TETRABACK: IDENTIFYING AND CORRECTING OPTIMIZATION BIAS IN MICROSCALED 4-BIT TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Microscaled 4-bit weight and activation quantization (W4A4) accelerates LLM inference, but maintaining accuracy at such low precision remains challenging. Activation quantization is a major source of this degradation, yet its effects on training dynamics remain underexplored. In this work, we show that this accuracy gap stems partly from optimization bias in the backward approximation, rather than solely from the limitations of low-bit representation. We consistently observe sustained growth in activation magnitudes and pre-FFN RMSNorm weights during standard MXFP4 W4A4 quantization-aware training (QAT), a pattern that impairs optimization and is absent when activations remain unquantized. We trace this behavior to the conventional straight-through estimator (STE) used for activation quantization, which ignores the gradient contribution from each group's scale and biases gradients reaching the preceding RMSNorm. We propose **TetraBack** to restore this missing gradient contribution without modifying forward computation. Our method curbs the abnormal growth of activation magnitudes and RMSNorm weights, consistently improves performance across 3B–94B MoE models, and reduces the loss gap induced by activation quantization by up to **41%**.

1 INTRODUCTION

Growing hardware support for microscaled 4-bit formats, including MXFP4 (Rouhani et al., 2023; AMD, 2025), HiF4 (Taghian et al., 2026), and NVFP4 (Alvarez et al., 2025), creates new opportunities for efficient large language model (LLM) inference. However, reducing numerical precision to four bits can substantially degrade model accuracy, limiting these benefits in practice. To mitigate this accuracy loss, quantization-aware training (QAT) (Jacob et al., 2018) incorporates quantization into training, allowing models to adapt to quantization errors before deployment. For example, gpt-oss (OpenAI, 2025) incorporates MXFP4 quantization of its mixture-of-experts weights during post-training and can reduce storage and memory bandwidth requirements. Nonetheless, fully realizing the acceleration potential of native 4-bit $\times$ 4-bit matrix multiplication requires quantizing activations to four bits as well.

In practice, this transition introduces additional quantization errors that can substantially degrade accuracy even with QAT (Chen et al., 2026a;b). To address this challenge, prior work has largely focused on improving forward representations through outlier compensation (Wang et al., 2025; Dong et al., 2026), rotations (Ashkboos et al., 2024; Liu et al., 2025), and selective higher-precision computation (Chen et al., 2026b). Despite these advances, a substantial performance gap remains relative to training with higher-precision activations (Chen et al., 2026a;b). Moreover, online transformations and auxiliary compensation paths can introduce additional computation or complexity at inference time. These limitations motivate us to examine how activation quantization affects the course of training, seeking accuracy improvements without adding inference computation.

To understand the remaining performance gap in Transformer-based MoE LLMs (Vaswani et al., 2017; Shazeer et al., 2017), we track internal model statistics during MXFP4 W4A4 training while using higher-precision backward computation (BF16 (Kalamkar et al., 2019) or MXFP8). This investigation reveals an unexpected pattern: activation magnitudes and pre-FFN RMSNorm (Zhang & Sennrich, 2019) weights grow persistently, diverging from their trajectories without activation quantization. In particular, MoE output magnitudes reach approximately 7$\times$ those of the activation-unquantized baseline by the end of training (Fig. 1(a)), whereas weight quantization alone has little

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

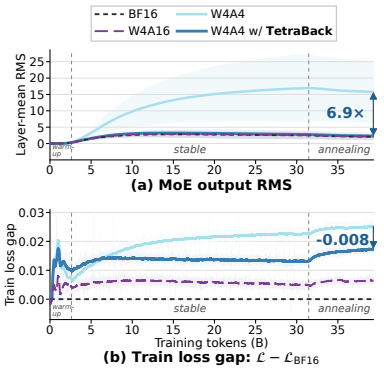


Figure 1: **TetraBack** corrects optimization bias in W4A4 training in MoE-3B adapted from DeepSeek-V4.

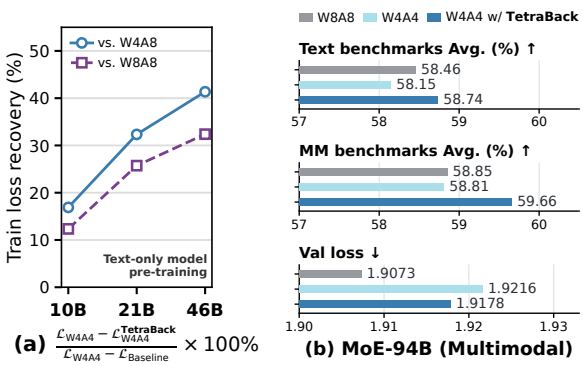


Figure 2: **(a)** Training-loss gap *reduction* across text-only 10B–46B MoE models. **(b)** Benchmarks and validation loss (Sec. 4) of DeepSeek-V4.1-Flash-style MoE-94B.

effect on these dynamics. Controlled ablations with BF16 backward computation show that this growth is harmful, as constraining RMSNorm weights improves W4A4 training but provides no benefit when activations remain unquantized (Fig. 4). Alongside this drift, the relative training-loss gap to W4A16 widens from 0.10% at the end of warmup to 0.97% by the end of training (Fig. 1(b)), raising concerns about the effects of activation quantization over longer training horizons.

We identify the cause of this harmful drift as a missing gradient contribution in the conventional straight-through estimator (STE) for microscaled activation quantization. Existing MXFP4/NVFP4 training implementations (NVIDIA et al., 2026; Cim et al., 2026; Chen et al., 2026b; Panferov et al., 2026) commonly apply a matrix-wise STE, $\frac{\partial \mathcal{L}}{\partial \mathbf{X}} \approx \frac{\partial \mathcal{L}}{\partial Q(\mathbf{X})}$, where $Q(\cdot)$ denotes the quantizer. This approximation passes gradients directly through the quantizer, ignoring that each group's scale depends on the absolute maximum of its input activations. Changing an activation can therefore alter the shared scale and affect all quantized values within the group, a dependency that the matrix-wise STE fails to capture. The resulting gradient bias propagates to preceding modules, driving the observed growth in pre-FFN RMSNorm weights and activation magnitudes.

To address this gradient bias, we introduce **TetraBack**, a backward correction that restores the missing scale-dependent gradient contribution while leaving the forward MXFP4 computation unchanged. Our method curbs the abnormal growth in pre-FFN RMSNorm weights and activation magnitudes, bringing their training trajectories closer to those observed without activation quantization. Across model sizes, this correction consistently reduces training loss and recovers up to **41%** of the loss gap induced by activation quantization, without adding inference overhead. Our main contributions are summarized below:

- **Identifying the optimization bias.** We uncover a previously overlooked Norm and activation drift in activation-quantized training and attribute this drastic model-level behavior to a subtle missing gradient path in activation-microscaling backpropagation.
- **TetraBack.** We introduce a simple group-wise correction that restores the omitted shared-scale gradient contribution, which suppresses the abnormal optimization drift, and improves W4A4 training without changing the forward computation or adding inference overhead.
- **Validation at scale.** We validate TetraBack across 3B–94B MoE models adapted from DeepSeek-V4 (DeepSeek-AI et al., 2026b) and DeepSeek-V4.1-Flash (DeepSeek-AI et al., 2026a) architectures. It consistently improves MXFP4 W4A4 training, recovering up to **41**% of the train-loss gap to W4A8. W4A4 with TetraBack even reaches downstream performance comparable to W8A8 on MoE-94B. TetraBack also effectively extends to NVFP4 training.

2 The Optimization Problem in Microscaled W4A4 Training

Standard microscaled W4A4 QAT. In this work, we train Mixture-of-Experts (MoE) (Shazeer et al., 2017) language models, and quantize the two linear projections of each routed FFN using

MXFP4 (Rouhani et al., 2023) for both weights and activations (W4A4). W4A16 applies the same MXFP4 weight quantization while keeping the corresponding activations in high precision.

Following standard QAT recipes, quantization is applied in the forward pass, while in the backward, the gradient for the non-differentiable quantizer is estimated by a matrix-wise identity straight-through estimator (STE) (Bengio et al., 2013). For an activation matrix $\mathbf{X}$ and its MXFP4 quantized-dequantized counterpart $Q(\mathbf{X})$,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{X}} \approx \frac{\partial \mathcal{L}}{\partial Q(\mathbf{X})}. \tag{1}$$

Since we adopt the truncation-free scaling of TetraJet (Chen et al., 2025), no clipping mask is needed for this STE. This natural estimation is commonly adopted for activation and weight quantization in modern FP4 training recipes (NVIDIA et al., 2026; Chen et al., 2026b; Panferov et al., 2026; Cim et al., 2026). In this work, we keep the backward in BF16 throughout diagnostic experiments to exclude additional errors from low-precision backward computation. Details of the MXFP4 are provided in App. B. We use the root-mean-square (RMS) to directly compare tensor magnitudes. For a vector $\mathbf{x} \in \mathbb{R}^d$ and a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, we define

$$\mathrm{RMS}(\mathbf{x}) = \frac{\|\mathbf{x}\|_2}{\sqrt{d}} = \sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}, \qquad \mathrm{RMS}(\mathbf{X}) = \frac{\|\mathbf{X}\|_F}{\sqrt{mn}} = \sqrt{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} X_{ij}^2}.$$

Activation quantization induces a persistent magnitude elevation. Although the weight-only MXFP4 quantization introduces only a modest training degradation (W4A16 in Fig. 1(b)), the activation quantization substantially enlarges the loss gap (W4A4 in Fig. 1(b)).

