Finer is Better (with the Right Scaling)

Clemens Schaefer Gil Tabak
Google LLC, Mountain View, Ca
{cjsschaefer,tabakg}@google.com

***Abstract*—Microscaling is a critical technique for preserving the quality of Large Language Models (LLMs) quantized to ultra-low precision formats. Intuitively, finer block sizes should yield lower quantization error; however, a paradox recently identified by Fasoli et al. (2026) demonstrates that standard abs-max scaling can actually result in degraded model quality as block sizes shrink. In this work, we investigate the underlying mechanics of this phenomenon. We demonstrate that this degradation is not an inherent limitation of finer granularity, but is primarily driven by how elements in smaller blocks statistically cluster closer to their local block maximum, interacting poorly with the coarse subnormal E4M3 values used as scaling factors. Specifically, we show that i) preventing the scaling factor from underflowing to zero mitigates errors caused by extreme underflow, ii) targeted algorithmic interventions like the 4-over-6 methodology that give more flexibility to the choice of scaling factor resolve the paradox for larger values, and iii) a brute-force search establishes an optimal baseline, confirming that the theoretical Mean Squared Error (MSE) strictly improves with finer block sizes. Ultimately, our findings highlight a critical insight for hardware-software co-design: the block-size paradox is partially an artifact of naive scale selection. While using hierarchical scaling factors or wider formats like UE5M3 interchangeably resolves much of the quality loss, we found the 4-over-6 scale selection heuristic can even further improve quality, especially for very small block sizes. Consequently, maximizing the performance of next-generation ML accelerators will require treating silicon format specifications and software scaling algorithms as tightly coupled design choices.**

I. Background

The ubiquitous deployment of Large Language Models (LLMs) across datacenter architectures and edge devices incurs substantial computational overhead and expands the memory footprint, fundamentally constraining system throughput via both arithmetic and memory bandwidth bottlenecks. To address this, hardware vendors have begun natively supporting ultra-low precision formats (four bits or fewer) to simultaneously increase compute throughput and reduce memory pressure, as seen in NVIDIA's Blackwell and AMD's MI-series accelerators. [1], [2].

To offset quality losses associated with low bit formats, microscaling has proven to be a useful technique. The central idea is to enable chunking of the inputs into small blocks with individual scaling factors (e.g., block sizes of 16 or 32). Using microscales effectively on accelerators usually requires built-in hardware support. In common microscaling formats, the scaling factors themselves are also represented using low-precision formats (e.g., 8-bit floating-point), reducing the memory footprint and additional hardware burden of the greater number of scales. Usually, this is most powerful for very narrow formats like FP4 (E2M1), which otherwise lack the necessary dynamic range to represent a wide variety of distributional values.

MX v1.0 [3] uses power-of-two scaling factors. Given inputs V for each block, it specifies as one option the largest power-of-two less than or equal to the maximum absolute value of the input entries divided by the largest representable value in the element data type (M_{max}). Mathematically, this block scale S is calculated as:

$$S = 2^{\lfloor \log_2 \left(\frac{\max_{V_i \in V} |V_i|}{M_{\text{max}}} \right) \rfloor} \tag{1}$$

For MXFP4, the element format is E2M1, where $M_{\text{max}} = 6.0$. Constraining the scaling factors to powers of two enables a high degree of hardware efficiency. Designing other quantization recipes is an area of active research. For example, [4] suggest mapping the scaling factor to the range $[3.5, 7.0]$; because clipping in the range $[6.0, 7.0]$ is mathematically equivalent to rounding in the range $[3.0, 3.5]$, this intentional clipping prevents underflows for smaller entries, resulting in less quantization error compared to no clipping at all.

Alternative microscaling specifications explore using 8-bit floating-point representations for the block scaling factors to provide finer numerical control. For instance, recent vendor-specific designs [1], [5] utilize an E4M3 representation (with the sign bit unused) for the block scale of size 16. The inclusion of mantissa bits in the scaling factor helps preserve higher precision, especially for the largest entry in each block. In typical applications the abs-max scale may be rounded to the nearest value in the scaling factor format. Because the exponent precision of E4M3 is relatively limited, this block scale is often paired with a second (per-tensor) scaling factor (i.e., hierarchical scaling).

The allocation of exponent versus mantissa bits can significantly influence the optimal quantization recipe (the conversion of higher-precision values into quantized entries along with their associated block-level scaling factors). For example, using all 8 bits for the exponent (unsigned E8M0) lowers the precision in the representation of values but covers a vast range of magnitudes. Redistributing some bits to the mantissa (e.g., E4M3 or UE5M3) lends higher precision to the entries at the expense of reducing the dynamic range, which for some distributions may require an additional tensor-level scaling factor.

The recent manuscript [6] shows that surprisingly, when following the typical abs-max methodology, there can be a

Experiment code available at https://github.com/clee1994/finer_is_better

significant degradation in quality as the block sizes become smaller in particular when using E4M3 scaling factors. This seems paradoxical because smaller block sizes are supposed to give finer-grained control. In the remainder of this paper, we set up experiments to explain why this happens, how it can be mitigated, and explain the implications for scale factor and format selection.

II. Experimental Setup

We start by analyzing the microscale behavior for a variety of quantization techniques and several possible block sizes (4, 8, 16, 32). Unless stated otherwise, the data is quantized using a scaling factor format of E4M3 and the data is stored in FP4 (OCP E2M1) [7].

FP4 has a small set of possible values: $\pm\{0.0, 0.5, 1.0, 1.5, 2.0, 3.0, 4.0, 6.0\}$. While the values are close near zero, they become farther between 4.0 and 6.0.

