RETHINKING SHRINKAGE BIAS IN LLM FP4 PRETRAINING: GEOMETRIC ORIGIN, SYSTEMIC IMPACT, AND UFP4 RECIPE

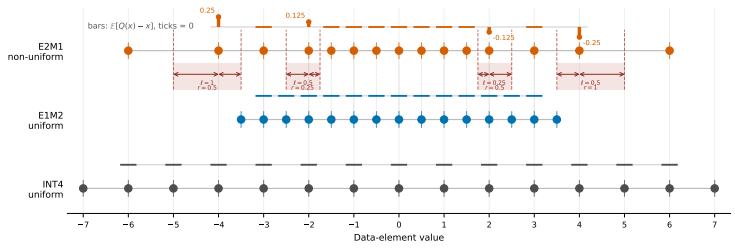
Anonymous authors
Paper under double-blind review

ABSTRACT

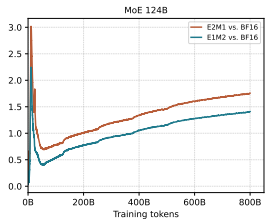
FP4 training promises substantial reductions in memory and computation cost for LLM pretraining, yet current FP4 hardware paths and recipes, including NVIDIA Blackwell/Rubin-class systems and AMD MI350-series GPUs, remain centered on E2M1 data elements. In this study, we identify a fundamental limitation of that choice: non-uniform formats such as E2M1 inherently suffer from *Shrinkage Bias*, a systematic negative rounding error caused by the geometric asymmetry of their representable bins. We show that this bias accumulates multiplicatively across layers and is amplified by the Random Hadamard Transform (RHT), providing a unified explanation for the training instability observed in existing E2M1-based FP4 recipes. In contrast, uniform grids (E1M2/INT4) bypass this grid-geometry error and better convert the improved bucket utilization from RHT into higher quantization quality. Based on this finding, we propose UFP4, a uniform 4-bit training recipe that applies RHT to all three training GEMMs while restricting stochastic rounding to dY alone. On Dense 1.5B, MoE 7.9B, and MoE 124B long-run pretraining, UFP4 consistently achieves lower BF16-relative loss degradation than strong E2M1-based baselines, supported by scaling-law analysis and ablation studies. Our results suggest that future accelerators should support E1M2/INT4-style uniform 4-bit grids as first-class training primitives alongside E2M1.

1 INTRODUCTION

Continued scaling of Large Language Models (LLMs) has pushed the frontier of model capabilities, yet the rapidly growing cost, memory footprint, and energy consumption of pretraining demand more efficient numerical formats (Hoffmann et al., 2022; Zhao et al., 2026). Low-precision training has therefore become a key direction for reducing training cost, with FP8 already adopted by several large-scale training systems (DeepSeek-AI et al., 2024; Team et al., 2025). The basic idea is to represent activations, weights, and gradients in lower precision during both forward and backward passes, leveraging the high-throughput low-precision matrix-multiply units on modern



(a) Representable values and RTNE bins of E2M1 vs. E1M2/INT4.



(b) BF16-relative loss degradation (124B MoE).

Figure 1: (a) Red spans mark RTNE rounding bins; bars show expected rounding error per bin. E2M1's non-uniform bins exhibit systematic toward-zero bias (Shrinkage Bias) due to geometric asymmetry, while E1M2/INT4 uniform bins remain unbiased. (b) This grid-level advantage translates to training quality: on 124B MoE long-run pretraining, E1M2-based UFP4 significantly outperforms E2M1-based baselines in BF16-relative loss degradation.

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.

accelerators (Micikevicius et al., 2022). More recently, accelerators such as NVIDIA Blackwell expose native FP4 computation paths (NVIDIA et al., 2026), offering a new opportunity to further halve the precision and reduce training cost.

Current FP4 training interfaces are largely designed around E2M1, a 4-bit floating-point format with 2 exponent bits and 1 mantissa bit. For example, MXFP4 (Rouhani et al., 2023a) and NVFP4 (NVIDIA et al., 2026) maintain independent scale factors for fine-grained tensor blocks, trading off dynamic range, precision, and hardware efficiency through different block sizes and scale hierarchies. Compared with MXFP4, NVFP4 typically provides better training accuracy through smaller blocks and a more refined scaling design. Nevertheless, with only 16 representable values, end-to-end FP4 training remains challenging: without additional stabilization, FP4 recipes suffer from severe convergence issues and loss degradation relative to BF16 (Wang et al., 2025; Chmiel et al., 2026; NVIDIA et al., 2026). To mitigate these problems, the NVFP4 recipe introduces the Random Hadamard Transform (RHT) and stochastic rounding (SR) to handle outliers and stabilize gradient updates (Ashkboos et al., 2024; Gupta et al., 2015). Subsequent work further reduces the gap between FP4 and BF16 training through improved scale design, rounding or gradient-estimation methods, outlier control, and quantization-aware training (Chen et al., 2026; Panferov et al., 2026; Cao et al., 2026; Cook et al., 2026; Li et al., 2026).

Despite effectiveness, most existing methods still retain the E2M1 data-element format unchanged. This paper revisits that default choice. Our central observations are: **(1)** Non-uniform formats such as E2M1 inherently suffer from *Shrinkage Bias*—a negative expected rounding error under Round-to-Nearest-Even (RTNE) caused by the geometric asymmetry of their representable bins. **(2)** This bias accumulates multiplicatively across layers, causing systematic signal decay. Moreover, RHT exacerbates the problem under E2M1 by shifting tensor mass into the most asymmetric bins, further degrading training stability. In contrast, as illustrated in Figure 1a, uniform grids (e.g., E1M2/INT4) bypass this grid-geometry error entirely, better match the post-RHT distribution, and convert improved bucket utilization into consistently higher quantization quality.

Based on this finding, we propose UFP4, a uniform 4-bit training recipe built on an E1M2/INT4-style grid. UFP4 applies RHT to the operands of all three linear-layer training GEMMs, i.e., forward (FPROP, `fwd_y`), data-gradient (DGRAD, `bwd_dx`), and weight-gradient (WGRAD, `bwd_dw`), and uses stochastic rounding only when quantizing the upstream gradient dY. The recipe demonstrates that full-RHT training is not inherently harmful; rather, the issue lies in the mismatch between E2M1 and the post-RHT tensor regime. With a uniform grid, RHT extends to all three paths while preserving stable long-run gains: on a 124B MoE long-run pretrain, E1M2-based UFP4 significantly outperforms strong E2M1-based baselines in BF16-relative loss degradation (Figure 1b).

We summarize our contributions as follows: 1) We identify and formalize *Shrinkage Bias*: a systematic negative rounding error inherent to non-uniform grids such as E2M1, caused by the geometric asymmetry of their RTNE bins. 2) We theoretically and empirically show that Shrinkage Bias accumulates multiplicatively across layers and is amplified by RHT, providing a unified explanation for the training instability observed in existing E2M1-based FP4 recipes, e.g., NVFP4 recipe. 3) We propose UFP4, a 4-bit training recipe built on uniform E1M2/INT4-style grids that enables RHT on all three training GEMMs while restricting stochastic rounding to dY alone. 4) We validate UFP4 on Dense 1.5B, MoE 7.9B, and MoE 124B long-run pretraining, supported by scaling-law analysis, ablation studies, and fused-kernel benchmarks, demonstrating that uniform grids are practical first-class FP4 training formats at industrial scale.

2 PRELIMINARIES

2.1 4-BIT FORMATS AND BLOCKWISE QUANTIZATION

FP4 formats use one sign bit and split the remaining bits into exponent and mantissa fields, denoted ExMy. We consider two FP4 formats, E2M1 and E1M2, and an INT4 codebook (Figure 1a).