Surprisingly, we observe a previously overlooked optimization behavior: introducing MXFP4 activation quantization substantially alters the evolution of activation magnitudes inside the model (Fig. 1(a)). As training progresses, the MoE output magnitudes under W4A4 become increasingly larger than those under W4A16 and W16A16 (BF16). The gap keeps widening over the course of training, reaching $\sim \mathbf{7}\times$ the corresponding high-precision-activation MoE output scale in the end.

Notably, this behavior is *specific to activation quantization*. With MXFP4 weight-only quantization, W4A16 training remains close to BF16 in the activation trajectories. Thus, MXFP4 activation quantization introduces not only forward representation error, but also gradually drives the model toward a different optimization trajectory.

The Norm weight divergence is localized before quantized FFN layers. We next locate which persistent model parameters follow this altered trajectory. Fig. 3 compares the RMSNorm modules preceding attention and FFN blocks. The Pre-Attention RMSNorms evolve similarly under BF16, W4A16, and W4A4. In contrast, the Pre-FFN RMSNorms diverge strongly under W4A4, and these Norms immediately precede the quantized FFN layers. Therefore, the optimization drift is concentrated immediately at the RMSNorms preceding the FFN. The contrast between Pre-FFN and Pre-Attention Norms is consistent with our setup, which quantizes routed-FFN activations but leaves attention projections in higher precision.

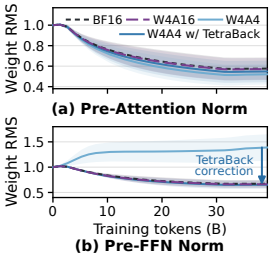


Figure 3: RMS of Norm weights in MoE-3B (layer mean ± SD).

The enlarged Norm scale is a *harmful* optimization outcome. To test whether this scale elevation causally contributes to the suboptimal outcome rather than merely accompanying the degradation, we intervene on the RMSNorm weight $\boldsymbol{\gamma}_i$ preceding each MoE layer i throughout training in three different ways: **(1) *Removing*** learnable channel-wise scaling from the Pre-FFN RMSNorm (w/o $\boldsymbol{\gamma}_i$) by fixing $\boldsymbol{\gamma}_i = \mathbf{1}$, **(2) *Fixing*** the Pre-FFN Norm weight scale (RMS $= 1$) after each optimizer update. **(3) *Matching*** the Pre-FFN Norm weight scale to that of the final W4A16 checkpoint (RMS $= \mathrm{RMS}_{\mathrm{W4A16}}$), respectively for each layer.

Their effect on the training loss is shown in Fig. 4. Strikingly, the Norm constraints provide *little benefit* for W4A16 and can even hurt training. **In contrast, all three interventions improve W4A4 training**. This contrast shows that the enlarged Norm scale is a *suboptimal* outcome of W4A4

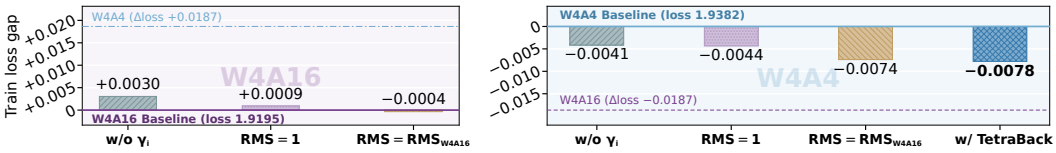


Figure 4: Constraining Pre-FFN Norm weights alone *reduces* W4A4 training loss, indicating that the learned Norm scales are suboptimal. Results are for MoE-3B, averaged over the final 200 steps.

optimization that directly contributes to the degradation. In the next section, we show this problem is ***resolvable*** by revisiting the backward of the microscaling quantizer.

3 TetraBack: Restoring the Missing Scale-Dependent Gradient

Sec. 2 shows that MXFP4 activation quantization drives the Norm parameters immediately preceding the quantized FFNs toward a suboptimal optimization trajectory. In this section, we show that the problem lies in the *backward* of microscaled activation quantization: the shared quantization scale is itself determined by the input activation, but the conventional STE *does not propagate gradients* through this path. We propose **TetraBack** to effectively resolve this problem.

The missing scale-dependent path. The conventional matrix-wise STE $\frac{\partial \mathcal{L}}{\partial \mathbf{X}} \approx \frac{\partial \mathcal{L}}{\partial Q(\mathbf{X})}$ in Eq. (1) is commonly adopted in the modern MXFP4 / NVFP4 training recipes (Cim et al., 2026; NVIDIA et al., 2026; Chen et al., 2026b; Panferov et al., 2026) and is considered a natural approximation when the quantizer is viewed as a single non-differentiable mapping.

However, this matrix-wise STE introduces *bias* during the backward. Each MXFP4 group $\mathbf{X}_g$ shares a *scale factor* S_g determined by its *group absolute max* M_g (abs-max). Consequently, an input $X_{g,j}$ affects the quantized-dequantized output $\widehat{\mathbf{X}}_g = Q(\mathbf{X}_g)$ (defined in App. B) through two paths:

$$X_{g,j} \longrightarrow \widehat{X}_{g,j}, \qquad \text{and} \qquad X_{g,j} \longrightarrow M_g \longrightarrow S_g \longrightarrow \widehat{\mathbf{X}}_g = \{\widehat{X}_{g,i}\}_{i=0}^{31}.$$

The conventional matrix-wise STE in Eq. (1) retains *only the first path* during backpropagation. We later show that this backward bias propagates to the Pre-FFN Norm weights, systematically biases their gradients, and significantly drives the optimization difference in Sec. 2. Therefore, adding the shared-scale dependency gives our solution, **TetraBack**:

$$\textit{Matrix-wise STE:} \quad \frac{\partial \mathcal{L}}{\partial X_{g,j}} \approx \frac{\partial \mathcal{L}}{\partial \widehat{X}_{g,j}}, \tag{2}$$

$$\textbf{\textit{TetraBack} (for activation):} \quad \frac{\partial \mathcal{L}}{\partial X_{g,j}} \approx \frac{\partial \mathcal{L}}{\partial \widehat{X}_{g,j}} + \underbrace{\left[\sum_{i=0}^{31} \frac{\partial \mathcal{L}}{\partial \widehat{X}_{g,i}} \cdot \frac{\partial \widehat{X}_{g,i}}{\partial S_g}\right] \cdot \frac{\partial S_g}{\partial M_g} \cdot \frac{\partial M_g}{\partial X_{g,j}}}_{\text{scale-dependent gradient } \textbf{correction term}}, \tag{3}$$

Thus, the STE in Eq. (2) discards a gradient path through the scale factor S_g, and Eq. (3) corrects it.

Differentiable surrogate for the microscaling factor. As shown in Fig. 5, the exact scale function $S_g(M_g)$ is a power-of-two staircase function. For a fixed scale S_g, the corresponding abs-max satisfies $3S_g < M_g \leq 6S_g$, where 6 is the largest finite E2M1 value and the lower bound is 3 because the next smaller UE8M0 scale is $S_g/2$. We use a simple linear surrogate $\widetilde{S}_g(M_g) = \frac{M_g}{4.5}$ passing through the center $M_g = 4.5S_g$ of each scale stair. This surrogate is used only in backward; the forward pass remains unchanged. Alternative surrogate designs are studied in App. D.

The resulting correction. Let $j_g^* := \arg\max_i |X_{g,i}|$ denote the element that determines the group scale. Using an identity STE for E2M1 rounding, and the differentiable surrogate function $\widetilde{S}_g(M_g)$ for the STE of UE8M0 scale factor, the correction in Eq. (3) simplifies to

$$\boxed{\frac{\partial \mathcal{L}}{\partial X_{g,j}} \approx \frac{\partial \mathcal{L}}{\partial \widehat{X}_{g,j}} + \frac{\mathbf{1}\{j = j_g^*\}\,\mathrm{sign}(X_{g,j})\widetilde{S}_g'(M_g)}{S_g} \sum_{i=0}^{31} \frac{\partial \mathcal{L}}{\partial \widehat{X}_{g,i}} \left(\widehat{X}_{g,i} - X_{g,i}\right).} \tag{4}$$

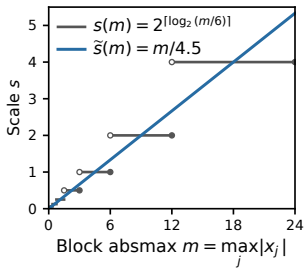


Figure 5: The MXFP4 scale and the differentiable surrogate.