We use combinations of strategies for selecting the scaling factor (discussed below).

To define abs-max scaling we first compute the scaling factor:

$$S_{\text{abs-max}}(M) = \text{Round}_{\text{SF}}\left(\frac{\max_{v\in V}|v|}{M}\right) \tag{2}$$

where the function Round_{SF} includes the rounding and clipping to the target scale format (e.g., E4M3) and M is a configurable maximum value (e.g., in the standard abs max scaling for E2M1, this would be 6.0). We assume round-to-nearest-even throughout. Abs-max scaling is often the default heuristic chosen in industry and was the primary method evaluated by [6]. As we demonstrate later, this specific choice is closely tied to the paradoxical behavior observed.

For a given scaling factor S, quantized entries are found by applying the scaling and rounding using a function quant: $v_i \mapsto \text{quant}\left(\frac{v_i}{S}\right)$; de-quantization simply multiplies by S.

Because the scaling factor format is also low-precision (FP8), extremely small inputs can result in a scaling factor that rounds to zero. Since this preserves no information about the inputs (e.g. all values are set to zero), our first proposed adjustment modifies a scaling factor S to disallow it from collapsing to zero:

$$S_{\text{prevent-zero}}(M) = \max\left(S(M), \epsilon_{\text{SF}}\right) \tag{3}$$

where ϵ_{SF} is the smallest nonzero value of the scaling factor format (for example $\epsilon_{\text{SF}} = 2^{-9}$ for E4M3).

Additionally, we provide comparisons against the setup used in the four-over-six paper [8], which considers two possibilities for the scaling factor (biggest and second biggest representable value), enabling better usage of the format for blocks that benefit more from one part of the E2M1 range versus another. Conceptually, this algorithm evaluates quantization error using both 4.0 and 6.0 as maximum bounds. By allowing the scale to adapt, it prevents the bulk of the block's elements from being forced into the wide, lossy quantization gap between 4.0 and 6.0 in the FP4 grid.

$$S_{\text{4-over-6}}(M) = \underset{M\in\{4,6\}}{\text{argmin}}\ \text{MSE}(v, S_{\text{abs-max}}(M)) \tag{4}$$

where $\text{MSE}(v, S)$ evaluates the quantization mean-squared error of the inputs v with a scaling factor S.

For our full-model evaluations we also use *hierarchical scaling* – a two-stage process where a per-tensor scale is first identified using the abs-max across the entire tensor. Then, local scaling factors are found after adjusting the inputs using the global scale. Mathematically it is analogous to the process used in [5] (Appendix B) with appropriate adjustments for other formats.

Finally, we establish an upper bound on performance via a brute-force search over possible scaling factors to minimize the MSE.

III. Results

We structure our experiments into two phases:

First, we conduct a numerical study based on sampling a normal distribution across a range of standard deviations and block sizes comparable to those explored in [6]. We sample $50,000 \times$ block_size values for each standard deviation σ from the normal distribution $\mathcal{N}(0, \sigma^2)$, where the block_size is chosen from $\{4, 8, 16, 32\}$ and σ is evaluated across 151 evenly spaced points in the interval $[0.0005, 0.05]$.

Second, we validate our findings by applying these scaling methods to the weight and activation quantization (MLP layers) of several large language models to observe their impact on downstream perplexity. In the full model experiments, we consider different combinations of prevent-zero-scales, 4-over-6, hierarchical scaling and using the UE5M3 scaling factor suggested in [6].

A. Numerical Study

Figure 1 illustrates the degradation of Mean Squared Error (MSE) when using standard abs-max scaling, comparing it to our proposed zero-prevention adjustment. We also evaluate the performance of the 4-over-6 scaling methodology against the theoretical minimum MSE achieved via a brute-force search.

The analysis reveals three distinct regions of behavior across the standard deviation (σ), denoted as A, B, and C. Under standard abs-max scaling, we observe the counter-intuitive phenomenon discovered in [6]: smaller block sizes can paradoxically result in a higher MSE.

In Region A (extremely small σ), the scaling factor frequently underflows to zero, resulting in an anomalous spike in MSE. By implementing the prevent-zero-scale method—forcing the scale to the smallest representable positive value—this localized degradation is completely eliminated.

However, in Region B (slightly larger distribution standard deviations σ), the inversion of MSE relative to block size persists even with zero-prevention. This inversion is driven by

