, Mike Lewis, and Luke Zettlemoyer. 2021. 8-bit optimizers via block-wise quantization. *arXiv preprint arXiv:2110.02861*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv preprint arXiv:2305.14314*.

Leo Gao, Stella Biderman, Sid Black, and 1 others. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, and 1 others. 2024. Olmo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, and 1 others. 2019. A study of bfloat16 for deep learning training. *arXiv preprint arXiv:1905.12322*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, and 1 others. 2022. Solving quantitative reasoning problems with language models. *arXiv preprint arXiv:2206.14858*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, and 1 others. 2024. DeepSeek-V3 Technical Report. *arXiv preprint arXiv:2412.19437*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.

Yuanyong Luo, Jing Huang, Yu Cheng, Ziwei Yu, Kaihua Tang, Xinda Ma, Xin Wang, Anping Tong, Guipeng Hu, Yun Xu, Mehran Taghian, Peng Wu, Guanglin Li, Yunke Peng, Tianchi Hu, Minqi Chen, Michael Bi Mi, Hu Liu, Xiping Zhou, and 3 others. 2026. Hifloat4 format for language model inference. *Preprint*, arXiv:2602.11287.

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

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and 1 others. 2017. Mixed precision training. *arXiv preprint arXiv:1710.03740*.

Paulius Micikevicius, Dusan Stosic, Neil Burgess, and 1 others. 2022. FP8 formats for deep learning. *arXiv preprint arXiv:2209.05433*.

NVIDIA. 2022. NVIDIA H100 Tensor Core GPU Architecture. Technical report, NVIDIA Corporation. Whitepaper, SP-11117-001_v1.0.

NVIDIA Corporation. 2025. Transformer engine.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*. https://arxiv.org/abs/2303.08774.

Houwen Peng, Kan Wu, Yixuan Wei, and 1 others. 2023. FP8-LM: Training FP8 large language models. *arXiv preprint arXiv:2310.18313*.

Colin Raffel, Noam Shazeer, Adam Roberts, and 1 others. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. DeepSpeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Bita Darvish Rouhani, Luke Hall, Merat Forootan, and 1 others. 2023. Microscaling data formats for deep learning. In *Proceedings of the 50th Annual International Symposium on Computer Architecture (ISCA)*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, and 1 others. 2019. Megatron-LM: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, and 1 others. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*.

Hugo Touvron, Louis Martin, Kevin Stone, Hugo Albert, Amjad Almahairi, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ben Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Haocheng Xi, Han Cai, Ligeng Zhu, and 1 others. 2024. CoAt: Compressing optimizer states and activation for memory-efficient FP8 training. *arXiv preprint arXiv:2410.19313*.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. 2022. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in Neural Information Processing Systems*, 35:2177–2191.

Xiang Yue, Xingwei Qu, Ge Zhang, and 1 others. 2023. MAmmoTH: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*.

Yu Zhang, Hui-Ling Zhen, Mingxuan Yuan, and Bei Yu. 2025. MOSS: Efficient and accurate FP8 LLM training with microscaling and automatic scaling. *arXiv preprint arXiv:2511.05811*.

A Empirical Validation of the Laplace Assumption

To empirically validate the Laplace distribution assumption used in our clipping and scaling analysis, we conduct a goodness-of-fit test on both forward activations and backward gradients collected at 1000 training iterations of OLMo-1B.