In practice, blockwise quantization is widely adopted to improve precision by partitioning a tensor $\mathbf{T}$ into contiguous blocks $\{B\}$ and mapping each element to a codebook level under a shared per-block scale. Let $G = \{g\}$ denote the normalized codebook of a chosen format, $g_{\max} = \max_{g \in G} |g|$ its

largest magnitude level, and ρ_G the rounding rule. The quantization process is then:

$$s_B = \frac{\max_{x_j \in B} |x_j|}{g_{\max}}, \qquad q_i = \rho_G\left(\frac{x_i}{s_B}\right) \in G.$$

We write $Q_G(\mathbf{T})$ for the quantized tensor; in arithmetic expressions it denotes the dequantized numerical tensor.[1] The rounding rule ρ_G is either Round-To-Nearest-Even (RTNE) or Stochastic Rounding (SR). RTNE selects the nearest level in G with deterministic tie-breaking. When x_i/s_B falls between adjacent levels $g_a < g_b$, SR samples:

$$\Pr[q_i = g_b] = \frac{x_i/s_B - g_a}{g_b - g_a}, \qquad \Pr[q_i = g_a] = \frac{g_b - x_i/s_B}{g_b - g_a}.$$

SR preserves the normalized value in expectation, while RTNE is deterministic. In practice, RTNE is widely adopted by advanced low-precision methods such as NVFP4 Recipe (NVIDIA et al., 2026).

2.2 Random Hadamard Transforms

Real training tensors often contain outlier coordinates, causing most codebook levels to be underutilized. Random Hadamard Transforms (RHT) address this by applying a norm-preserving rotation that disperses outlier energy across all coordinates before quantization.

Concretely, we use the Sylvester Hadamard matrices defined recursively as

$$\mathbf{H}_{2n} = \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{H}_n & \mathbf{H}_n \\ \mathbf{H}_n & -\mathbf{H}_n \end{bmatrix}, \qquad \mathbf{H}_1 = [1],$$

where $\mathbf{H}_n^\top \mathbf{H}_n = I_n$ for $n = 2^k$. An RHT additionally applies a random sign matrix $\mathbf{S}_n = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_n)$, $\epsilon_i \in \{-1, +1\}$, before the Hadamard transform. Since $\mathbf{H}'_n = \mathbf{S}_n \mathbf{H}_n$ is orthogonal, applying the same transform to the shared GEMM dimension preserves the full-precision result:

$$\mathbf{H}'_n = \mathbf{S}_n \mathbf{H}_n,$$
$$\mathbf{Y} = \mathbf{X}\mathbf{W}^\top = (\mathbf{X}\mathbf{H}'_n)(\mathbf{W}\mathbf{H}'_n)^\top.$$

The corresponding low-precision GEMM quantizes the rotated operands:

$$\widehat{\mathbf{Y}} = Q_G(\mathbf{X}\mathbf{H}'_n)\, Q_G(\mathbf{W}\mathbf{H}'_n)^\top.$$

RHT reduces outlier dominance and improves codebook utilization; however, its net effect on quantization error depends on the FP4 grid, which we analyze in Section 3.2. For the three training GEMMs, RHT is applied along the shared reduction dimension: the input-channel dimension for `fwd_y`, the output-channel dimension for `bwd_dx`, and the batch-token dimension for `bwd_dw`.

3 Shrinkage Bias in 4-bit Format Grids

In this section, we investigate the systematic *Shrinkage Bias* inherent to non-uniform quantization formats like E2M1, driven by the geometric asymmetry of their rounding bins. We demonstrate how this bias accumulates multiplicatively across deep networks and expose a hidden pitfall: standard outlier-mitigation techniques like the RHT inadvertently exacerbate this instability when paired with non-uniform grids.

3.1 Geometric Origin: Shrinkage Bias from Asymmetric RTNE Bins

Consider a block with scale $s > 0$, and let $t = |x|/s$ denote the normalized magnitude. We define *Shrinkage Bias* as a negative expected Round-to-Nearest-Even (RTNE) error in the normalized magnitude space. Specifically, for a distribution $\mathcal{P}$ over t, let the expected error $b_G(\mathcal{P}) = \mathbb{E}_{t\sim\mathcal{P}}[\rho_G(t) - t]$, where $\rho_G(\cdot)$ is the RTNE mapping to the codebook G. A value $b_G(\mathcal{P}) < 0$ indicates Shrinkage Bias.

Assuming no clipping ($t \le \max(G)$), we focus on the non-negative normalized levels of the format:

$$G_+ = \{q_0, q_1, \ldots, q_n\}, \qquad 0 = q_0 < q_1 < \cdots < q_n.$$

[1] MXFP4 uses 1×32 blocks with an E8M0 scale; NVFP4 uses 1×16 blocks with two-level scaling.

For an interior quantization level q_i ($i \in \{1, \ldots, n-1\}$), the interior of its RTNE rounding bin is

$$\mathcal{B}_i = \left(\frac{q_{i-1} + q_i}{2}, \frac{q_i + q_{i+1}}{2} \right), \qquad \ell_i = \frac{q_i - q_{i-1}}{2}, \quad r_i = \frac{q_{i+1} - q_i}{2}, \tag{1}$$

where ℓ_i and r_i are the left and right bin widths around q_i. For $t \in \mathcal{B}_i$, the magnitude error is $\rho_G(t) - t = q_i - t$. If the density inside this bin is locally uniform, we can calculate the conditional expected error by changing variables to $u = t - q_i$:

$$\mathbb{E}[\rho_G(t) - t \mid t \in \mathcal{B}_i] = \frac{1}{\ell_i + r_i} \int_{-\ell_i}^{r_i} (-u)\, du = \frac{\ell_i - r_i}{2} = \frac{2q_i - q_{i-1} - q_{i+1}}{4}. \tag{2}$$

Consequently, an asymmetric RTNE bin with $r_i > \ell_i$ inherently yields a negative expected error under uniform local density. This reveals a grid-geometry source of magnitude shrinkage, distinct from distribution-induced quantization error inside a symmetric bin.

As the mainstream low-precision format utilized in advanced pipelines (e.g., NVFP4), E2M1 inherently contains such asymmetric bins at spacing-transition points. For example, given the non-negative E2M1 magnitudes $\{0, 0.5, 1, 1.5, 2, 3, 4, 6\}$, the level $q_i = 2$ yields a conditional bias of -0.125, and $q_i = 4$ exhibits the same asymmetry at double the scale.[2] By contrast, uniform grids (e.g., E1M2) satisfy $\ell_i = r_i$ for all bins, thereby eliminating this fundamental source of bias. We visually corroborate this contrast in Figure 1a, which compares the E2M1, E1M2, and INT4 grids to explicitly illustrate both the geometric origin of Shrinkage Bias and its resulting error distribution.

> **Key Finding 1:** Non-uniform quantization formats (e.g., E2M1) inherently suffer from *Shrinkage Bias* due to the geometric asymmetry of their rounding bins, whereas uniform grids (e.g., E1M2) bypass this fundamental source of grid-geometry error.

3.2 Systemic Impact: Propagation and RHT Exacerbation

This bin-level bias is not just harmless zero-mean noise. When it enters matrix multiplications (GEMMs), it acts as a systematic attenuation that cascades through the network.

Propagation through Multiplicative Accumulation. The bin-level bias discussed above becomes critical because it survives tensor aggregation and accumulates across deep paths. The operand projection and exact GEMM expansion used below are given in Appendix E. When the residual terms are incoherent with Z, the principal signal is attenuated by a factor of $\eta \approx \alpha_A \alpha_B < 1$. This decay propagates systematically. For a sequence of K quantized GEMMs, let $\eta_k \approx \alpha_{A,k}\alpha_{B,k}$ denote the coherent attenuation factor of the k-th operation. The initial clean signal is cumulatively scaled by:

$$\prod_{k=1}^{K} \eta_k = \prod_{k=1}^{K} (1 - \delta_k) \approx \exp\left(-\sum_{k=1}^{K} \delta_k \right), \qquad \delta_k = 1 - \eta_k. \tag{3}$$

Thus, unlike zero-mean noise which tends to cancel out, this systematic shrinkage accumulates multiplicatively (see Appendix E.2 for the full derivation). Crucially, while quantization errors in `bwd_dw` are leaf-gradients directly consumed by the optimizer, errors in non-leaf outputs (`fwd_y` and `bwd_dx`) cascade through subsequent layers, actively compounding this systematic decay.