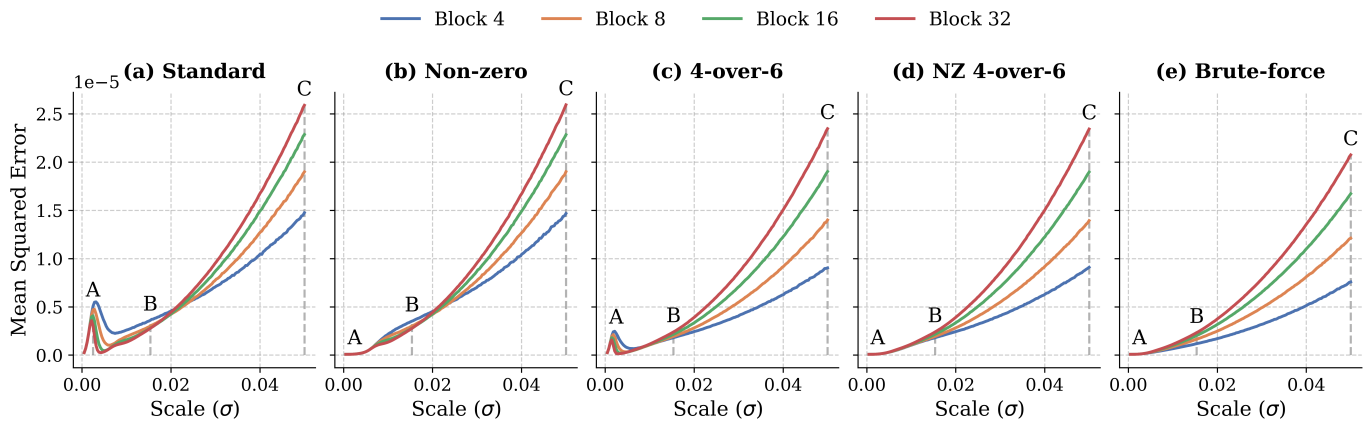


Fig. 1: Comparing MSE against distribution σ behavior. **(a)** Standard max-abs scaling exhibits a visible bump in Region A. **(b)** Setting zero scales to the smallest representable value eliminates the Region A bump, but a reverse order of block sizes remains in Region B. **(c)** The 4-over-6 methodology retains the Region A bump but resolves Region B. **(d)** Shows the combination of prevent-zero and 4-over-6. **(e)** Brute-force search represents the optimal baseline.

a destructive interaction between the E4M3 scaling format and the statistics of small blocks. For values at this magnitude, the required E4M3 scaling factors fall into the subnormal range, making the scaling grid extremely coarse. Consequently, a block's true maximum value aligns poorly with the available scaling factors. When this coarse, poorly aligned scale is applied, it disproportionately impacts smaller blocks: because the maximum value in a block of 4 is statistically closer to the rest of the elements than in a block of 32, a much larger proportion of the elements are forced into the upper extremes of the FP4 format (near 6.0). Because the FP4 quantization grid is coarsest at these extremes, the result is massive clipping and rounding error. For more details about these mechanics see Appendix A.

This inversion vanishes by employing the 4-over-6 methodology. By adaptively evaluating both 4 and 6 as possible target scales, the 4-over-6 method gives the option to restrict values to the more uniform portion of the FP4 range, and to avoid strong clipping.

Finally, the brute-force search establishes an optimal baseline, demonstrating that when the ideal scaling factor is selected, the paradoxical inversion behavior disappears entirely. Across all points A, B, and C, the optimal MSE monotonically increases with larger block sizes, confirming that finer granularity fundamentally improves quantization quality when paired with an optimal scaling strategy.

At Region C (somewhat larger values) we never see the inversion–however, 4-over-6 and the brute force methods have significantly lower MSE than the abs-max based methods especially at smaller block sizes (explained similarly by the large potion of entries at the bins of magnitude 6).

B. Downstream LLM Perplexity

To confirm that the theoretical MSE improvements translate to practical applications, we evaluated the perplexity gap of four distinct models—Granite-3.3-8B, Llama-3.1-8B, DeepSeek-LLM-7B-Base, and Qwen2.5-14B—across multiple block sizes, comparing various FP4 quantization formats against an unquantized baseline (Figure 2).

A key observation from these evaluations is a stark dichotomy in model behavior under abs-max quantization. DeepSeek and Qwen behave intuitively: as the block size decreases (offering finer-grained scaling), the perplexity gap narrows, indicating improved quality. Conversely, Granite and Llama clearly exhibit the paradoxical degradation identified in our numerical study and [6]; for these models, standard E4M3 scaling results in a sharply increased perplexity gap at smaller block sizes, most notably at block sizes 4 and 8.

By systematically evaluating different scaling configurations, we demonstrate how this non-intuitive behavior can be successfully mitigated. Isolating our proposed prevent-zero-scale (PZ) adjustment reveals noticeable but localized improvements; for instance, at the problematic block size of 4, the PZ adjustment reduces Granite's perplexity gap from 0.865 to 0.769, and Llama's from 0.725 to 0.591. However, we find that the 4-over-6 methodology [8] can have significantly greater impact on reducing the perplexity gap. Applying 4-over-6 alone reduces the perplexity gap by more than half for the ill-behaved models at the finest granularity, dropping Llama to 0.270 and Granite to 0.434 at block size 4. While the PZ adjustment does not resolve the U-shaped curve seen for Granite and Llama, 4-over-6 does.

Additionally, our results help discern the impact of hierarchical scaling and the 4-over-6 method and compare them to changing the scale format to UE5M3. Specifically, using E4M3 with a hierarchical scale (`e4m3 + H`) yields comparable results to using UE5M3, and this change is synergetic with including 4-over-6 scale selection–that is, using E4M3 with both hierarchical scales and 4-over-6 (`e4m3 + 4o6 + H`) yields better results, and is comparable to using UE5M3 with