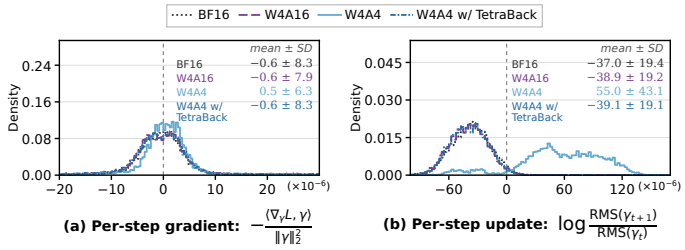


Figure 6: W4A4 exhibits a subtle shift in the Pre-FFN Norm gradient statistic and a pronounced shift toward growth in per-step updates in MoE-3B. **TetraBack** effectively corrects this bias.

The full derivation is provided in App. C. The correction has a simple group-wise interpretation. Its value is determined by a length-32 per-group inner product $\langle \partial \mathcal{L}/\partial \widehat{\mathbf{X}}_g, \widehat{\mathbf{X}}_g - \mathbf{X}_g \rangle$ between the incoming gradient and the quantization residual. It therefore aggregates how the shared scale affects the entire group, while passing the correction back *sparsely*, only through the abs-max element j_g^* that determines that scale. All other elements retain exactly the conventional matrix-wise STE.

Relation to prior scale-gradient methods. LSQ (Esser et al., 2020) optimizes a *learnable* per-tensor quantization scale using its gradient. For *input-dependent* microscaling factors, Hu et al. (2025) include scale derivatives, but their loss-gradient expression uses an *incomplete* chain rule:

$$\textit{Hu et al. (2025):} \quad \frac{\partial \mathcal{L}}{\partial \mathbf{X}_g} \approx \frac{\partial \mathcal{L}}{\partial \widehat{\mathbf{X}}_g} + \frac{\partial \mathcal{L}}{\partial \widehat{\mathbf{X}}_g} \odot \frac{\partial \widehat{\mathbf{X}}_g}{\partial S_g} \odot \frac{\partial S_g}{\partial \mathbf{X}_g} \qquad \times \text{ Incomplete}$$

$$\textbf{\textit{TetraBack:}} \quad \frac{\partial \mathcal{L}}{\partial \mathbf{X}_g} \approx \frac{\partial \mathcal{L}}{\partial \widehat{\mathbf{X}}_g} + \left(\sum_{i=0}^{31} \frac{\partial \mathcal{L}}{\partial \widehat{X}_{g,i}} \cdot \frac{\partial \widehat{X}_{g,i}}{\partial S_g} \right) \cdot \frac{\partial S_g}{\partial \mathbf{X}_g} \qquad \checkmark \text{ Complete}$$

Here, the scale S_g affects all group outputs $\{\widehat{X}_{g,i}\}_{i=0}^{31}$, so their gradient contributions must be summed. Beyond completing this chain rule with Eq. (4), we show that substantial Norm drift persists both with the standard STE and with the incomplete correction above, a problem not identified in their study. App. F.4 provides a detailed theoretical and experimental comparison.

Why does this correction resolve the drift? To connect this local backward correction to the model-level drift, we measure the normalized radial component of the Pre-FFN RMSNorm gradient, $-\langle \nabla_\gamma \mathcal{L}, \gamma \rangle / \|\gamma\|_2^2$, where a positive value corresponds to a gradient-descent tendency to *increase* the Norm scale. Also, we use $\log(\mathrm{RMS}(\gamma_{t+1})/\mathrm{RMS}(\gamma_t))$ to measure the actual scale updates of the weight at step t (a 500-step window around 7.5% of MoE-3B training). As a result, omitting this scale path induces a small systematic shift in the Norm gradient statistic (Fig. 6(a)) and a pronounced shift toward Norm growth in per-step updates (Fig. 6(b)), while TetraBack effectively corrects both shifts. Together with the recovery of the long-horizon RMSNorm and activation trajectories in Figs. 1 and 3, this shows that **TetraBack resolves the optimization drift** identified in Sec. 2.

Scope. We apply TetraBack only to activation quantization, as the weight-side correction does not further improve training (App. E). Notably, TetraBack modifies only the backward computation; the MXFP4 forward computation is unchanged, and no modification is needed to the inference system.

4 Experiments and Analysis

4.1 Experimental Setup

Models and training. We validate TetraBack across two pre-norm MoE architecture families, including text-only and multimodal training. We train MoE-3B for diagnostics on 39.3B tokens, which is adapted from DeepSeek-V4 (DeepSeek-AI et al., 2026b) with GQA (Ainslie et al., 2023) rather than sparse attention. For scaling experiments, we train 10B, 21B, and 46B text-only MoEs on 60B,

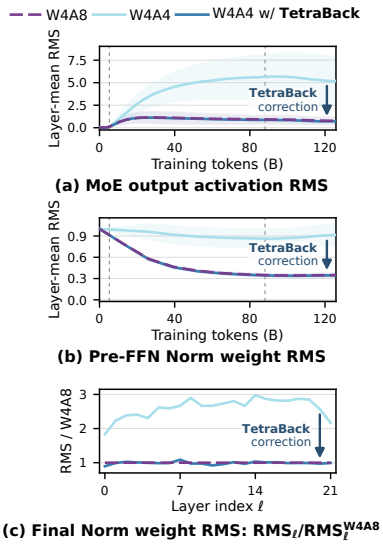


Figure 7: Dynamics of MoE-21B. **(a)** MoE output RMS. **(b)** Pre-FFN Norm weight RMS. **(c)** Final per-layer Norm RMS relative to W4A8. Shading shows ± 1 layer-wise SD.

Table 1: Benchmark results of the MoE-94B training (566B tokens). Best scores are **bold**; second-bests are underlined.

Benchmark	**W8A8** (MXFP8)	**W4A4** (MXFP4)	**W4A4** (MXFP4) w/ TetraBack
Text-only benchmarks ↑			
MMLU (5-shot) (Hendrycks et al., 2021a)	**66.38**	65.50	66.33
MMLU-Pro (5-shot) (Wang et al., 2024)	37.39	37.38	**37.68**
C-Eval (5-shot) (Huang et al., 2023)	**68.28**	65.82	67.76
TriviaQA (5-shot) (Joshi et al., 2017)	**62.57**	61.03	61.99
HellaSwag (0-shot) (Zellers et al., 2019)	**76.98**	76.35	76.96
PIQA (0-shot) (Bisk et al., 2020)	**80.79**	80.58	80.63
BBH (3-shot) (Suzgun et al., 2023)	**65.41**	64.63	64.10
DROP (F1) (1-shot) (Dua et al., 2019)	64.02	65.85	**66.31**
RACE-High (5-shot) (Lai et al., 2017)	**49.83**	47.91	49.26
GSM8K (8-shot) (Cobbe et al., 2021)	70.05	**71.42**	70.89
MATH (4-shot) (Hendrycks et al., 2021b)	38.34	38.42	**40.38**
HumanEval (0-shot) (Chen et al., 2021)	42.07	**43.29**	**43.29**
BigCodeBench (3-shot) (Zhuo et al., 2025)	37.81	37.72	**37.98**
Text-only Benchmarks Avg.	58.46	58.15	**58.74**
Multimodal benchmarks ↑			
MMMU-Pro (4-shot) (Yue et al., 2025)	28.18	**29.51**	28.61
MMStar (4-shot) (Chen et al., 2024)	**47.41**	43.23	46.26
CVBench (4-shot) (Tong et al., 2024)	58.00	58.34	**61.26**
RealWorldQA (4-shot) (xAI, 2024)	55.29	57.91	**58.43**
AI2D (4-shot) (Kembhavi et al., 2016)	69.98	69.79	**70.24**
MathVista (4-shot) (Lu et al., 2024)	**46.80**	46.20	46.10
DocVQA (4-shot) (Mathew et al., 2021)	89.79	**91.38**	90.84
ChartQA (4-shot) (Masry et al., 2022)	75.36	74.12	**75.52**
MM Benchmarks Avg.	58.85	58.81	**59.66**
Overall Benchmarks Avg. ↑	58.61	58.40	**59.09**
Pile (Gao et al., 2020) Val. loss ↓	**1.9073**	1.9216	1.9178

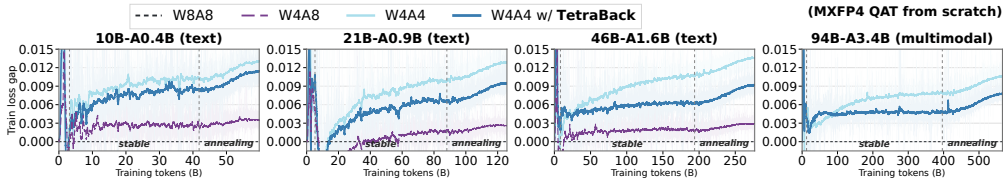


Figure 8: Training-loss gaps to W8A8 across MoE model scales (text-only training for MoE-10B – 46B; multimodal for MoE-94B). **TetraBack** effectively reduces the loss gap of W4A4.