Exacerbation under RHT. While E2M1's wide dynamic range is initially a sensible choice for mitigating saturation in outlier-heavy tensors, RHT fundamentally alters this regime. By spreading outlier energy across coordinates, RHT transforms the tensor from being dynamic-range-limited to local-resolution-limited. Consequently, the bottleneck shifts from representing extreme outliers to accurately preserving the dense probability mass at typical magnitudes.

To formalize this shift, we measure utilization across K quantized magnitude buckets. Given empirical fractions p_i for each bucket (including zero), we define the bucket entropy $\mathcal{E}(G, T)$ and report the effective bucket ratio $B_{\text{eff}}(G, T)$:

$$B_{\text{eff}}(G, T) = \frac{\exp(\mathcal{E}(G, T))}{K}, \quad \text{where} \quad \mathcal{E}(G, T) = -\sum_{i=1}^{K} p_i \log(p_i + \epsilon) \tag{4}$$

[2] For $q_i = 2$: bin $(1.75, 2.5)$, $\ell_i = 0.25$, $r_i = 0.5$; Eq. (2) gives $\mathbb{E}[\rho_G(t) - t \mid t \in (1.75, 2.5)] = -0.125$.

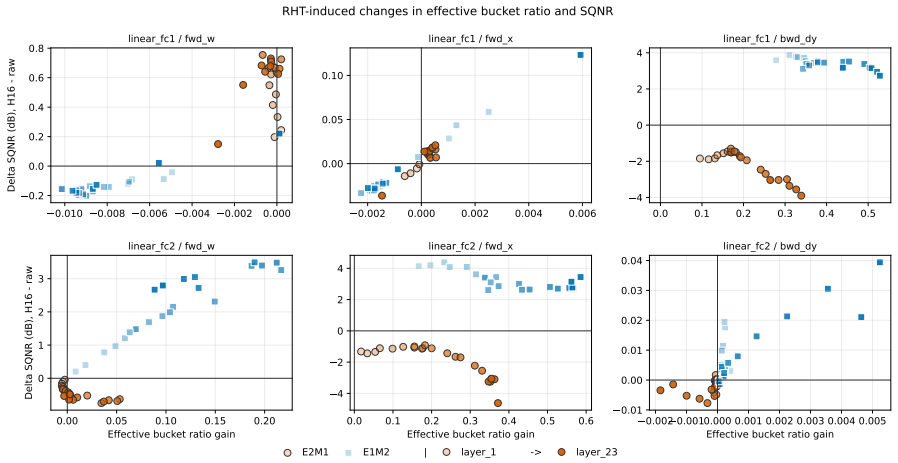


Figure 2: Impact of RHT on effective bucket ratio and SQNR. Colors distinguish the quantization formats (E2M1 vs. E1M2), and marker opacity increases with network depth. Evaluated on real linear-layer tensors: weights (`fwd_w`), inputs (`fwd_x`), and output gradients (`bwd_dy`).

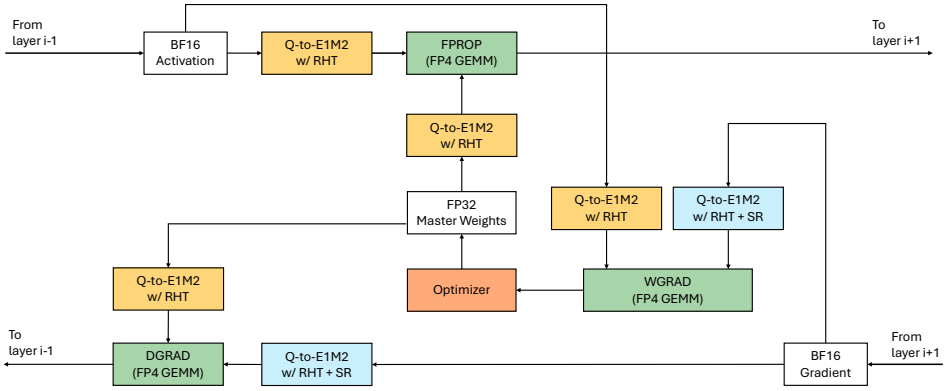


Figure 3: Overview of the UFP4 recipe. By leveraging a uniform grid, UFP4 enables pervasive RHT across all forward and backward GEMMs.

where $B_{\text{eff}} \in [1/K, 1]$ spans from single-bucket collapse to uniform utilization. To assess quantization quality, we compute the Normalized MSE, $\text{NMSE}_A(G,T) = \|Q_G(TA) - TA\|_F^2 / \|TA\|_F^2$. Setting $A = I$ and $A = \mathbf{H}_{16}$ yields the raw and post-RHT errors, respectively, allowing us to summarize the RHT-induced change via the Signal-to-Quantization-Noise Ratio:

$$\Delta\text{SQNR} = 10 \log_{10} \frac{\text{NMSE}_I}{\text{NMSE}_{\mathbf{H}_{16}}}, \tag{5}$$

where $\Delta\text{SQNR} > 0$ indicates that rotation improves tensor-level fidelity.

Figure 2 analyzes real training tensors from MLP blocks (`fwd_w`, `fwd_x`, and `bwd_dy`). As expected, RHT substantially increases B_{eff} for outlier-heavy tensors (e.g., `linear_fc1/bwd_dy` and `linear_fc2/fwd_x`), consistent with the outlier-amplifying behavior of SwiGLU (Jiang et al., 2026). Crucially, however, this improved utilization yields divergent outcomes depending on grid geometry. By shifting tensor mass from extreme tails into mid-magnitude regions, RHT inadvertently forces data into E2M1's most asymmetric rounding bins, exacerbating quantization noise ($\Delta\text{SQNR} < 0$). In contrast, the uniformly spaced E1M2 grid safely translates this flattened distribution into higher fidelity ($\Delta\text{SQNR} > 0$). This exposes a hidden pitfall: despite effectively dispersing outliers, applying RHT to non-leaf paths (`fwd_y`, `bwd_dx`) systematically degrades E2M1-based recipes.

> **Key Finding 2:** Shrinkage Bias accumulates multiplicatively across network layers, causing systematic signal decay. However, applying RHT exacerbates this decay under E2M1 by shifting tensor mass into the most biased asymmetric bins, degrading deep training stability.

Table 1: Comparison of E1M2-based recipe UFP4 with E2M1-based recipe.

Configuration	E2M1-based recipe	E1M2-based recipe (UFP4)
Format	E2M1	E1M2/INT4-style uniform grid
Quant block size	1×16	1×16
Scale hierarchy	FP32 single-level	FP32 single-level
RHT scope	`bwd_dw`	`fwd_y`, `bwd_dx`, `bwd_dw`
RHT block size	16	16
SR scope	dY	dY
2D weight scaling	✗	✗

4 UFP4: 4-BIT TRAINING RECIPE WITH UNIFORM GRIDS

Most existing FP4 training recipes default to the E2M1 format (NVIDIA et al., 2026; Chen et al., 2026; Panferov et al., 2026). While initially attractive for raw outlier-heavy tensors, current mitigation techniques (e.g., specialized rounding, scaling, or tensor preprocessing) merely treat the symptoms of *Shrinkage Bias*. Because they leave the underlying E2M1 grid geometry unchanged, they fail to eliminate its format-level source.

UFP4 Recipe. Our prior analysis yields a clear design principle: once RHT shifts tensors from a dynamic-range-limited to a local-resolution-limited regime, the 4-bit grid must prioritize local magnitude preservation over extreme dynamic range. Therefore, we propose UFP4, a uniform 4-bit training recipe based on an E1M2/INT4-style uniform grid (Figure 3). For every linear-layer GEMM, UFP4 applies RHT and quantizes the operands to this uniformly spaced grid. Crucially, while E2M1 recipes typically restrict RHT to the weight-gradient path to avoid compounding geometric errors, the unbiased nature of UFP4 allows us to safely enable RHT across all three GEMMs: FPROP (`fwd_y`), DGRAD (`bwd_dx`), and WGRAD (`bwd_dw`). Stochastic rounding (SR) is applied to dY to preserve gradient expectations. Scale hierarchy design remains orthogonal to this format-level solution; one may use single-level, two-level, or block scales depending on hardware efficiency. Our broader recommendation is that future ML accelerators should elevate E1M2/INT4-style uniform grids to first-class training primitives, challenging the status quo that treats E2M1 as the sole viable FP4 format.