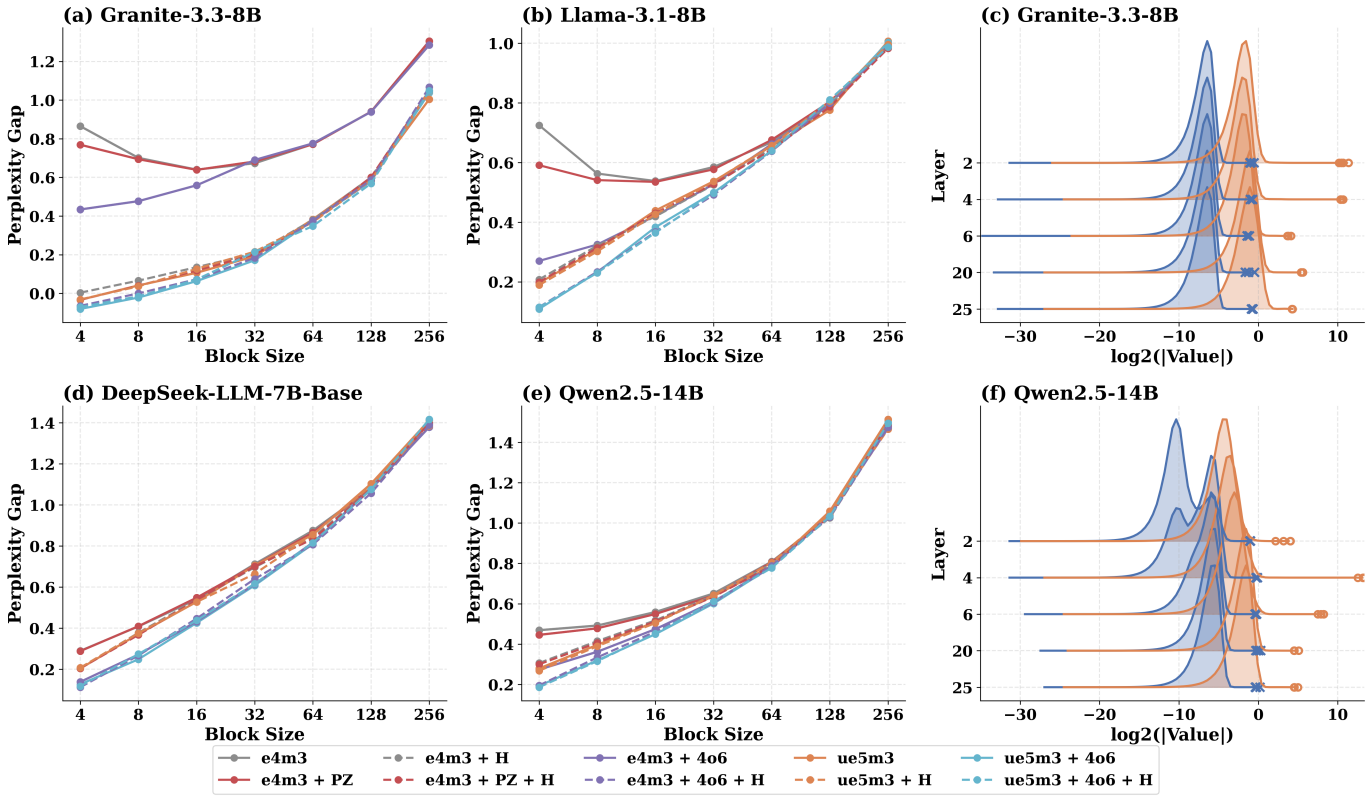


Fig. 2: **(a,b,d,e)** show the perplexity gap of the Granite-3.3-8B, Llama-3.1-8B, DeepSeek-7B, and Qwen2.5-14B models with various FP4 formats compared to a unquantized baseline. Note that only Granite and Llama demonstrate the paradoxical increased perplexity gap at lower block sizes, meanwhile DeepSeek and Qwen improve the perplexity with smaller block sizes. Techniques such as hierarchical scales (H), four over six (`4o6`) [8], and more exponent bits (`ue5m3`) fully eliminate the ill behaved Granite and Llama at smaller block sizes. **(c,f)** show representative activation and weight distribution of Granite and Qwen for layers 2, 4, 6, 20, and 25, where blue represents the weights and orange the activations.

4-over-6 (`ue5m3 + 4o6`). Finally, incorporating hierarchical scaling with UE5M3 regardless of whether using 4-over-6 does not make an impact beyond simply UE5M3 (`ue5m3` vs. `ue5m3 + H` and `ue5m3 + 4o6` vs. `ue5m3 + 4o6 + H`). These results suggest that using either E4M3 with hierarchical scaling or UE5M3 are sufficient for covering the important range of the data, **although there is an additional separate impact from the scale selection technique**.

Finally, we examine the raw model weight and activation distributions to understand this divergence in model behavior. Figure 2 (c, f) highlights these distributions for Granite and Qwen. Despite a large portion of Qwen's weights having small magnitudes, it is relatively robust to E4M3 scaling variants. To understand why Granite degrades while Qwen improves, we conducted an ablation study masking specific magnitude intervals to zero (see Appendix B, Fig. 10). We found that Qwen's performance relies heavily on larger magnitude weights, shielding it from the coarse E4M3 subnormal gap. Conversely, Granite relies heavily on smaller values that are effectively destroyed by coarse upper bins. This ablation acts as a pre-quantization diagnostic: rather than relying solely on raw tensor distributions, masking specific magnitude intervals can successfully predict a model's sensitivity to specific entries.

Adopting `ue5m3` instead inherently reduces this degradation by offering a wider absolute range—often circumventing the need for an additional tensor-level scale.

Hardware Tradeoffs: Vendors must weigh the physical silicon area costs of supporting a 5-bit exponent in hardware ALUs (UE5M3) against the memory bandwidth and latency overheads of fetching a secondary, tensor-level scale from memory (hierarchical scaling). In our experiments, both approaches yield similar quality improvements, though UE5M3 offers greater general flexibility. Furthermore, while the 4-over-6 algorithm significantly improves quality, it incurs substantial computational overhead by requiring two quantizations and MSE calculations per block. This is reasonable for inference weights (which are quantized offline), but the latency overhead may be prohibitive for dynamic activation quantization.