126B, and 277B tokens, respectively, and a 94B multimodal MoE on 566B tokens; these models downscale DeepSeek-V4.1-Flash (DeepSeek-AI et al., 2026a) while omitting Engram (Cheng et al., 2026). Details are provided in App. A.

Quantization setup. Quantization is enabled for the routed-expert linear projections in *all* MoE layers throughout the training. Our main setting is MXFP4 W4A4, with W4A16 as the weight-only reference for MoE-3B diagnostics and W4A8/W8A8 as larger-scale baselines. Other components remain at higher precision; backward uses BF16 for MoE-3B and MXFP8 for larger-scale.

4.2 From-Scratch W4A4 Quantization-Aware Training at Scale

Tab. 1 reports MoE-94B results on 13 text and 8 multimodal benchmarks, spanning language, reasoning, coding, and multimodal understanding. These results show that correcting the optimization bias and improving training loss translate into downstream benefits: TetraBack raises the text and multimodal benchmark averages by 0.59 and 0.85 percentage points over standard W4A4, respectively, reaching an overall average of 59.09, which is even better than W8A8.

The MoE-21B trajectories in Fig. 7 show that TetraBack *corrects* the biased activation-magnitude and Pre-FFN Norm weight trajectories relative to conventional W4A4 training. Fig. 8 presents the W4A4 training-loss gap relative to W8A8 for the 10B–46B text-only and the 94B multimodal model.

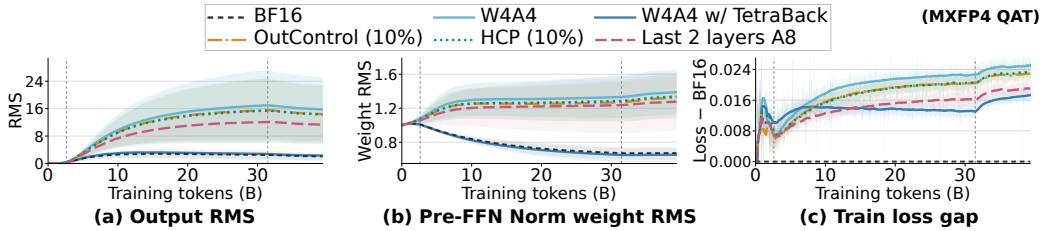


Figure 9: Forward-precision baselines in MoE-3B MXFP4 W4A4 training. These baselines *add forward precision* but retain *biased* optimization ((a) enlarged MoE outputs and (b) Norm weights).

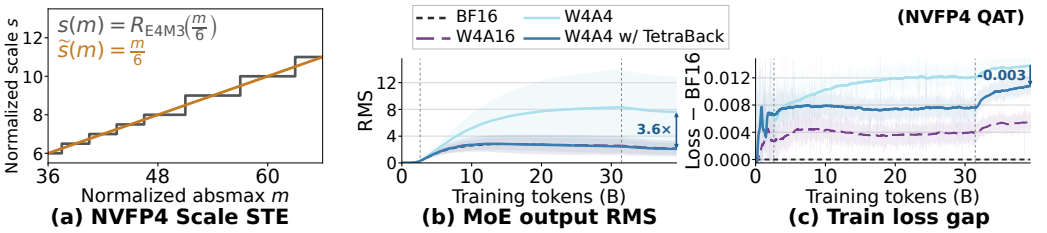


Figure 10: Transferring TetraBack to NVFP4 training on MoE-3B. **(a)** differentiable surrogate for scale factor. **(b,c)** TetraBack effectively corrects optimization and reduces train-loss gap.

4.3 Ablation: Separating Backward Bias from Forward Quantization Error

Can reducing forward quantization error in W4A4 also prevent the abnormal growth of activation and Norm magnitudes? We compare TetraBack with three baselines that change the forward activation precision using the same 3B MXFP4 W4A4 training setup:

- **OutControl**: retaining selected 10% outlier channels of activation in BF16 (Chen et al., 2026b);
- **HCP (activation-only)**: adding an MXFP4 activation-residual patch on selected 10% hot channels for GEMM error compensation (Dong et al., 2026);
- **Last-2-layer A8**: using MXFP8 activations on last 16.7% layers inspired by Agrusa et al. (2026).

As shown in Fig. 9, forward-error compensation can *recover part of* the W4A4 loss gap to the BF16 baseline on MoE-3B, but ***does not fully correct*** the abnormal Pre-FFN Norm and activation-magnitude dynamics. In contrast, **TetraBack** directly corrects the optimization drift ***without modifying the forward***. This suggests that forward compensation addresses only part of the W4A4 problem; the backward optimization bias remains and should be corrected separately.

4.4 Generalization and Practicality

We find this optimization bias also exists in NVFP4 (Alvarez et al., 2025) training, so we extend TetraBack to NVFP4 activation quantization with a similar gradient correction. As shown in Fig. 10, TetraBack improves NVFP4 training and corrects the optimization drift. This indicates that the issue is not specific to MXFP4 and can also arise in other microscaled formats such as NVFP4.

Practically, TetraBack modifies only the backward computation, leaving the forward path unchanged and adding no inference-time overhead. During training, the group-wise correction can be fused into existing activation-quantization and SwiGLU backward kernels following the computation of input gradients. When the original BF16 activations are already retained for backward, TetraBack can reuse them without saving additional activation tensors. Further system-level optimization under lower-precision backward computation and communication is left for future work.

4.5 How Does the Gradient Correction Term Affect Upstream Norms?

In this subsection, we analyze how the gradient correction of the FFN activation quantizer changes the optimization of preceding Norms. We first isolate activation quantization at the two FFN linears. We find that quantizing either `Linear1` or `Linear2` alone raises the Pre-FFN Norm scale and

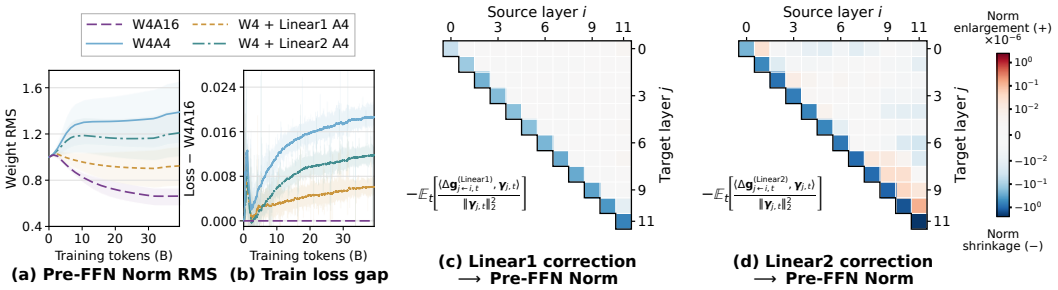


Figure 11: Tracing the Pre-FFN Norm drift and the correction effect on W4A4 training of MoE-3B. **(a,b)** Either `Linear1` or `Linear2` activation quantization increases Pre-FFN Norm RMS and train loss w/o TetraBack. **(c,d)** We enable the gradient correction at one Linear input for one layer i at a time, and trace its *backpropagated effect* to the upstream Norm layers $j \le i$ by Eq. (5).

increases the training-loss gap without TetraBack correction, with a substantially stronger effect from `Linear2` (Fig. 11(a,b)). This shows that activation quantization at both FFN projections contributes to the optimization drift identified in Sec. 2.