Comparison with E2M1-based Recipe. As shown in Table 1, we keep auxiliary configurations identical across recipes, including block size, scale hierarchy, and SR scope. The methodology differs in only two structural choices: UFP4 employs a uniform grid to eliminate geometric Shrinkage Bias, which, in turn, permits extending RHT coverage from a single GEMM path (`bwd_dw`) to all three (`fwd_y`, `bwd_dx`, `bwd_dw`). This design directly isolates the impact of the grid format and RHT scope on training stability.

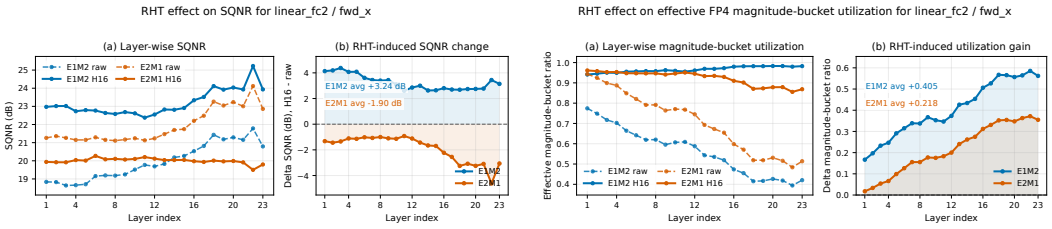
5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Long-run models are Dense 1.5B (a variant of Ling-mini-2.0), Ling-3.0-tiny (MoE 7.9B, 1.3B active), and Ling-3.0-flash (MoE 124B, 5.1B active); Appendix A gives their architectures and training settings. Within each experimental family, compared runs share data, architecture, optimizer, learning-rate schedule, batch construction, evaluation cadence, and training horizon. For end-to-end FP4 training, we compare the E1M2-based UFP4 recipe with a strong E2M1 reference selected by a controlled configuration search (Appendix C); Table 1 summarizes the resulting recipes. We match auxiliary choices such as block size, scale hierarchy, and SR scope whenever possible, so the comparison focuses on the joint effect of FP4 grid choice and RHT scope.

5.2 Q1: DOES RHT CHANGE THE PREFERRED 4-BIT GRID IN BOTH TENSOR QUANTIZATION AND GEMM OUTPUTS?

We first test the interaction between FP4 grid geometry and RHT before end-to-end training. Using MLP tensors collected from real training, we compare E2M1 and E1M2 with and without block RHT at two levels: single-tensor quantization and single-GEMM output.



(a) SQNR. (b) Effective bucket ratio.

Figure 4: Single-tensor quantization diagnostics for the outlier-heavy `linear_fc2/fwd_x` tensors. RHT reverses the SQNR ordering: E2M1 is better before rotation, while E1M2 is better after rotation.

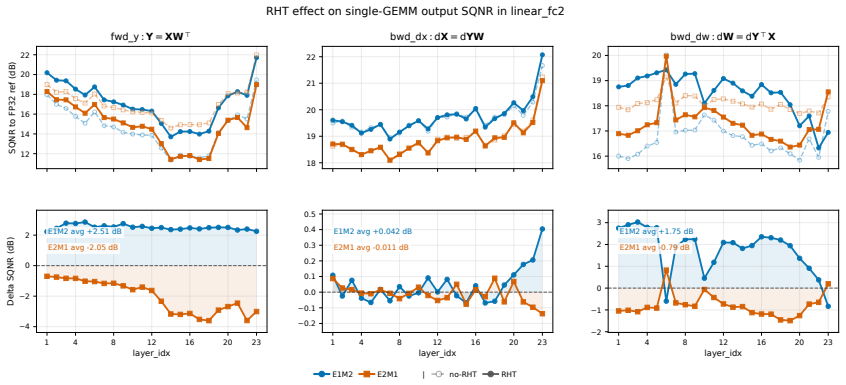


Figure 5: Single-GEMM output SQNR for `linear_fc2`. The format-dependent inversion is more pronounced than in `linear_fc1`. RHT improves E1M2 output SQNR on the dominant paths, but often decreases E2M1 output SQNR after the operands enter the post-RHT concentrated regime.

Single-tensor quantization. We report SQNR and effective bucket ratio, an entropy-based bucket-utilization metric from Section 3.2. For the well-behaved `linear_fc1/fwd_x` tensor, RHT is nearly neutral: mean ΔSQNR is -0.008 dB for E1M2 and $+0.007$ dB for E2M1, with negligible effective bucket ratio changes (-0.0008, $+0.0001$; Figure 8 in Appendix B). For outlier-heavy `linear_fc2/fwd_x`, RHT reverses the format ranking: E2M1 leads before rotation (21.90 vs. 19.94 dB), whereas E1M2 leads after rotation (23.19 vs. 20.00 dB) and raises effective bucket ratio from 0.56 to 0.97 on average (Figure 4). Thus, RHT is not universally beneficial; by shifting outlier-heavy tensors from a dynamic-range-limited to a local-resolution-limited regime, it changes the preferred FP4 grid. Extended tensor- and GEMM-level diagnostics across MLP and attention layers are summarized in Appendix B.

Single-GEMM output. The same pattern survives GEMM composition. We measure output SQNR for `fwd_y`, `bwd_dx`, and `bwd_dw` using the same collected tensors. For `linear_fc1`, RHT is relatively benign on the forward GEMM but exposes format-dependent degradation on backward paths: E1M2 preserves or improves output SQNR, whereas E2M1 loses SQNR after rotation (Figure 8 in Appendix B). For `linear_fc2`, the inversion is stronger: E1M2 converts post-RHT bucket utilization into higher output SQNR, while E2M1 often degrades after rotation (Figure 5). Therefore, the RHT-induced format inversion is not limited to single-tensor diagnostics; it persists in GEMM outputs used by the training computation graph.

5.3 Q2: Does UFP4 reduce the BF16-relative training loss gap?

We compare BF16, the E2M1 reference, and the E1M2-based UFP4 recipe on Dense 1.5B, MoE 7.9B, and MoE 124B long-run pretraining, reporting BF16-relative LM loss error $|\mathcal{L}_r - \mathcal{L}_{\text{BF16}}|/\mathcal{L}_{\text{BF16}}$. Across all three settings, UFP4 stays closer to BF16 (Figure 6): latest-1000-step relative error drops from 1.2570% to 0.9673% on Dense 1.5B, from 2.3596% to 1.8469% on MoE 7.9B, and from 1.7308% to 1.3863% on MoE 124B. Thus, the local interaction between FP4 grid geometry and RHT scope observed in Q1 persists in long-run training, although both FP4 recipes still incur a measurable BF16 gap.