Given a tensor $\mathbf{x}$, we assess its fit to a Laplace distribution, $\mathrm{Laplace}(0, b)$, where the scale parameter b is estimated via maximum likelihood as

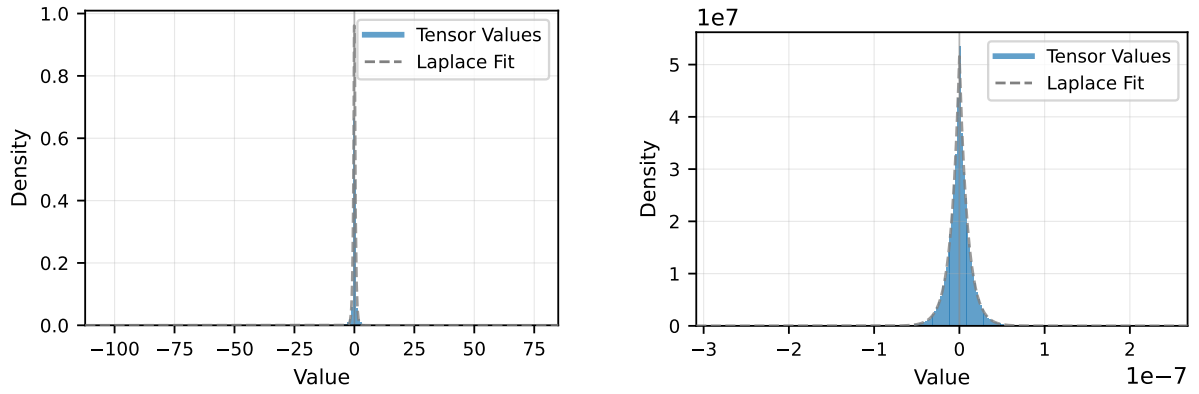
$$\hat{b} = \mathbb{E}[|x|].$$

We fix the location parameter to zero, which is empirically justified by the near-zero sample means observed in both cases. Due to the large tensor dimensionality, we randomly subsample 5×10^5 elements for statistical testing.

We apply the Kolmogorov–Smirnov (KS) test to quantify the discrepancy between the empirical distribution and the fitted Laplace model. In addition, we report skewness and kurtosis as diagnostic statistics to characterize symmetry and tail behavior.

For the forward tensor, we obtain an estimated scale parameter $\hat{b} = 0.41$, with negligible skewness (0.01) and a near-zero sample mean. The empirical kurtosis, however, reaches 59.99, substantially exceeding the theoretical Laplace value of 6, which indicates the presence of rare but extreme outliers. This results in a KS statistic of 0.074. We note that such a statistical discrepancy is expected in large-sample regimes and is primarily driven by tail discrepancies rather than deviations in the central mass. Since the MSE-optimal clipping threshold is dominated by the central distribution, and the bulk of the activation distribution exhibits a near-linear decay in log-density consistent with a Laplace model, we find it appropriate to adopt the Laplace assumption for our theoretical analysis.

In contrast, the backward gradient tensor aligns very closely with the Laplace assumption. The empirical kurtosis is 7.02, nearly matching the theoretical value of 6. The KS statistic is also exceptionally small at 0.0073, signifying a high degree of fit between the empirical data and the model. While statistical tests are sensitive to minuscule deviations in large samples, the combination of closely matched moment statistics and a minimal distributional distance provides strong evidence to model the backward gradients as Laplace-distributed.



(a) Forward tensor at 1000 iterations

(b) Backward tensor at 1000 iterations

Figure 9: Half-S ($S_{\max}/2$) achieves the global minimum MSE for both forward and backward tensors, empirically validating our theoretical optimality bounds over 10,000 tensor blocks(10333 tensors).

Taken together, these results justify the Laplace modeling assumption for both forward and backward tensors in practice, with particularly strong agreement observed for backward gradients. This empirical evidence supports the validity of the clipping and scaling optimality analysis developed in Section 4.2.1.

B Optimal Clipping Threshold for MXFP4 under Laplace Distribution

The analysis explicitly accounts for the non-uniform floating-point grid of MXFP4 and decomposes the error into clipping and rounding components.