We then ask how the added correction term $\Delta \mathbf{g}_{\mathbf{X}_i}$ in layer i affects the upstream Norms at layer j ($j \le i$). Recall that TetraBack restores the scale-dependent term obtained by decomposing the input Jacobian $\partial \widehat{\mathbf{X}}_g / \partial \mathbf{X}_g$. For each microscaling group g, the omitted contribution to $X_{g,l}$ is

$$\Delta \mathrm{g}_{X_{g,l}} = \left[\sum_{r=0}^{31} \frac{\partial \mathcal{L}}{\partial \widehat{X}_{g,r}} \frac{\partial \widehat{X}_{g,r}}{\partial S_g} \right] \frac{\partial S_g}{\partial M_g} \frac{\partial M_g}{\partial X_{g,l}}, \qquad l \in \{0, \dots, 31\}.$$

i.e., the term omitted by the conventional matrix-wise STE. For either `Linear1` or `Linear2` in source layer i, denote its quantizer input by $\mathbf{X}_i$ and the corresponding TetraBack correction by $\Delta \mathbf{g}_{\mathbf{X}_i}$. We backpropagate only this additional gradient through the original network to each upstream Pre-FFN Norm weight $\boldsymbol{\gamma}_j$, $j \le i$:

$$\Delta \mathbf{g}_{j \leftarrow i} = \left(\frac{\partial \, \mathrm{vec}(\mathbf{X}_i)}{\partial \, \boldsymbol{\gamma}_j^\top} \right)^\top \mathrm{vec}(\Delta \mathbf{g}_{\mathbf{X}_i}).$$

We measure its effect on the Norm scale by

$$H_{j \leftarrow i} = -\mathbb{E}_t \left[\frac{\langle \Delta \mathbf{g}_{j \leftarrow i,t}, \boldsymbol{\gamma}_{j,t} \rangle}{\|\boldsymbol{\gamma}_{j,t}\|_2^2} \right]. \tag{5}$$

Thus, $H_{j \leftarrow i} < 0$ corresponds to a correction that tends to *shrink* the Norm weight scale under gradient descent, while $H_{j \leftarrow i} > 0$ tends to *enlarge* it. The value $|H_{j \leftarrow i}|$ indicates the strength of such effects. The metric is averaged over steps t across a window ($t = 2\text{k} \sim 2.5\text{k}$ steps in Fig. 11).

Fig. 11(c,d) shows that the correction effect is strongly concentrated on the *same-layer* Norm ($j = i$) and is predominantly a shrinking effect, especially for `Linear2`. This matches the stronger drift induced by `Linear2` in Fig. 11(a,b), and directly explains why correction can curb the abnormal growth of Pre-FFN Norms under W4A4. Equivalently, omitting this correction induces an enlarging bias on the same-layer Pre-FFN Norm weight. More analysis on the effect of TetraBack is in App. F.

5 Related Work

Post-training quantization. Post-training quantization (PTQ) quantizes pretrained models for efficient inference. For large language models (LLMs), SmoothQuant (Xiao et al., 2023) addresses activation outliers by shifting quantization difficulty to weights. Subsequent methods introduce fixed rotations in QuaRot (Ashkboos et al., 2024), learned rotations in SpinQuant (Liu et al., 2025), and more general affine transformations in FlatQuant (Sun et al., 2025). These approaches avoid end-to-end retraining but can incur substantial accuracy degradation at low bit widths.

Low-bit representations and scaling. Recent work also revisits the low-bit representations and block scaling rules in Microscaling (Rouhani et al., 2023): any4 (Elhoushi & Johnson, 2025) learns arbitrary 4-bit codebooks, Four Over Six (Cook et al., 2026) adaptively modifies NVFP4 block scaling, and Zhao et al. (2026) advocate uniform 4-bit representations to mitigate E2M1 shrinkage bias. These methods modify the forward quantization, whereas our work keeps the forward quantizer unchanged and corrects an overlooked backward bias from the omitted shared-scale gradient path.

4-bit training. 4-bit training incorporates 4-bit quantization into model optimization for efficient inference or training, targeting weights, backward operands, and activations.

Weight quantization reduces storage and arithmetic costs but can cause optimization challenges. To address master-weight oscillations between adjacent quantization levels, Nagel et al. (2022) use dampening and iterative weight freezing, while TetraJet-v2 (Chen et al., 2026b) periodically resets oscillating master weights to quantization bin centers. Separately, CAGE (Tabesh et al., 2026) adds curvature-aware corrections to the weight STE, while earlier EWGS (Lee et al., 2021) uses error-aware element-wise gradient scaling informed by Hessian estimates. The optimization issue studied in this work arises from microscaled *activation* quantization rather than weight quantization.

Backward quantization accelerates training by reducing the precision of operands used in gradient computation. Tseng et al. (2025) combine stochastic rounding with random Hadamard transforms for unbiased gradient estimation; NVIDIA's NVFP4 recipe (NVIDIA et al., 2026) similarly transforms GEMM operands to mitigate outliers. TetraJet (Chen et al., 2025) introduced forward-aligned double MXFP4 quantization for unbiased estimation of the corresponding STE gradients, which TetraJet-v2 (Chen et al., 2026b) extends to NVFP4. Quartet II (Panferov et al., 2026) further reduces backward quantization variance while preserving unbiasedness. Our work examines an *activation-quantization-induced* optimization bias that persists even with BF16 backward computation.

Activation quantization and optimization. Prior work on training with 4-bit activations largely focuses on handling outliers and evolving activation distributions. Wang et al. (2025) combine outlier clamping with compensation to reduce activation quantization error. Metis (Cao et al., 2026) partitions anisotropic spectra into narrower sub-distributions for separate quantization, preserving spectral structure. TetraJet-v2 (Chen et al., 2026b) retains selected activation channels in higher precision, while Dong et al. (2026) trace the emergence of persistent hot channels and compensate for their quantization residuals. These approaches primarily improve representational accuracy. Our work instead corrects an optimization bias in microscaled W4A4 QAT that drives harmful growth in pre-FFN RMSNorm weights and activation magnitudes, without adding inference computation.

6 Conclusion and Limitations

Conclusion. In this work, we identify an unexpected optimization problem in MXFP4 W4A4 training: activation magnitudes and pre-FFN RMSNorm weights progressively rise above the activation-unquantized baseline, leading to a suboptimal optimization trajectory. We trace this behavior to the standard matrix-wise STE for microscaled activation quantization, which omits the gradient contribution from the input-dependent shared scale to the quantizer input. We therefore propose TetraBack to restore the missing gradient path, which resolves the optimization bias and improves W4A4 training with no inference overhead.

We validate TetraBack on 3B–94B models across two architectures adapted from DeepSeek-V4 and DeepSeek-V4.1-Flash, including text-only and multimodal training, and extend it from MXFP4 to NVFP4. Microscaled 4-bit activation quantization can affect training not only through forward quantization error but also through backward optimization bias. When such a quantizer lies on an intermediate activation path, omitting the shared-scale gradient can introduce a small systematic bias that accumulates over training. Our findings suggest that this backward path should be carefully considered when designing training algorithms with microscaling quantization.

Limitations. Our experiments cover two MoE architecture families and MXFP4/NVFP4, while broader architectures and microscaling formats remain to be studied. Joint system design and optimization analysis under further reduced precision for backward computation and communication, as well as alternative correction and surrogate designs, are left for future work.

AI USE STATEMENT

In this work, we used generative AI tools for implementation assistance, validation of experimental results, and figure plotting. We have not used generative AI tools for the core idea and experiment design. We have reviewed all AI-assisted work and results. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of AI.

ETHICS STATEMENT

This work studies optimization methods for low-precision model training and does not involve human-subject experiments. We do not identify additional ethical concerns specific to the proposed method beyond those generally associated with training and deploying large language models.

REPRODUCIBILITY STATEMENT

We provide model architectures, training configurations, and optimizer settings in App. A. The MXFP4 quantization details are in App. B. App. C derives the group-wise backward correction formula. The subsequent appendices show alternative scale surrogates, correction-location ablations, and additional optimization diagnostics. We use in-house training data and infrastructure. Within each model setting, all compared methods use the same training data and evaluation protocol.

REFERENCES

Anjulie Agrusa, Andrei Panferov, Elizabeth Wei, Keith Wyss, Paul Gibbons, Erik Schultheis, Tijmen Blankevoort, and Dan Alistarh. What matters for NVFP4 training? a scaling study of low-precision pre-training recipes. In *ICML 2026 Workshop on Resource-Adaptive Foundation Model Inference (AdaptFM)*, 2026. URL https://openreview.net/forum?id=jlkIyaG32w.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.298. URL https://aclanthology.org/2023.emnlp-main.298/.

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

AMD. AMD Instinct MI350 Series and Beyond: Accelerating the future of AI and HPC. AMD Blog, June 2025. URL https://www.amd.com/en/blogs/2025/amd-instinct-mi350-series-and-beyond-accelerating-the-future-of-ai-and-hpc.html.

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 *Advances in Neural Information Processing Systems*, volume 37, pp. 100213–100240. Curran Associates, Inc., 2024. doi: 10.52202/079017-3180. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/b5b939436789f76f08b9d0da5e81af7c-Paper-Conference.pdf.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020. URL https://arxiv.org/abs/2004.05150.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*, 2020. URL https://api.semanticscholar.org/CorpusID:208290939.