While our evaluation spans models from 7B to 14B parameters—a substantial and widely deployed scale—this places our study right at the critical threshold of outlier emergence. Prior literature demonstrates that extreme activation outliers system-

atically appear as models cross the 6.7B and 13B parameter boundaries, escalating in larger architectures [9]. Because these massive activations grow with the model size [10], the heavy-tailed distributions that disrupt the FP4 quantization grid are likely to become even more pronounced in 70B+ regimes. Additionally, recent work has uncovered intra-level sparsity [11], which may also interact with low-precision quantization recipes. Verifying how scaling heuristics hold up against these aggressive outliers and sparse regimes at massive scales remains an important direction for future research.

IV. Conclusion

This work investigates the paradoxical degradation of quantization quality at finer microscaling block sizes under standard abs-max scaling. Through numerical analysis and downstream LLM evaluation, we demonstrate that this anomaly is not an inherent limitation of finer granularity, but rather a consequence of how elements in smaller blocks statistically cluster closer to the per-block maximum, interacting harmfully with the coarse upper quantization bins of the FP4 (E2M1) element format.

While enforcing a strictly positive minimum scale (prevent-zero) acts as a computationally lightweight mitigation for extreme underflow, it is insufficient to resolve the structural paradox for slightly larger values. Resolving the paradox on real, deployed models requires targeted algorithmic interventions, such as the 4-over-6 strategy, to prevent elements from being forced into wide quantization gaps.

Crucially, our brute-force baseline reveals that the block-size paradox is entirely an artifact of the scale-selection heuristic, not a limitation of the FP4 format itself. This highlights a critical gap in current hardware-software co-design: while vendors are rapidly adopting ultra-low precision formats, the standard arithmetic heuristics (like abs-max) used to map values into these formats leave significant theoretical performance on the table. Unlocking the true potential of microscaling will require moving beyond naive maximums and developing silicon-friendly heuristics that approximate optimal scale selection without incurring prohibitive computational latency.

Acknowledgment

The authors would like to thank Denali Molitor for careful review of the manuscript.

References

[1] NVIDIA Corporation, "Introducing NVFP4 for efficient and accurate low-precision inference," https://developer.nvidia.com/blog/introducing-nvfp4-for-efficient-and-accurate-low-precision-inference/, 2025, accessed: 2026-04-23.

[2] Advanced Micro Devices, Inc., "AMD Instinct MI300 series accelerators and the era of adaptive microscaling," https://www.amd.com/en/products/accelerators/instinct.html, 2024, accessed: 2026-04-23.

[3] B. D. Rouhani, N. Garegrat, T. Savell, A. More, K.-N. Han, R. Zhao, M. Hall, J. Klar, E. Chung, Y. Yu, M. Schulte, R. Wittig, I. Bratt, N. Stephens, J. Milanovic, J. Brothers, P. Dubey, M. Cornea, A. Heinecke, A. Rodriguez, M. Langhammer, S. Deng, M. Naumov, P. Micikevicius, M. Siu, and C. Verrilli, "Open compute project ocp microscaling formats (mx) specification," Open Compute Project, Tech. Rep., September 2023. [Online]. Available: https://www.opencompute.org/documents/ocp-microscaling-formats-mx-v1-0-spec-final-pdf

[4] J. Chhugani, G. Jeong, B.-Y. Su, Y. Pan, H. Yang, A. Ankit, J. Yu, S. Deng, Y. Chen, N. Satish *et al.*, "Unveiling the potential of quantization with mxfp4: Strategies for quantization error reduction," *arXiv preprint arXiv:2603.08713*, 2026.

[5] F. Abecassis, A. Agrusa, D. Ahn, J. Alben, S. Alborghetti, M. Andersch, S. Arayandi, A. Bjorlin, A. Blakeman, E. Briones *et al.*, "Pretraining large language models with nvfp4," *arXiv preprint arXiv:2509.25149*, 2025.

[6] A. Fasoli, M. Kar, C.-C. Liu, S. Venkataramani, V. Srinivasan, L. Chang, and N. Wang, "Is finer better? the limits of microscaling formats in large language models," *arXiv preprint arXiv:2601.19026*, 2026.

[7] P. Micikevicius, S. Oberman, P. Dubey, M. Cornea, A. Rodriguez, I. Bratt *et al.*, "OCP 8-bit Floating Point Specification (OFP8) Revision 1.0," Open Compute Project, Tech. Rep., December 2023. [Online]. Available: https://www.opencompute.org/documents/ocp-8-bit-floating-point-specification-ofp8-revision-1-0-2023-12-01-pdf-1

[8] J. Cook, J. Guo, G. Xiao, Y. Lin, and S. Han, "Four over six: More accurate nvfp4 quantization with adaptive block scaling," *arXiv preprint arXiv:2512.02010*, 2025.

[9] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, "Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale," *Advances in neural information processing systems*, vol. 35, pp. 30 318–30 332, 2022.

[10] M. Sun, X. Chen, J. Z. Kolter, and Z. Liu, "Massive activations in large language models," *arXiv preprint arXiv:2402.17762*, 2024.

[11] J. Park, S. Kim, Z. Gu, I. Stoica, and A. Cheung, "Uncovering intra-expert activation sparsity for efficient mixture-of-expert model execution," *arXiv preprint arXiv:2605.08575*, 2026.