Problem setup. Let the random variable X follow a zero-mean Laplace distribution,

$$p(x) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right), \tag{12}$$

where $b > 0$ is the scale parameter. We apply symmetric clipping to the interval $[-\alpha, \alpha]$ and subsequently quantize using MXFP4 with a shared scaling factor

$$S = \frac{\alpha}{6}. \tag{13}$$

The set of representable positive values of MXFP4 (E2M1) is

$$\mathcal{Q}_+ = \{0,\ 0.5,\ 1,\ 1.5,\ 2,\ 3,\ 4,\ 6\}. \tag{14}$$

MSE decomposition. The total MSE is decomposed as

$$\mathrm{MSE}(\alpha) = E_c(\alpha) + E_r(\alpha), \tag{15}$$

where E_c denotes the clipping error and E_r denotes the rounding error induced by MXFP4 quantization within the clipping range.

Clipping error. For a Laplace distribution, the clipping MSE can be written as

$$\begin{aligned} E_c(\alpha) &= \mathbb{E}\big[(|X| - \alpha)_+^2\big] \\ &= \int_{|x|>\alpha} (|x| - \alpha)^2\, p(x)\, dx \\ &= 2 \int_{\alpha}^{\infty} (x - \alpha)^2\, \frac{1}{2b} \exp\Big(-\frac{x}{b}\Big)\, dx, \end{aligned} \tag{16}$$

where the factor of 2 follows from symmetry. Evaluating the integral yields the closed-form expression

$$E_c(\alpha) = 2b^2 \exp\Big(-\frac{\alpha}{b}\Big). \tag{17}$$

Rounding error for MXFP4. Let $\{q_i\}_{i=0}^{7}$ denote the ordered elements of $\mathcal{Q}_+$. The bin boundaries on the positive axis are defined as

$$\begin{aligned} m_0 &= 0, \\ m_{i+1} &= \frac{q_i + q_{i+1}}{2}, \quad i = 0, \ldots, 6, \\ m_8 &= 6. \end{aligned} \tag{18}$$

The rounding MSE can be written as

$$E_r(\alpha) = \sum_{i=0}^{7} \int_{m_i S}^{m_{i+1} S} (x - q_i S)^2\, \frac{1}{b} \exp\Big(-\frac{x}{b}\Big)\, dx. \tag{19}$$

Scale normalization. Introduce the dimensionless variables

$$z = \frac{x}{b}, \qquad \hat{\alpha} = \frac{\alpha}{b}, \qquad \hat{S} = \frac{S}{b} = \frac{\hat{\alpha}}{6}. \tag{20}$$

Substituting into Eq. (19) yields

$$E_r(\alpha) = b^2\, \Phi(\hat{\alpha}), \tag{21}$$

where

$$\Phi(\hat{\alpha}) = \sum_{i=0}^{7} \int_{m_i\hat{S}}^{m_{i+1}\hat{S}} (z - q_i\hat{S})^2 e^{-z}\, dz. \quad (22)$$

Similarly, Eq. (16) becomes

$$E_c(\alpha) = 2b^2 e^{-\hat{\alpha}}. \quad (23)$$

Therefore, the normalized MSE takes the form

$$\frac{\mathrm{MSE}(\alpha)}{b^2} = 2e^{-\hat{\alpha}} + \Phi(\hat{\alpha}), \quad (24)$$

which depends on the clipping threshold only through $\hat{\alpha}$.

Optimal threshold. Minimizing Eq. (24) yields the optimality condition

$$\Phi'(\hat{\alpha}^*) = 2e^{-\hat{\alpha}^*}. \quad (25)$$

Equation (25) depends only on the MXFP4 grid structure. Consequently, the optimal clipping threshold scales linearly with the Laplace scale parameter,

$$\alpha^* = \hat{\alpha}^*\, b. \quad (26)$$

Numerical solution via bisection. The function $\Phi(\hat{\alpha})$ defined in Eq. (22) does not admit a closed-form expression due to the piecewise structure induced by the non-uniform MXFP4 grid. We therefore evaluate $\Phi(\hat{\alpha})$ numerically by explicitly integrating over each quantization bin.