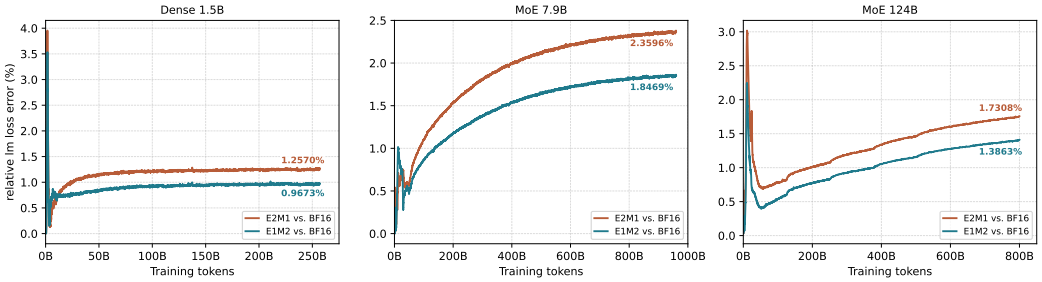
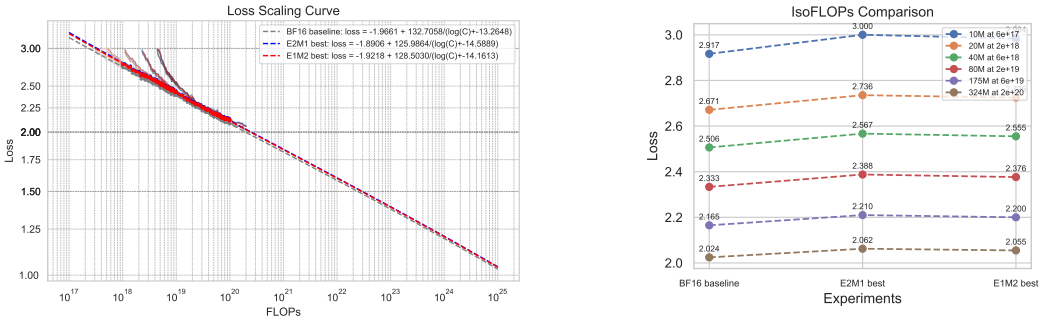


Figure 6: The E1M2-based UFP4 recipe stays closer to BF16 over long-run training. Each panel reports the BF16-relative language-modeling loss error $|\mathcal{L}_r - \mathcal{L}_{\mathrm{BF16}}|/\mathcal{L}_{\mathrm{BF16}}$ for $r \in \{\mathrm{E2M1}, \mathrm{E1M2}\}$. Lower is better.



(a) Scaling-law fit and extrapolation.

(b) Loss comparison across model sizes.

Figure 7: Scaling-law validation of the E1M2 recipe (UFP4) against the E2M1 reference. The fitted curve summarizes the trend across the scaling sweep, while the loss comparison shows the observed gap at the measured model sizes.

5.4 Q3: Does the advantage persist across model scales?

Following the scaling-law protocol of Tian et al. (2026), we train 10M–324M MoE models under matched data, optimization, and compute budgets, and fit loss as a function of training compute. Across the measured range and fitted extrapolation, the E1M2 curve remains below the E2M1 reference (Figure 7), indicating that the advantage is not confined to the smallest models. The fitted FP4-to-BF16 gap also decreases with compute, suggesting that the FP4 penalty does not grow over the scaling sweep, although a residual gap to BF16 remains.

5.5 Q4: Which recipe components matter, and can E2M1 approximate a uniform grid?

RHT scope and stochastic rounding. We ablate RHT scope and SR on Dense 1.5B E1M2 FP4 runs trained beyond 100B tokens (Table 2). Full-RHT coverage is best: relative to no RHT, `bwd_dw`-only RHT reduces loss by 0.00481, `fwd_y`+`bwd_dw` by 0.00644, `bwd_dx`+`bwd_dw` by 0.00290, and full RHT by 0.01123. With full RHT fixed, SR on dY gives another 0.00456 reduction. Thus, both full RHT and SR contribute. This result is notable because prior NVFP4 recipes typically avoid rotating non-leaf GEMM outputs such as `fwd_y` and `bwd_dx`, where RHT can be harmful under E2M1-style grids (NVIDIA et al., 2026; Chen et al., 2026); our ablation supports the mechanism: once the post-RHT regime is represented by a uniform grid, full-RHT coverage becomes beneficial in this setting.

Can range-restricted E2M1 emulate a uniform grid? Current FP4 hardware paths may be tied to E2M1/NVFP4-style data elements, so we test whether range-restricted E2M1 can emulate uniform-grid behavior. The E2M1 reference uses `max_fpx` $= 6$ and RHT only on `bwd_dw`; full-RHT variants reduce `max_fpx`, with `max_fpx` $= 2.0$ retaining only $\{0, 0.5, 1.0, 1.5, 2.0\}$. Although this avoids high-magnitude asymmetric bins, it also sacrifices dynamic range and bucket utilization: all tested range-restricted variants underperform the E2M1 reference on Dense 1.5B and MoE 7.9B

Table 2: Ablation study of RHT scope and SR under FP4 E1M2 training. For the RHT ablation block, Δ is measured against SR enabled with no RHT. For the SR ablation block, Δ is measured against the full-RHT, no-SR variant.

Setting	SR	`fwd_y`	`bwd_dx`	`bwd_dw`	Mean LM loss	Δ loss
RHT scope ablation						
No RHT	✓	–	–	–	1.89202	0.00000
RHT on `bwd_dw`	✓	–	–	✓	1.88721	-0.00481
RHT on `bwd_dx`, `bwd_dw`	✓	–	✓	✓	1.88912	-0.00290
RHT on `fwd_y`, `bwd_dw`	✓	✓	–	✓	1.88558	-0.00644
Full RHT w/ SR (UFP4)	✓	✓	✓	✓	**1.88079**	**-0.01123**
SR ablation under full RHT						
Full RHT w/o SR	–	✓	✓	✓	1.88535	0.00000
Full RHT w/ SR (UFP4)	✓	✓	✓	✓	**1.88079**	**-0.00456**

(Figure 9; full sweep and hardware discussion in Appendix D.1). In these tested recipes, E2M1 range restriction is therefore not a satisfactory substitute for native E1M2/INT4 support.

5.6 Q5: Can RHT be fused into FP4 quantization efficiently?

Full RHT adds only a small block Hadamard transform before FP4 quantization. When the Hadamard block matches the quantization block, the transform can be fused before scale estimation and packing, avoiding an intermediate rotated tensor. For block size 16, the fused RHT +quantization is about $1.06\times$ and $1.07\times$ the latency of standalone quantization across tested BF16 matrix shapes on SM90 and SM100, respectively, whereas unfused RHT+quantization has $1.62\times$ and $1.41\times$ the fused latency (Figure 10 in Appendix D.2). Thus, full-RHT quantization has small fused-kernel overhead, although end-to-end training overhead still depends on full-system integration.

6 Related Work

MXFP4 and NVFP4 use fine-grained block scaling, and NVFP4 further uses smaller blocks and a two-level scale hierarchy to improve E2M1 training accuracy (Rouhani et al., 2023b; NVIDIA et al., 2026). A second line modifies the quantizer or training estimator to make FP4 training more stable. Rotation-based methods such as RHT, QuaRot, SpinQuant, and FlatQuant disperse outlier energy before quantization (Ashkboos et al., 2024; Liu et al., 2025; Sun et al., 2025). In this paper, we show that a post-RHT tensor may still suffer from *Shrinkage Bias* when quantized to non-uniform E2M1. This limitation is imposed by the format grid itself. Therefore, under E1M2/INT4-style uniform formats, the same tensor-side preprocessing methods may be better able to translate improved tensor distributions into quantization-quality gains (see Appendix F).

7 Conclusion

We revisit the default use of E2M1 for FP4 training under RHT-based outlier mitigation. Our analysis shows that RHT changes the quantization regime from dynamic-range-limited to local-resolution-limited, where the asymmetric RTNE bins of E2M1 can introduce Shrinkage Bias. This effect is visible in real tensor and GEMM diagnostics, and the same format-dependent pattern persists in long-run dense and MoE training. By using an E1M2/INT4-style uniform grid, applying RHT to all three training GEMMs, and using stochastic rounding only on dY, UFP4 consistently reduces the BF16-relative FP4 loss error compared with the E2M1 reference recipe.

Our implication is narrow but actionable: E2M1 should remain available for range-limited workloads, but it should not be the only first-class FP4 training format. Future training accelerators should support E1M2/INT4-style uniform 4-bit grids as first-class training data elements, enabling recipes such as UFP4 to combine post-RHT numerical stability with native 4-bit matrix throughput.

AI USE STATEMENT

Generative AI tools were used to improve the wording of selected paragraphs and assist in creating some figures. They were not used to formulate research questions or hypotheses, develop mathematical claims or proofs, design methods or experiments, implement methods, generate or process data, analyze or interpret results, or translate material. The authors checked the technical accuracy of the edited paragraphs and verified the AI-assisted figures against the underlying data, numerical values, axis labels, and captions. The authors take responsibility for the final content of this paper.