Appendix

A. Entry and Scale Histograms for Normal Distributions

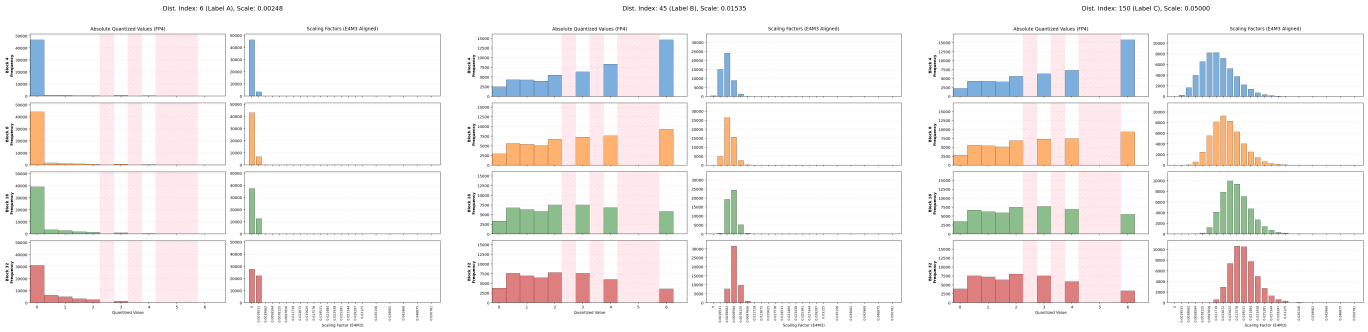
The following histograms illustrate the absolute quantized values and the corresponding scaling factors selected by each methodology across the three critical standard deviations (σ) identified as Regions A, B, and C in the numerical study discussed in Section III-A and shown in Figure 1. Within each figure, we show the histograms for the entries (left side) along with their corresponding scaling factors (right side). The color and row within each figure represent the four choices of block size. The shaded-out red region represents values not representable using the FP4 format.

These diagrams can help understand the numerical behavior under quantization using different recipes by inspecting the utilization of each bin of the entry and scale formats.

B. A closer look at model weights

This this section we inspect the raw weight values of Granite-3.3-8B and Qwen2.5-14B shown in Figure 2 to better understand how quantization and the range of values impact the perplexity gap.

Figures 8 and 9 show the structure and aggregate distribution of the Granite and Qwen models examined. For Granite, the distribution was more consistent among different layers, while for Qwen we find individual columns of smaller values for earlier layers (suggesting effectively structural sparsity). In later layers, the weight values of Qwen fall into larger magnitude bins at a higher proportion than Granite. This may help explain why despite the aggregate distributions in Figure 2 show more small values for Qwen, Granite was more sensitive to quantization with techniques requiring preservation of small values. In particular using E4M3 scaling factors without hierarchical scaling resulted in a consistently greater perplexity gap.

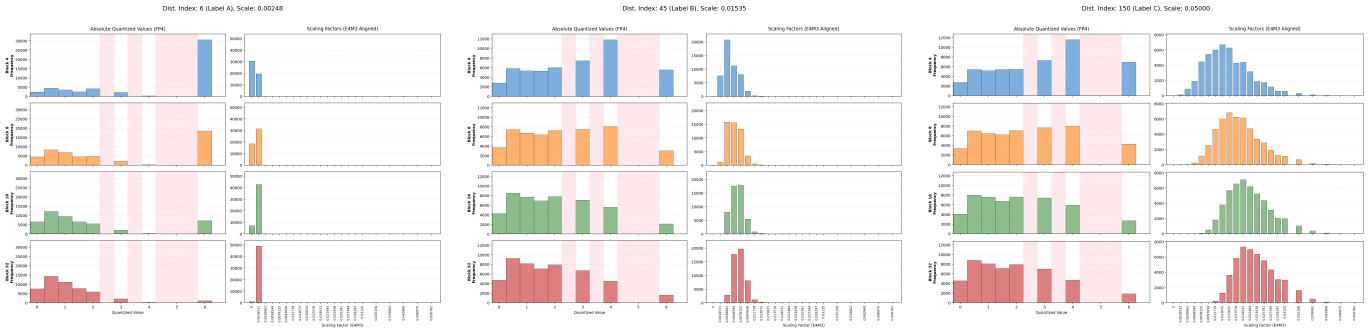


(a) Region A (b) Region B (c) Region C

Fig. 3: Standard Abs-Max Scaling. **(a) Region A:** At extremely small σ, the standard algorithm frequently selects a scaling factor of zero, obliterating element information and causing a sharp MSE spike. **(b) Regions B and C:** At these ranges, we find smaller blocks have more entries at the bin of magnitude 6. At Region B the scales are relatively concentrated at only a few bins, while at Region C they are spread out across a wider range.

(a) Region A (b) Region B (c) Region C

Fig. 4: Prevent-Zero Adjustment. **(a) Region A:** Restricting the scaling factor to the smallest positive representable value avoids a collapse to zero, preserving information and eliminating the localized MSE spike. **(b) Region B:** Elements remain pushed into the coarse quantization gap between 4 and 6, visually confirming that preventing zero scales does not resolve the structural element-level anomaly. **(c) Region C:** Behavior matches the standard abs-max methodology.



(a) Region A (b) Region B (c) Region C

Fig. 5: 4-over-6 Methodology. **(a) Region A:** While providing better overall bin utilization, the scaling factors can still struggle with extreme underflow in this tiny magnitude regime. **(b) Region B:** By adaptively allowing a maximum target scale of 4, the algorithm successfully avoids forcing elements into the massive upper quantization gap of FP4, resolving the scaling inversion paradox. **(c) Region C:** Similar to Region B, with scaling factors spread across a wider set of bins. In both Regions B and C, while there are fewer entries aggregated at the bin of magnitude 6 for small blocks, there are more at the bin of magnitude 4.