Specifically, for a given $\hat{\alpha}$, the normalized scale is $\hat{S} = \hat{\alpha}/6$, and the integration domain is partitioned according to the MXFP4 decision boundaries $\{m_i\hat{S}\}_{i=0}^{8}$. Within each bin $[m_i\hat{S}, m_{i+1}\hat{S}]$, The rounding error is integrated as

$$\int_{m_i\hat{S}}^{m_{i+1}\hat{S}} (z - q_i\hat{S})^2 e^{-z}\, dz.$$

The total value of $\Phi(\hat{\alpha})$ is obtained by summing the contributions from all bins. In practice, each integral is evaluated using Simpson's rule with a fixed number of subdivisions, which is sufficient given the smoothness of the integrand.

To obtain the derivative $\Phi'(\hat{\alpha})$, We employ a central finite-difference approximation,

$$\Phi'(\hat{\alpha}) \approx \frac{\Phi(\hat{\alpha} + h) - \Phi(\hat{\alpha} - h)}{2h},$$

with a small stepsize h. This approach avoids the need to differentiate the piecewise-defined integrals analytically.

We then define the root-finding function

$$g(\hat{\alpha}) \triangleq \Phi'(\hat{\alpha}) - 2e^{-\hat{\alpha}},$$

which is continuous on $\hat{\alpha} > 0$. Since $\Phi(\hat{\alpha})$ is monotonically increasing and $2e^{-\hat{\alpha}}$ is strictly decreasing, the equation $g(\hat{\alpha}) = 0$ admits a unique solution. The root is found using a bisection method on an interval $[\hat{\alpha}_{\min}, \hat{\alpha}_{\max}]$ chosen such that $g(\hat{\alpha}_{\min})$ and $g(\hat{\alpha}_{\max})$ have opposite signs.

Using this procedure, we obtain

$$\hat{\alpha}^* \approx 5.86453, \quad (27)$$

which corresponds to the MSE-optimal clipping threshold

$$\alpha^* = \hat{\alpha}^*\, b \approx 5.86\, b.$$

C Hyperparameters and Evaluation Metrics in Experiment

C.1 Optimization and Training Hyperparameters

All models are trained using the AdamW optimizer (Loshchilov and Hutter, 2019) with decoupled weight decay. Unless otherwise specified, we fix the optimizer hyperparameters to $\beta_1 = 0.9$, $\beta_2 = 0.95$, and a weight decay coefficient of 0.1. Global gradient norm clipping is applied with a threshold of 1.0 to ensure training stability under low-precision arithmetic.

For large-scale pretraining, models are trained on the Dolma dataset for a total of 20B tokens using a global batch size of 2048 and a fixed sequence length of 2048. For instruction fine-tuning, we train for a single epoch on the MAmmoTH mathematical reasoning corpus. The proposed Half-S mechanism is applied to quantize model weights and activations, while optimizer states are maintained in higher precision to preserve numerical stability during 4-bit training.

C.2 Models and Datasets

For pretraining experiments, we use the OLMo-7B and OLMo-1B architecture (Groeneveld et al., 2024) trained on the Dolma corpus (Soldaini et al., 2024). Evaluation is performed using perplexity on standard language modeling benchmarks, including WikiText-103 (Merity et al., 2016), C4 (Raffel et al., 2020), and The Pile (Gao et al., 2020). In addition, we report zero-shot accuracy on reasoning benchmarks, including COPA, ARC-Easy, and HellaSwag.

For fine-tuning experiments, we instruction-tune Llama-2-7B (Touvron et al., 2023) on the MAmmoTH dataset (Yue et al., 2023). Performance is evaluated on challenging mathematical and reasoning benchmarks, including GSM8K (Cobbe et al., 2021), and Minerva Math (Lewkowycz et al., 2022).

C.3 Evaluation Metrics

We report perplexity (PPL) for language modeling performance and zero-shot accuracy for downstream reasoning tasks. Unless otherwise noted, all reported results are averaged over the full evaluation set.