REFERENCES

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: outlier-free 4-bit inference in rotated llms. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Hengjie Cao, Mengyi Chen, Yifeng Yang, Fang Dong, Ruijun Huang, Jixian Zhou, Anrui Chen, Mingzhi Dong, Yujiang Wang, Jinlong Hou, Yuan Cheng, FAN WU, Fan Yang, Tun Lu, Ning Gu, and Li Shang. Metis: Training LLMs with FP4 quantization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=I2ZrCi5O84.

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.

Brian Chmiel, Maxim Fishman, Ron Banner, and Daniel Soudry. FP4 all the way: Fully quantized training of large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=kuzye4EPLR.

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.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bing-Li Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dong-Li Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, JingChang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Jun-Mei Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Min Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, Rui-Qi Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shao-Kang Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, Weizhen Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wen xuan Yu, Wentao Zhang, X. Q. Li, Xiangyu Jin, Xianzu Wang, Xiaoling Bi, Xiaodong Liu, Xiaohan Wang, Xi-Cheng Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yao Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yi Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng

Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxiang Ma, Yuting Yan, Yu-Wei Luo, Yu mei You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Zehui Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhen guo Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Ziling Li, Ziwei Xie, Zi-Han Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2024. URL https://arxiv.org/abs/2412.19437. arXiv:2412.19437.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1737–1746, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/gupta15.html.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

inclusionAI. Ling-3.0-flash-base model card, 2026a. URL https://huggingface.co/inclusionAI/Ling-3.0-flash-base.

inclusionAI. Ling-3.0-tiny-base model card, 2026b. URL https://huggingface.co/inclusionAI/Ling-3.0-tiny-base.

Peijie Jiang, Yuqi Feng, Cunyin Peng, Qian Zhao, Jia Liu, Kunlong Chen, Zhiqiang Zhang, and Jun Zhou. Powlu: An activation function for stable pre-training of llms, 2026. URL https://api.semanticscholar.org/CorpusID:288669934.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Hanglin Li, Shuchang Tian, Chen Lin, Zhiyong Zhao, and Kun Zhan. Faar: Format-aware adaptive rounding for nvfp4, 2026. URL https://arxiv.org/abs/2603.22370.

Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank component for 4-bit diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ogO6DGE6FZ.

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.

Andrei Panferov, Erik Schultheis, Soroush Tabesh, and Dan Alistarh. Quartet ii: Accurate llm pre-training in nvfp4 by improved unbiased gradient estimation, 2026. URL https://arxiv.org/abs/2601.22813.

Bita Darvish Rouhani, Ritchie Zhao, Ankit More, Mathew Hall, Alireza Khodamoradi, Summer Deng, Dhruv Choudhary, Marius Cornea, Eric Dellinger, Kristof Denolf, Stosic Dusan, Venmugil Elango, Maximilian Golub, Alexander Heinecke, Phil James-Roxby, Dharmesh Jani, Gaurav Kolhe, Martin Langhammer, Ada Li, Levi Melnick, Maral Mesmakhosroshahi, Andres Rodriguez, Michael Schulte, Rasoul Shafipour, Lei Shao, Michael Siu, Pradeep Dubey, Paulius Micikevicius, Maxim Naumov, Colin Verrilli, Ralph Wittig, Doug Burger, and Eric Chung. Microscaling data formats for deep learning, 2023a. URL https://arxiv.org/abs/2310.10537.

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

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, et al. Flatquant: Flatness matters for llm quantization. In *International Conference on Machine Learning*, pp. 57587–57613. PMLR, 2025.

Mehran Taghian, Yunke Peng, Xing Huang, Yao Wang, Yaoyuan Wang, Wei Guo, Yuanyong Luo, Tianchi Hu, Junsong Wang, Xin Wang, Hu Liu, Yu Cheng, Ziwei Yu, Hongliang Li, Mehdi Rahimifar, Lei Yan, Xuefei Wang, Zhuang Ma, Lei Liu, Hui Yu, Anandharaju Durai Raju, Hoang Le, Hei Yi Mak, Tanzila Rahman, and Shadan Golestan. Hifloat4 format for language model pre-training on ascend npus, 2026. URL https://arxiv.org/abs/2604.08826.

Ling Team, Ang Li, Ben Liu, Binbin Hu, Bing Li, Bi Zeng, Borui Ye, Caizhi Tang, Changxin Tian, Chao Huang, Chao Zhang, Chen Qian, Chenchen Ju, Chenchen Li, Chengfu Tang, Chilin Fu, C. Ren, Chunwei Wu, Cong Zhang, Cunyin Peng, Dafeng Xu, Daixin Wang, Dalong Zhang, Dingnan Jin, Dingyuan Zhu, Di Hu, Fa-Chang Zhao, Feifan Wu, Feng Zhu, Gangshan Wang, Haitao Zhang, Haili Zhao, Han-Shu Zhang, Hanzi Wang, Hao-Yang Qian, Hao Yu, Heng Zhang, Hongliang Zhang, Hongzhi Luan, Hui-Jun Dong, Huizhong Li, Jia Li, Jia Liu, Jialong Zhu, Jian-Jun Sha, Jianping Wei, Jiaolong Yang, Jie Ma, Jiewei Wu, Jinjing Huang, Jin-Shou Tian, Jingyuan Zhang, Jin ling Sun, Juan Tu, Jun Liu, Jun Xu, Jun Zhou, Junjie Ou, Junpeng Fang, Kai-Guang Zhang, Kaiqin Hu, Ke Shi, Kun Tang, Kunlong Chen, Lanyin Mei, Lei Liang, Lei Xu, Libo Zhang, Lin Ju, Lin Yuan, Ling-Tao Zhong, Lintao Ma, Lu Liu, Lu Yu, Lun Cai, Mei qi Zhu, Mengying Li, Min Chen, Minghao Xue, Mi Cai, Ming Yin, Peijie Jiang, Peilong Zhao, Pingping Liu, Qian Zhao, Qing Cui, Qin Huang, Qingyuan Yang, Quan Yu, Shaowei Wei, Shijie Lian, Shoujian Zheng, Shun Song, Shun-Yao Zhang, Shuo Zhang, Siyuan Li, Song Liu, Ting Guo, Tong Zhao, Wanli Gu, Weichang Wu, Wei Han, Wen-Qian Fang, Wubin Wang, Xiang Shu, Xiaowei Shi, Xi Lan, Xiaolu Zhang, Xiaqin Sun, Xin Zhao, Xingyu Lu, Xiong Xu, Xudong Wang, Xue-Jun Yang, Yajie Yang, Yang Xiang, Yanzhe Li, Yi Zhang, Yilong Wang, Yingxue Li, Yong-Chang Guo, Yuzhuo Fu, Yuanyuan Wang, Yue Yang, Yue Yu, Yufeng Deng, Yun Zhang, Yunfei Yu, Yuqi Zhang, Yuxiao He, Z. Gui, Zhaoxin Huan, ZhaoYang Wang, Zhibo Zhu, Zhihao Wang, Zhiqiang Zhang, Zhou Wang, Zi-Ying Zeng, Ziqi Liu, Zitao Xuan, and Zuoli Tang. Every activation boosted: Scaling general reasoner to 1 trillion open language foundation, 2025. URL https://arxiv.org/abs/2510.22115. arXiv:2510.22115.

Changxin Tian, Kunlong Chen, Jia Liu, Ziqi Liu, Zhiqiang Zhang, and JUN ZHOU. Towards greater leverage: Scaling laws for efficient mixture-of-experts language models. In *The Fourteenth*

International Conference on Learning Representations, 2026. URL https://openreview.net/forum?id=7r2lkhDGUj.

Ruizhe Wang, Yeyun Gong, Xiao Liu, Guoshuai Zhao, Ziyue Yang, Baining Guo, Zheng-Jun Zha, and Peng CHENG. Optimizing large language model training using FP4 quantization. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=uK7JArZEJM.