(a) Region A (b) Region B (c) Region C

Fig. 6: Brute-Force Search. **(a) Region A:** The scaling factors chosen align with the prevent-zero adjustment. Choosing a larger scaling value would result in underflowing too many entries **(b) Region B:** The scales show a somewhat wider distribution compared to 4-over-6, and significantly larger compared to abs-max. Although for smaller blocks a larger portion of entries fall into the magnitude 6 bin it is much less compared to abs-max (comparable to 4-over-6). However, many more values fall into bins smaller than magnitude 4 compared to 4-over-6, suggesting in some cases the optimal result requires using only the lower portion of the FP4 range. **(c) Region C:** Visually comparable to the 4-over-6 case. Although similar to Region B, there are more entries in smaller magnitude bins.

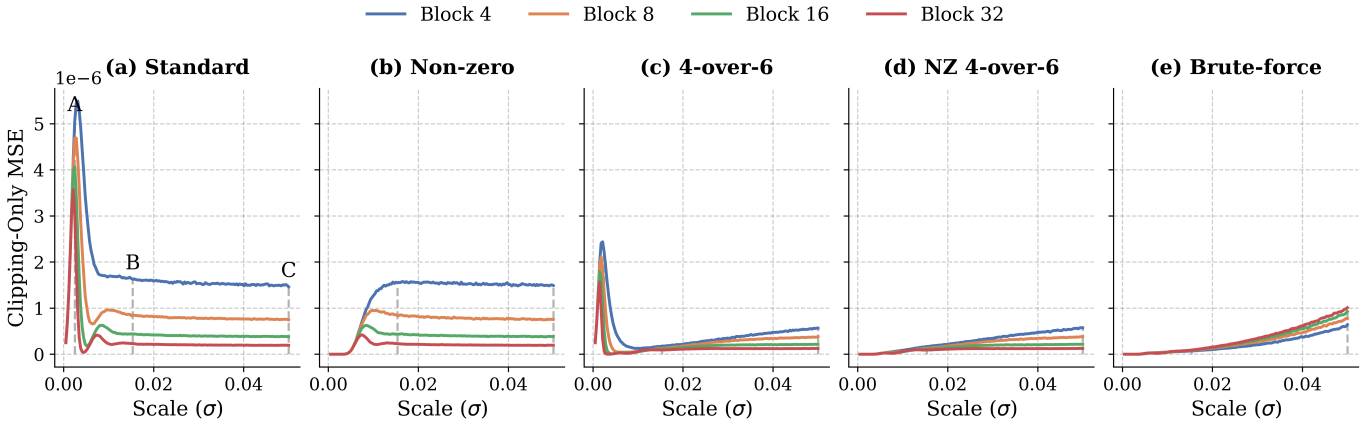
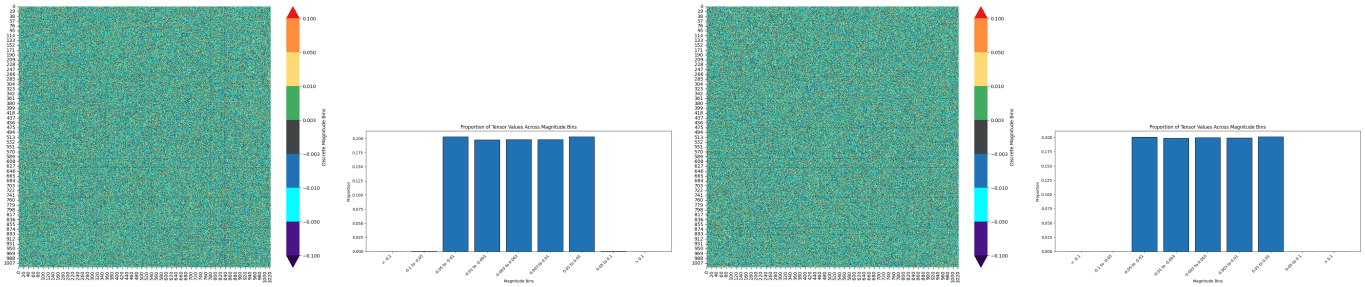


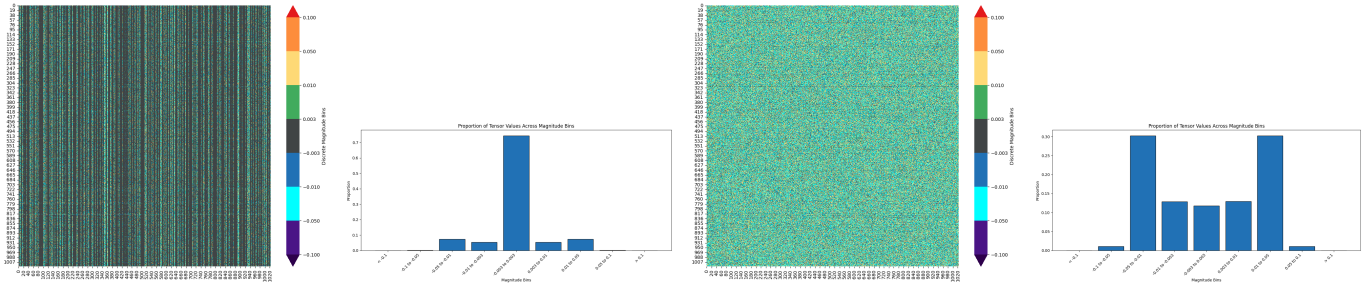
Fig. 7: Clipping only MSE. This plot is analogous to Figure 1, showing only the clipping portion. Using 4-over-6 or brute force substantially reduces the clipping error for smaller block sizes, explaining much of the inversion of MSE vs block size at Region B.