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.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 27056–27087. Curran Associates, Inc., 2024. doi: 10.52202/079017-0850. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/2f8ee6a3d766b426d2618e555b5aeb39-Paper-Conference.pdf.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Mengzhao Chen, Chaoyi Zhang, Jing Liu, Yutao Zeng, Zeyue Xue, Zhiheng Liu, Yunshui Li, Jin Ma, Jie Huang, Xun Zhou, and Ping Luo. Scaling law for quantization-aware training. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=fXr3uPr1G5.

Yuxiang Chen, Haocheng Xi, Jun Zhu, and Jianfei Chen. Oscillation-Reduced MXFP4 Training for Vision Transformers. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 9400–9414. PMLR, 2025. URL https://proceedings.mlr.press/v267/chen25bu.html.

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. In *Forty-third International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=7ZQhm5HnOA.

Xin Cheng, Rui Tian, Wangding Zeng, Damai Dai, Qinyu Chen, Bingxuan Wang, Zhenda Xie, Kezhao Huang, Xingkai Yu, Chengqi Deng, Shangyan Zhou, Chenggang Zhao, Zhewen Hao, Yukun Li, Han Zhang, Zhengyan Zhang, Yixu Wei, M. Y Xu, Huishuai Zhang, Dongyan Zhao, and Wenfeng Liang. Conditional memory via scalable lookup: A new axis of sparsity for large language models, 2026. URL https://arxiv.org/abs/2601.07372.

Musa Cim, Sarthak Arora, Poovaiah Palangappa, Miro Hodak, Ravi Dwivedula, Meena Arunachalam, and Mahmut Taylan Kandemir. Pretraining large language models with mxfp4 on native fp4 hardware, 2026. URL https://arxiv.org/abs/2605.09825.

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

Jack Cook, Junxian Guo, Guangxuan Xiao, Yujun Lin, Keith Wyss, Mahdi Nazemi, Asit Mishra, Carlo del Mundo, Tijmen Blankevoort, and Song Han. Four Over Six: More Accurate NVFP4 Quantization with Adaptive Block Scaling, 2026. URL https://arxiv.org/abs/2512.02010.

DeepSeek-AI, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, A. X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. Deepseek llm: Scaling open-source language models with longtermism, 2024. URL https://arxiv.org/abs/2401.02954.

DeepSeek-AI, Anyi Xu, B. Li, Bangcai Lin, Bing Xue, BingCheng Xian, Bingzheng Xu, Bochao Wu, Bowei Zhang, Boyi Deng, C. C. Yu, Chao Jin, Chaofan Lin, Chen Dong, Chenbing Wang, Chenfan Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyuan Zhang, Chenhao Xu, Chenqi Zhao, Chenze Shao, Chuhao Wang, Chuqi Zhang, Damai Dai, Dejian Yang, Deli Chen, Di Huang, Di Wu, Donghao Li, Erhang Li, Eric Fu, F. Zhou, Fangwei Zhou, Fangyun Lin, Fangzhou Yuan, Feiyu Xia, Fucong Dai, Guangbo Hao, Guanglin Li, Guanting Chen, Guoai Cao, Guofan Fan, Guolai Meng, Guowei Li, Haichuan Zhang, Haiyang Ma, Haiyang Shen, Han Li, Han Yu, Han Zhang, Hangyuan Deng, Hanwei Xu, Hanxiang Xu, Hanxun Zhong, Hao Guo, Hao Jiang, Hao Li, Hao Qin, Haodong Wen, Haofen Liang, Haofeng Huang, Haohua Liu, Haoling Zhang, Haoming Luo, Haoran Yang, Haotian Xu, Haotian Yuan, Haoting Huang, Haowen Luo, Haoyang Cai, Haoyu Chen, Haozhe Ji, Hengran Zhang, Hengrui Wang, Hengxu Wu, Honghui Ding, Hongxuan Tang, Huadong Wang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, J. Yang, J. H. Jin, J. H. Zhang, J. X. Zou, Jia Yu, Jiahui Zhou, Jiajun Chen, Jialiang Huang, Jialin Zhao, Jiamin Tang, Jian Zhou, Jianan Tong, Jianwen Li, Jiaqi Zhu, Jiarui Wang, Jiasheng Ye, Jiashi Li, Jiaxin Xu, Jiaying Ding, Jibai Lu, Jiewen Hu, Jin Yan, Jincheng Zhai, Jingchang Chen, Jingcheng Hu, Jingli Zhou, Jingsheng Xu, Jingting Xiang, Jingyan Yun, Jingyang Yuan, Jingyuan Cheng, Jinhua Zhu, Jinpeng Wang, Jinyi Chen, Jinyi Hu, Jiping Yu, Jueliang Guo, Junbo Pei, Junbo Sun, Junguang Jiang, Junjie Qiu, Junkang Zhou, Junqi Liu, Junren Li, Junxian Li, Junxiao Song, Junyi Guo, Kai Dong, Kaifeng Chen, Kaige Gao, Kang Guan, Kangdong Yuan, Ke Hong, Ke Xu, Kefan Zhao, Kexin Ji, Kexin Zhang, Kexing Zhou, Kuai Yu, Lan Zhang, Lean Wang, Lecong Zhang, Lei Wang, Letian Gao, Liang Zhao, Liansheng Xu, Lihua Guo, Lingxiao Luo, Lingyue Fu, Litao Deng, Litong Wang, Liyue Zhang, Longhao Chen, Lu Chen, Luotian Huang, Luyao Ma, Luyao Wang, M. S. Di, Max Mei, Menghao Ye, Miao Cui, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingjing Zhang, Mingqi Wei, Mingshu Chen, Mingxing Liu, Mingxu Zhou, Mingyu Xu, Mingyu Yang, Mingze Wang, Muyang Chen, Ni Shentu, Ning Wang, Niufang Ning, Panpan Huang, Peixin Cong, Peiyi Wang, Peiyuan Xin, Pengfei Ren, Pengfei Yan, Pengle Zhang, Qi Kang, Qi Tang, Qiancheng Wang, Qiang Li, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Qizhou Guo, Rongxian Xu, Rui Ding, Rui Hu, Rui Tian, Rui Yu, Ruidong Zhu, Ruifan Xu, Ruihan Yang, Ruihang Xia, Ruijie Lu, Ruilin Geng, Ruipeng Hong, Ruiqi Ge, Ruisong Zhang, Ruize Sun, Ruizhe Pan, Runji Wang, Runqian Chen, Runxin Xu, Ruohong Tian, Ruomeng Shen, Ruoyu Zhang, Ryan X., S. H. Liu, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoheng Nie, Shaoyuan Chen, Shengding Hu, Shengkai Lin, Shengwen Ran, Shengyu Liu, Shengyuan Jia, Shi Bai, Shi Feng, Shicheng Xu, Shichun Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shiyuan Feng, Shufan Gong, Shuhan Lin, Shuiping Yu, Shunfeng Zhou, Shuo Yang, Shuomeng Wang, Shuting Guo, Shuting Pan, Shuying Yu, Sinuo Cao, Siyi Lin, Sizhe Chen, Songyang Chen, Songyang Zhou, Tao Ni, Tao Yun, Tian Jin, Tian Pei, Tian Ye, Tianle Lin, Tianran Ji, Tianyi Cui, Tianyuan Yue, Tingting Yu, Tongrui Xiong, Wangding Zeng, Wei Liu, Wei Zhang, Weibin Xu, Weihao Zeng, Weilin Zhao, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjing Yao, Wenjun Gao, Wenkai Shao, Wenkai Yang, Wenli Zhang, Wenlu Wang, Wenlve Huang, Wenqian Yan, Wentao Zhang, Xi Gao, Xiang He, Xiang Li, Xiangli Li, Xiangwen Wang, Xiangying Zhang, Xiankui Wei, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojian Qu, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xiaoyao Zou, Xiaoyuan Li, Xicheng Guo, Xieting Chu, Xin Cheng, Xin Liu, Xin Xie, Xinbo Xu, Xingchao Liu, Xingchen Liu, Xingkai Yu, Xingyou Li, Xintong Yao, Xinyang Chen, Xinyong Jiang, Xinyu Yang, Xinyu Yang, Xu Chen, Xuanyu