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.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Zican Dong, Yupeng Hou, Beichen Zhang, Yingqian Min, Junjie Zhang, Peiyu Liu, et al. A survey of large language models. *Frontiers of Computer Science*, 20(12):2012627, 2026.

Jiaxiang Zou, Yonghao Chen, Ruilong Wu, and Xinyu Chen. Mixfp4: Enhancing nvfp4 with adaptive fp4/int4 block representations, 2026. URL https://arxiv.org/abs/2605.31035.

A Model Architectures and Training Hyperparameters

Dense 1.5B is a variant of Ling-mini-2.0 (Team et al., 2025). It uses GQA with 4 query groups and RoPE, SwiGLU, RMSNorm, and the Adam optimizer (Kingma & Ba, 2015). Ling-3.0-tiny and Ling-3.0-flash use KDA and Gated MLA attention (inclusionAI, 2026b;a) with the Muon optimizer (Jordan et al., 2024). Table 3 compares the three models; dashes indicate fields inapplicable to the dense model.

Table 3: Architecture and training settings.

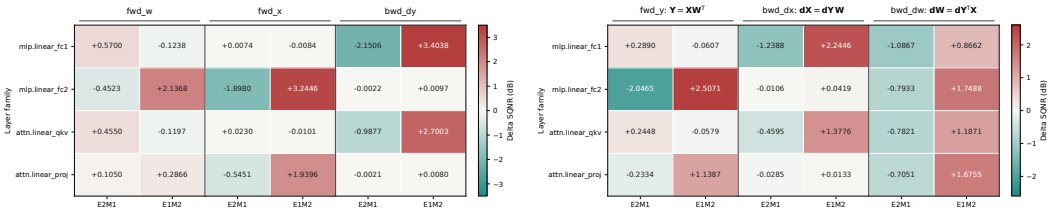
Setting	Dense 1.5B	Ling-3.0-tiny	Ling-3.0-flash
Total parameters	1.5B	7.9B	124B
Active parameters	1.5B	1.3B	5.1B
Layers	20	24	42
Attention blocks	20 GQA	KDA:MLA=3:1	KDA:MLA=5:1
Attention heads	16	16	32
Attention head dimension	128	128	128
Hidden size	2,048	1,536	2,560
Dense FFN layers	20	1	2
Dense FFN hidden size	5,120	4,608	6,144
Routed experts	–	128	512
Shared experts	–	1	1
Active experts	–	8	8
Expert FFN hidden size	–	512	768
Vocabulary size	157,184	157,184	157,184
Sequence length	4,096	8,192	8,192
Global batch size	1,024	8,192	7,680
Base learning rate	5.54×10^{-4}	3.39×10^{-4}	2.12×10^{-4}
Weight decay	0.1	0.1	0.1
Optimizer	Adam	Muon	Muon

B RHT-Induced SQNR Changes in Tensors and GEMM Outputs

The main text uses layerwise MLP traces to explain how RHT changes the preferred FP4 grid. Here we aggregate the same $\Delta\mathrm{SQNR}$ diagnostic over MLP and attention layer families, covering both single-tensor quantization and single-GEMM outputs.

Figure 8 gives two takeaways. First, the RHT effect is format-dependent: E1M2 produces large gains on rotation-sensitive tensors (e.g., `linear_fc2/fwd_x`) and remains near-neutral on already well-behaved ones (e.g., `linear_fc1/fwd_x`), whereas E2M1 often loses SQNR on these rotation-sensitive tensors. This is the empirical signature expected from the shrinkage-bias mechanism: after RHT, local resolution matters more than excess dynamic range.

Second, the same format-ranking inversion appears in several GEMM outputs and extends to attention modules. This is not a blanket failure of RHT under E2M1: some E2M1 FPROP outputs show small SQNR gains, such as `linear_fc1` and `linear_qkv`. Rather, E2M1 degradation is concentrated on rotation-sensitive paths, including `linear_fc2` FPROP (`fwd_y`) and attention backward GEMMs, whereas E1M2 yields positive $\Delta\mathrm{SQNR}$ on the main rotation-sensitive paths, supporting full-RHT coverage in UFP4.



(a) Single-tensor quantization. (b) Single-GEMM output.

Figure 8: Mean RHT-induced $\Delta\mathrm{SQNR}$ for tensor quantization and GEMM outputs. Rows average over 23 layers; positive/negative values indicate SQNR gain/loss after RHT.

C Controlled E2M1 Reference Selection

To avoid an undertuned E2M1 baseline, we tune its training configuration through controlled one-factor ablations over the dimensions in Table 4. Each candidate run trains Dense 1.5B for 200B tokens with identical data, architecture, optimizer, learning-rate schedule, and all non-ablated recipe settings. Within each ablation, we choose the setting with the lowest average LM loss over the final 1,000-step window. The resulting E2M1 configuration is then frozen for the long-run, scaling, and component-ablation experiments.

The selected E2M1 structure is WGRAD-only RHT (`bwd_dw`) with dY-only SR. Under E2M1, FPROP RHT (`fwd_y`) remains harmful, while DGRAD RHT (`bwd_dx`) is initially neutral but yields higher LM loss at later training stages; the selected RHT scope is therefore `bwd_dw`. For stochastic rounding, dY-only SR achieves the lowest LM loss. Adding SR to the backward GEMM operands X_{WGRAD} and W_{DGRAD} slightly increases LM loss in our E2M1 setting. This observation differs from the NVFP4 setting studied by TetraJet-v2 (Chen et al., 2026), where applying SR beyond dY to additional backward GEMM operands is reported to improve training. We therefore use the best observed E2M1 setting from these ablations as the reference configuration.

Table 4: Controlled one-factor ablation space for selecting the E2M1 reference. For a set $\mathcal{S}$, $2^{\mathcal{S}}$ denotes its power set, i.e., the set of all combinations formed from elements of $\mathcal{S}$, including the empty combination. The SR candidate set is $\mathcal{S}_{\text{SR}} = \{dY, X_{\text{FPROP}}, W_{\text{FPROP}}, X_{\text{WGRAD}}, W_{\text{DGRAD}}\}$.

Dimension	Search space	Selected settings	Notes
RHT scope	$2^{\{\texttt{fwd_y},\texttt{bwd_dx},\texttt{bwd_dw}\}}$	`bwd_dw`	WGRAD-only selected; FPROP RHT is detrimental and DGRAD RHT raises LM loss at later stages.
RHT block size	16 / 32 / 64 / 128	16	Comparable LM loss; 16 aligns with quant blocks and facilitates fused RHT+quantization.
SR scope	$2^{\mathcal{S}_{\text{SR}}}$	dY	dY-only selected; backward $X_{\text{WGRAD}}, W_{\text{DGRAD}}$ SR slightly increases LM loss.
2D weight scaling	✓ / ✗	✗	Disabled.
`max_fpx`	4 / 6	6	Maximum FP4 magnitude; 6 preserves the full E2M1 range and minimizes LM loss.

D Additional Format and Kernel Evaluations

D.1 Range-Restricted E2M1 with Full RHT

We test whether restricting E2M1's representable range can provide finer local resolution for post-RHT tensors while retaining the E2M1 data format. On Dense 1.5B and MoE 7.9B, we evaluate full-RHT variants with $\texttt{max_fpx} \in \{4, 3, 2\}$ against the selected E2M1 reference ($\texttt{max_fpx} = 6$, WGRAD-only RHT; Figure 9). The final experimental results show that every range-restricted variant has higher LM loss than the reference for both models. Restricting the range removes the upper asymmetric bins but also reduces the number of usable levels and changes block scaling. Under these tested configurations, range-restricted E2M1 with full RHT therefore does not reproduce the E1M2 recipe's benefit.

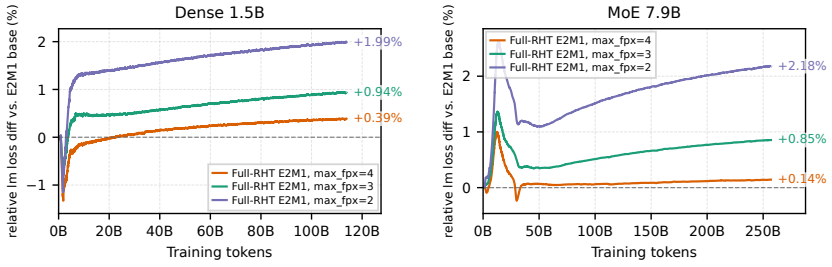


Figure 9: Relative LM loss of full-RHT E2M1 variants compared to the E2M1 reference.