Figure 10 shown an ablation where values between a lower and upper magnitude thresholds have been set to zero. This confirms that the smaller magnitude values seen in Qwen are less consequential.



(a) Granite weights for layer 2 down projection (sample). (b) Granite weights for layer 2 down projection (distribution). (c) Granite weights for layer 2 down projection (sample). (d) Granite weights for layer 25 down projection (distribution).

Fig. 8: Granite 3.3 8B sample weights for layer 2 and layer 25 down projections. The distributions are similar at both parts of the network, with values approximately equally distributed among the five chosen bins.



(a) Qwen weights for layer 2 down projection (sample). (b) Qwen weights for layer 2 down projection (distribution). (c) Qwen weights for layer 25 down projection (sample). (d) Qwen weights for layer 25 down projection (distribution).

Fig. 9: Qwen 2.5 14B sample weights for layer 2 and layer 25 down projections. Earlier layers (a) show many columns with small values. In the aggregate the center bin [-0.003, 0.003] contains most of the values (b). However, the structural sparsity decreases for later layers, showing a more dense pattern (c). In the aggregate, larger magnitude bins contain most of the weight entries (d).

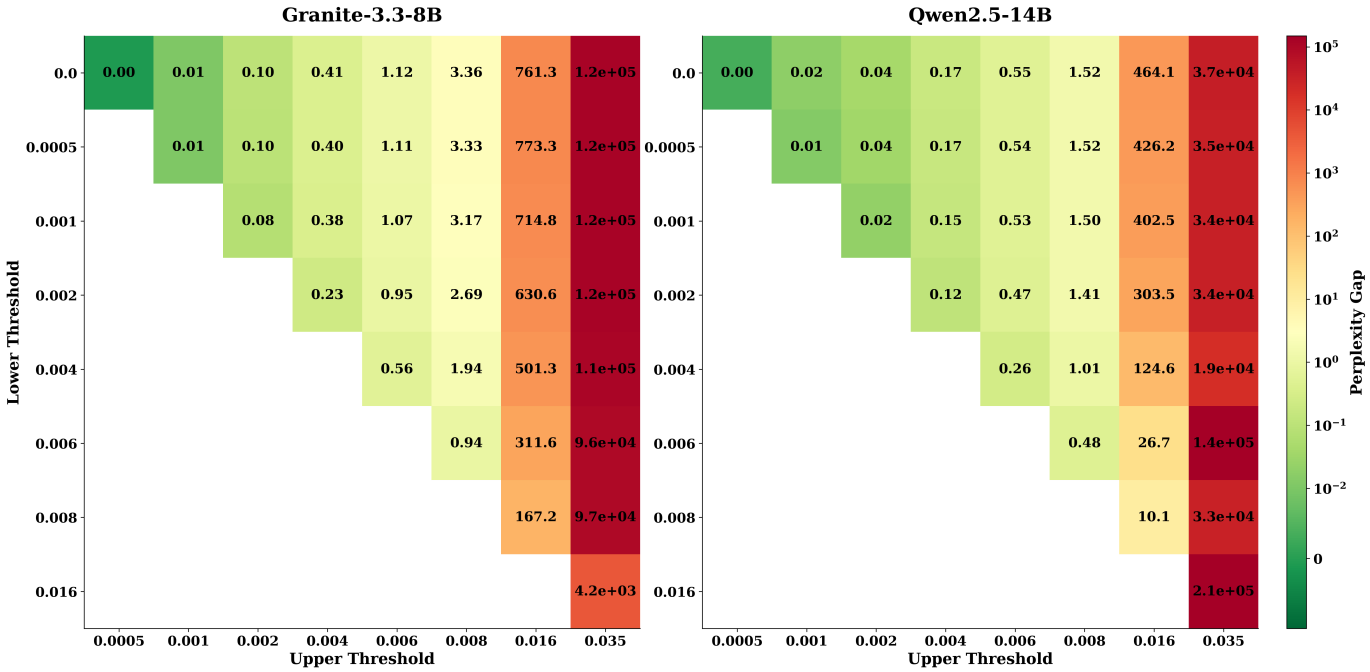


Fig. 10: Unquantized ablation with mapping of specific entry value ranges to zero (between lower and upper thresholds specified on the y and x axis). The heatmaps illustrate the downstream perplexity gap when specific intervals of absolute weight and activation values are masked to zero at full precision. Sweeping boundaries from 0.0 up to 0.035 reveals general robustness of Qwen2.5-14B compared to Granite-3.3-8B. The diagonal which zeros out only one bucket shows that drastic increase in the perplexity gap comes later for the Qwen model, generally showing that larger values in activation and weights are more important for the Qwen model while Granite also relies on smaller values. This verifies our main findings that our prevent zero method is effective for Granite to recover some of the perplexity with small block sizes. Meanwhile, Qwen is not negatively affected by smaller block sizes, since its most impactful weights are larger than the region most impacted by the coarse E4M3 scaling factors. Additionally, this highlights that purely analyzing raw weight or activation values is not sufficient to pick good quantization schemes since Qwen had several layers with small values (see Fig. 2), however, those value proved to be inconsequential for downstream perplexity.