Wang, Xubei Zhong, Xuecheng Su, Xuejie Liu, Xuheng Lin, Xujie Fan, Xuncheng Zhao, Xuwei Fu, Y. C. Yan, Y. H. Jiang, Y. T. Wu, Y. W. M., Y. Z. Wang, Yafei Gao, Yang Yang, Yang Zhang, Yanru Ma, Yanwen Huang, Yao Li, Yao Li, Yao Meng, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yaoyang Ye, Yehang Yin, Yexinrui Wu, Yi Qian, Yi Tao, Yi Yu, Yichao Zhang, Yichen Jiang, Yicheng Wang, Yifan Ding, Yifan Shi, Yifeng Peng, Yifeng Zhai, Yijia Wu, Yiliang Xiong, Yilun Wang, Ying He, Ying Zhou, Yingjia Luo, Yinmin Zhong, Yiping Wang, Yisong Wang, Yixiang Zhang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyao Yang, Yiyuan Liu, Yizai Cai, Yizhen Wei, Yizhi Wang, Yonglun Yang, Yongqi Zhuo, Yongqiang Guo, Yongtong Wu, Yu Wu, Yu Zhang, Yuan Bian, Yuan Cheng, Yuan Ou, Yuan Sun, Yuanfan Xu, Yuanhang Sun, Yuanhao Li, Yuchen Liu, Yuchen Yao, Yudong Han, Yuduan Wang, Yuhan Wu, Yuhao Meng, Yuheng Zou, YuKun Li, Yunchuan Wang, Yunfan Xiao, Yunfan Xiong, Yupeng Chen, Yuqian Cao, Yuqian Wang, Yuqing Chen, Yushun Zhang, Yutong Lin, Yuwei Xiao, Yuxian Gu, Yuxiang Chen, Yuxiang Huang, Yuxiang Luo, Yuxiang You, Yuxin Chen, Yuxin Xiang, Yuxuan Liu, Yuxuan Zhou, Yuyang Zhou, Yuzhe Guo, Yuzhen Huang, Yuzhuo Bai, Z. Y. Z., Zanlin Ni, Zehao Wang, Zehua Zhao, Zehui Ren, Zejun Zhao, Zhangli Sha, Zhanying Wang, Zhaochen Zhang, Zhaoshuai Du, Zhe Fu, Zhean Xu, Zhenda Xie, Zheng Liu, Zhengyan Zhang, Zhenhua Dong, Zhewen Hao, Zhibang Wang, Zhibin Gou, Zhicheng Ma, Zhihao Li, Zhihong Shao, Zhihuan Huang, Zhijie Li, Zhirui Lu, Zhixian Huang, Zhixuan Chen, Zhixuan Chen, Zhixuan Pan, Zhiyu Wu, Zhizhou Ren, Zhu He, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihui Gu, Zijia Zhu, Zili Zhang, Zilin Li, Zilong Hou, Zilong Lyu, Ziqiao Wang, Ziwei Xie, Ziya Zhang, Ziyi Gao, Zizheng Pan, Zonglin Li, Zongqing Yao, Zui Chen, Zuofan Wu, Chenchen Ling, Chengyu Hou, Chong Chen, D. Li, Di Qi, Dongjie Ji, Fang Wei, Fanyi Xia, Fei Xie, Feiyi Tan, Hailong Guo, Haiyan Zhai, Hui Zhou, Huihui Tan, Huijie Li, Jia Luo, Jia Song, Jialu Cai, Jian Liang, Jiangting Zhou, Jiaqi Gao, Jiayi Shao, Jie Chen, Jieyu Yang, Jin Chen, Jingde Zhang, Jingzi Zhou, Jinqian Wang, Jinyang Liu, JinZhao Sun, Junhua Ling, Junmin Zheng, Kaicheng Yang, Ke Xu, Le Su, Leyi Xia, Liangfeng Ding, Lin Zhuo, Linwang Ma, Linyan Zhu, Liyu Cai, Luqi Yao, M. K. Zhang, Meng Li, Miao Lin, Miaojun Wang, Min Zhang, Mingming Li, Mingming Wang, Mingze Yin, Minmin Han, Nan Cao, Ning Wang, Ningxin Ma, Panpan Wang, Peihan Lin, Peng Sun, Peng Zhang, Qian Ying, Qiang Xiang, Qiao Wang, Qingmiao Mao, Qiwei Jiang, Rongli Jin, Ruyi Chen, Sha Tao, Shangmian Sun, Shaoqing Wu, Shichao Zou, Si Lei, Tianyang Zhang, Tianyu Sun, Tingting Yin, W. L. Xiao, Wei An, Wei Li, Wei Wang, Weiwei Lin, Wenqing Hou, X. Lin, Xiangfei Meng, Xianzhu Huang, Xiao Peng, Xiaoqian Li, Xiaoting Zhang, Xiaowen Sun, Xiaoxiang Wang, Xiaoyu Ye, Xinrou Zhang, Xinyu Zhang, Xue Cao, Xueyin Chen, Yanan Zhou, Yanhong Xu, Yao Xia, Yao Xu, Yi Shao, Yihong Zhang, Yiling Ma, Ying Tang, Yining Lou, Yiru Chen, Yishi Piao, Yixuan Chen, Yong Xiong, Yuchen Xuan, Yuehan Yang, Yuer Xu, Yukun Zha, Yunxian Ma, Yuping Lin, Yuting Yan, Yutong Xie, Yuwen Sheng, Yuxuan Zhu, Zekai Zhang, Zhe Ju, Zhenzhen Lin, Zheren Gao, Zheyang Sun, Zhigang Yan, Zhongyu Wu, Zi Wang, Zihua Qu, Ziling Yan, and Ziyi Wan. Deepseek-v4.1-flash: Pushing the limits of kv cache compression, 2026a. URL https://arxiv.org/abs/2609.19969.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyu Hou, Chenhao Xu, Chenze Shao, Chong Ruan, Conner Sun, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Donghao Li, Dongjie Ji, Erhang Li, Fang Wei, Fangyun Lin, Fangzhou Yuan, Feiyu Xia, Fucong Dai, Guangbo Hao, Guanting Chen, Guoai Cao, Guolai Meng, Guowei Li, Han Yu, Han Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoling Zhang, Haoming Luo, Haoran Wei, Haotian Yuan, Haowei Zhang, Haowen Luo, Haoyu Chen, Haozhe Ji, Hengqing Zhang, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, J Yang, JQ Zhu, Jia Luo, Jia Song, Jia Yu, Jialiang Huang, Jialu Cai, Jian Liang, Jiangting Zhou, Jiasheng Ye, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jieyu Yang, Jin Chen, Jin Yan, Jingchang Chen, Jingli Zhou, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jingzi Zhou, Jinhua Zhu, Jiping Yu, Joseph Sun, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junmin Zheng, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Leyi Xia, Li Zhang, Liang Zhao, Lihua Guo, Lingxiao Luo, Linwang Ma, Linyan Zhu, Litong Wang, Liyu Cai, Liyue Zhang, Longhao Chen, MS Di, MY Xu, Max Mei, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Mingxu Zhou, Minmin Han, Ning Wang, Panpan Huang, Panpan Wang, Peixin Cong, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Qiwei Jiang, Rui Tian, Ruifan Xu, Ruijie Lu, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang,

Runqian Chen, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, Ruyi Chen, SH Liu, Shanghao Lu, Shangmian Sun, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoheng Nie, Shaoqing Wu, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Shuying Yu, Songyang Zhou, Tao Ni, Tao Yun, Tian Jin, Tian Pei, Tian Ye, Tianle Lin, Tianran Ji, Tianyi Cui, Tianyuan Yue, Tingting Yu, Tun Wang, W Zhang, WL Xiao, Wangding Zeng, Wei An, Weilin Zhao, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjing Yao, Wenjun Gao, Wenkai Yang, Wenlve Huang, Wenqing Hou, Wentao Zhang, Wenting Ma, Xi Gao, Xiang He, Xiangwen Wang, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingchen Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyu Zhang, Xu Chen, Xuanyu Wang, Xuecheng Su, Xueyin Chen, Xuheng Lin, Xuwei Fu, YC Yan, YQ Wang, YW Ma, Yanfeng Luo, Yang Zhang, Yanhong Xu, Yanru Ma, Yanwen Huang, Yao Li, Yao Li, Yao Xu, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Shao, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yijia Wu, Yiliang Xiong, Yiling Ma, Ying He, Ying Tang, Ying Zhou, Yingjia Luo, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiang Zhang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, YuKun Li, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuanhao Li, Yuduan Wang, Yuehan Yang, Yuer Xu, Yuhan Wu, Yuhao Meng, Yuheng Zou, Yukun Zha, Yunfan Xiong, Yupeng Chen, Yuping Lin, Yuqian Cao, Yuqian Wang, Yushun Zhang, Yuting Yan, Yutong Lin, Yuxian Gu, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuxuan Zhou, Yuyang Zhou, Yuzhen Huang, ZF Wu, Zehao Wang, Zehua Zhao, Zehui Ren, Zekai Zhang, Zhangli Sha, Zhe Fu, Zhe Ju, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zheren Gao, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhixuan Chen, Zhiyu Wu, Zhizhou Ren, Zhongyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihua Qu, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Ziyi Wan, Zizheng Pan, and Zongqing Yao. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026b. URL https://arxiv.org/abs/2606.19348.