D.2 Fused RHT and Quantization

To isolate the quantization-side cost of full RHT, we compare standalone quantization, fused RHT+quantization, and separate RHT followed by quantization. Figure 10 reports the aggregate and per-shape latency measurements. Fusion avoids materializing an intermediate rotated tensor and keeps the measured quantization-kernel overhead near the standalone path.

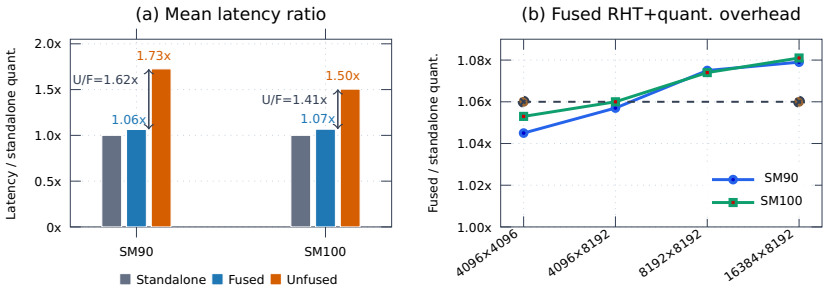


Figure 10: Relative latency of RHT+quantization. Left: mean latency ratio relative to standalone quantization; bar heights and labels are normalized by standalone quantization, while gray arrows report U/F (unfused-to-fused) ratios. Right: per-shape fused RHT+quantization overhead.

E Operand Projection and Cumulative Attenuation

The main text models persistent quantization error as coherent signal attenuation. We separate the exact single-GEMM decomposition from the additional assumption needed to compose attenuation factors along a computation path.

E.1 Operand Projection and GEMM Expansion

For a GEMM $Z = AB^\top$ with quantized operands $\widehat{A} = Q_G(A)$ and $\widehat{B} = Q_G(B)$, we measure magnitude shrinkage by projecting the quantized operands onto their exact BF16 counterparts:

$$\alpha_A = \frac{\langle \widehat{A}, A\rangle_F}{\|A\|_F^2}, \qquad \widehat{A} = \alpha_A A + R_A, \qquad \langle R_A, A\rangle_F = 0, \tag{6}$$

and analogously $\widehat{B} = \alpha_B B + R_B$. A scaling factor $\alpha_A < 1$ indicates a reduced signal component aligned with A, acting as the tensor-level manifestation of magnitude shrinkage.

Substituting these orthogonal decompositions into the GEMM yields the exact identity:

$$Z_q = \widehat{A}\widehat{B}^\top = \underbrace{\alpha_A\alpha_B AB^\top}_{\alpha_A\alpha_B Z} + \underbrace{\alpha_A AR_B^\top + \alpha_B R_A B^\top + R_A R_B^\top}_{\text{residual noise}}. \tag{7}$$

The expansion is exact; treating $\alpha_A\alpha_B$ as output attenuation assumes that the residual terms have negligible projection onto Z.

E.2 Exponential Approximation of Multiplicative Accumulation

Under the residual-incoherence approximation above, let $\eta_k \approx \alpha_{A,k}\alpha_{B,k}$ be the coherent factor of GEMM k and $\delta_k = 1 - \eta_k$. The cumulative factor in Equation (3) is then

$$\prod_{k=1}^{K} \eta_k = \prod_{k=1}^{K} (1 - \delta_k). \tag{8}$$

Taking the logarithm of the product gives the exact identity

$$\begin{aligned} \prod_{k=1}^{K} (1 - \delta_k) &= \exp\left(\log \prod_{k=1}^{K} (1 - \delta_k) \right) \\ &= \exp\left(\sum_{k=1}^{K} \log(1 - \delta_k) \right). \end{aligned} \tag{9}$$

For small per-GEMM multiplicative loss, the logarithm admits the Taylor expansion

$$\log(1 - \delta_k) = -\delta_k + O\big(\delta_k^2\big), \qquad |\delta_k| \ll 1. \tag{10}$$

Substituting this expansion into Equation (9) yields

$$\prod_{k=1}^{K} (1 - \delta_k) = \exp\left(-\sum_{k=1}^{K} \delta_k + O\left(\sum_{k=1}^{K} \delta_k^2 \right) \right). \tag{11}$$

When the second-order term $O(\sum_k \delta_k^2)$ is small relative to the first-order accumulation, we obtain the approximation

$$\prod_{k=1}^{K} (1 - \delta_k) \approx \exp\left(-\sum_{k=1}^{K} \delta_k \right). \tag{12}$$

This shows that even small per-GEMM multiplicative losses add in the exponent. Consequently, a weak but consistently positive δ_k can produce a visible cumulative decay over long computation paths.

F Related Work Details

FP4 formats, block scaling, and scale hierarchy. A first line of FP4 training work improves quantization quality by refining the data format, block size, and scale hierarchy. MXFP4 and NVFP4 use fine-grained block scaling, and NVFP4 further uses smaller blocks and a two-level scale hierarchy to improve E2M1 training accuracy (Rouhani et al., 2023b; NVIDIA et al., 2026). Recent studies show that this design is not limited to a fixed E2M1 path: HiFloat4 adopts a uniform S1P2 data element with hierarchical scales on Ascend NPUs (Taghian et al., 2026), MixFP4 adaptively chooses E2M1 or E1M2 per block within an NVFP4-compatible scale hierarchy (Zou et al., 2026), and Four Over Six improves NVFP4 by evaluating alternative block scales to better represent near-maximal values (Cook et al., 2026). INT-vs-FP comparisons further show that fine-grained INT formats can become competitive with FP formats once block size and outlier mitigation are considered (Chen et al., 2025). Together, these works show that scale hierarchy, block size, and data format design are all critical for practical FP4 training. UFP4 studies a complementary axis: under matched block size and scale hierarchy, it isolates whether post-RHT tensors should be quantized to non-uniform E2M1 or to an E1M2/INT4-style uniform grid.

Quantizer-side and training-aware methods. A second line modifies the quantizer or training estimator to make FP4 training more stable. Microsoft FP4 introduces a differentiable quantization estimator and an outlier clamping/compensation strategy to prevent activation collapse (Wang et al., 2025); Quartet II improves unbiased gradient estimation with a microscaling EDEN routine (Panferov et al., 2026); FAAR learns format-aware adaptive rounding decisions by explicitly accounting for the non-uniform E2M1 grid (Li et al., 2026); and TetraJet-v2 adds backward alignment, stochastic-rounding changes, oscillation suppression, and outlier control to NVFP4 training (Chen et al., 2026). These methods reduce estimator bias, rounding error, variance, or optimization instability, and are complementary to UFP4: improved estimators, adaptive rounding rules, or stabilizers can be combined with a uniform grid.

Tensor-side preprocessing. Another line changes tensors before quantization through mathematically equivalent rewrites. Rotation-based methods such as RHT, QuaRot, SpinQuant, and FlatQuant disperse outlier energy before quantization (Ashkboos et al., 2024; Liu et al., 2025; Sun et al., 2025). Tensor-decomposition methods reduce low-bit error by decomposing tensors before quantization; related approaches include mean–SVD-style decompositions, outlier-channel separation, and smoothing methods (Cao et al., 2026; Xiao et al., 2023; Li et al., 2025; Chen et al., 2026). These methods act upstream of the quantizer: they change the distribution presented to the format grid. In this paper, we show that a post-RHT tensor may still suffer from *Shrinkage Bias* when quantized to non-uniform E2M1. This limitation is imposed by the format grid itself. Therefore, under E1M2/INT4-style uniform formats, the same tensor-side preprocessing methods may be better able to translate improved tensor distributions into quantization-quality gains.