Peijie Dong, Ruibo Fan, Yuechen Tao, Di Mou, Wenhu Hu, Zhenheng Tang, Yinghao Yu, Jiamang Wang, Wenbo Su, Guodong Yang, Liping Zhang, Xiaowen Chu, Baochun Li, and Bo Li. Dissecting outlier dynamics in LLM NVFP4 pretraining, 2026. URL https://arxiv.org/abs/2602.02047.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2368–2378, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1246. URL https://aclanthology.org/N19-1246/.

Mostafa Elhoushi and Jeff Johnson. any4: Learned 4-bit numeric representation for LLMs. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 15215–15231. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/elhoushi25a.html.

Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgO66VKDS.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, 2020. URL https://arxiv.org/abs/2101.00027.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021b. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

Robert Hu, Carlo Luschi, and Paul Balanca. Elucidating the design space of fp4 training, 2025. URL https://arxiv.org/abs/2509.17791.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, jiayi lei, Yao Fu, Maosong Sun, and Junxian He. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 62991–63010. Curran Associates, Inc., 2023. doi: 10.52202/075280-2749. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/c6ec1844bec96d6d32ae95ae694e23d8-Paper-Datasets_and_Benchmarks.pdf.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Jacob_Quantization_and_Training_CVPR_2018_paper.html.

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/.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, Jiyan Yang, Jongsoo Park, Alexander Heinecke, Evangelos Georganas, Sudarshan Srinivasan, Abhisek Kundu, Misha Smelyanskiy, Bharat Kaul, and Pradeep Dubey. A study of bfloat16 for deep learning training, 2019. URL https://arxiv.org/abs/1905.12322.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision – ECCV 2016*, pp. 235–251, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46493-0.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1082. URL https://aclanthology.org/D17-1082/.

Junghyup Lee, Dohyung Kim, and Bumsub Ham. Network quantization with element-wise gradient scaling. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6444–6453, 2021. doi: 10.1109/CVPR46437.2021.00638.

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.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 23439–23554, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/663bce02a0050c4a11f1eb8a7f1429d3-Paper-Conference.pdf.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177. URL https://aclanthology.org/2022.findings-acl.177/.

Minesh Mathew, Dimosthenis Karatzas, and C.V. Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2200–2209, January 2021.

Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. Overcoming oscillations in quantization-aware training. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16318–16330. PMLR, 2022. URL https://proceedings.mlr.press/v162/nagel22a.html.

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.

OpenAI. gpt-oss-120b & gpt-oss-20b Model Card. Technical report, OpenAI, 2025. URL https://cdn.openai.com/pdf/419b6906-9da6-406c-a19d-1bb078ac7637/oai_gpt-oss_model_card.pdf.

Andrei Panferov, Erik Schultheis, Soroush Tabesh, and Dan Alistarh. Quartet II: Accurate LLM Pre-Training in NVFP4 by Improved Unbiased Gradient Estimation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=CciWEZZDVb.

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, 2023. URL https://arxiv.org/abs/2310.10537.

Noam Shazeer. Glu variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, Xin Jiang, Wulong Liu, and Jun Yao. FlatQuant: Flatness Matters for LLM Quantization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 57587–57613. PMLR, 2025. URL https://proceedings.mlr.press/v267/sun25l.html.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

Soroush Tabesh, Mher Safaryan, Andrei Panferov, Alexandra Volkova, and Dan Alistarh. CAGE: Curvature-aware gradient estimation for accurate quantization-aware training. In *Ninth Conference on Machine Learning and Systems*, 2026. URL https://openreview.net/forum?id=Fubm1TtWeo.

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. *CoRR*, abs/2604.08826, 2026. doi: 10.48550/ARXIV.2604.08826. URL https://doi.org/10.48550/arXiv.2604.08826.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 87310–87356. Curran Associates, Inc., 2024. doi: 10.52202/079017-2771. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9ee3a664ccfeabc0da16ac6f1f1cfe59-Paper-Conference.pdf.

Albert Tseng, Tao Yu, and Youngsuk Park. Training LLMs with MXFP4. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 1630–1638. PMLR, 2025. URL https://proceedings.mlr.press/v258/tseng25a.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 62937–62957. PMLR, 2025. URL https://proceedings.mlr.press/v267/wang25ae.html.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 95266–95290. Curran Associates, Inc., 2024. doi: 10.52202/079017-3018. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ad236edc564f3e3156e1b2feafb99a24-Paper-Datasets_and_Benchmarks_Track.pdf.

xAI. Grok-1.5 vision preview, April 2024. URL https://x.ai/news/grok-1.5v.

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 *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 38087–38099. PMLR, 2023. URL https://proceedings.mlr.press/v202/xiao23c.html.

Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, Damai Dai, Huazuo Gao, Mingyu Xu, Kuai Yu, Liang Zhao, Shangyan Zhou, Zhean Xu, Zhengyan Zhang, Wangding Zeng, Shengding Hu, Yuqing Wang, Jingyang Yuan, Lean Wang, and Wenfeng Liang. mHC: Manifold-constrained hyper-connections. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=mDhyxu8WRb.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. MMMU-pro: A more robust multi-discipline multimodal understanding benchmark. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15134–15186, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.736. URL https://aclanthology.org/2025.acl-long.736/.

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

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/1e8a19426224ca89e83cef47f1e7f53b-Paper.pdf.

Qian Zhao, Kunlong Chen, Changxin Tian, Zhonghui Jiang, Haitao Zhang, Chaofan Yu, Peijie Jiang, Mingliang Gong, Jia Liu, Ziqi Liu, Zhiqiang Zhang, and Jun Zhou. Rethinking shrinkage bias in llm fp4 pretraining: Geometric origin, systemic impact, and ufp4 recipe, 2026. URL https://arxiv.org/abs/2606.20381.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen GONG, James Hoang, Armel Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, Prateek Yadav, Naman Jain, Alex Gu, Zhoujun Cheng, Jiawei Liu, Qian Liu, Zijian Wang, Binyuan Hui, Niklas Muennighoff, David Lo, Daniel Fried, Xiaoning Du, Harm de Vries, and Leandro Von Werra. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 66602–66656, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a6a90bcc2aa470c3871b2d39a67d26e8-Paper-Conference.pdf.

A Model Details and Training Configurations

MoE-3B: adapted from DeepSeek-V4. MoE-3B is adapted from the DeepSeek-V4 MoE (DeepSeek-AI et al., 2026b) including four-stream mHC (Xie et al., 2026), first-layer hash routing, and square-root softplus routing scores. We instead use a modified Attention design for MoE-3B without sparse attention: we adopt GQA (Ainslie et al., 2023) with 8 KV heads with learned per-head softmax-denominator biases (OpenAI, 2025). We use a SwiGLU clamp threshold of 10.

Scaling models: adapted from DeepSeek-V4.1-Flash. Our 10B–94B models adapt downscaled DeepSeek-V4.1-Flash (DeepSeek-AI et al., 2026a), retaining CSA2 and four-stream mHC (Xie et al., 2026) but omitting Engram (Cheng et al., 2026). We train the 10B–46B models on text only. The 94B model additionally uses a vision encoder and projector, with the encoder frozen throughout training; the dataset for MoE-94B contains $1/8$ multimodal and $7/8$ text-only data.

Optimizer details. AdamW (Loshchilov & Hutter, 2019) uses $(\beta_1, \beta_2) = (0.9, 0.95)$ and $\epsilon = 10^{-20}$. Muon (Jordan et al., 2024) uses Nesterov momentum 0.95, 10 Newton–Schulz iterations, and a learning-rate multiplier of 0.18. RMSNorms, token embeddings, and the LM head are optimized with AdamW with weight decay $= 0.1$.

Quantization settings. We apply weight and activation quantization to the two linear projections in routed experts, while keeping other components in higher precision. This is consistent with the predominantly Pre-FFN Norm scale growth observed in Sec. 2. Whether quantizing other modules, such as attention projections, induces similar optimization shifts remains a question for future work.

Loss reporting. The loss of multimodal MoE-94B includes both text-only and multimodal sequences. The recovery rates in Fig. 2 are from the averaged final-200-step loss. The plotted loss curves use moving averages over 200 / 500 steps. Faint lines in the figures show unsmoothed data.

Table 2: Model architectures and training configurations.

	MoE-3B *Ablations*	**MoE-10B**	**MoE-21B**	**MoE-46B**	**MoE-94B**
		Scaling experiments			
Architecture					
Transformer layers	12	16	22	22	34
Hidden dimension	1,280	1,024	1,536	2,304	2,304
Attention[a]	GQA	CSA2 (with SWA)			
Query heads	16	8	16	16	24
QK dims. (non-RoPE / RoPE)	128/64	448/64			
Sequence length	4,096	65,536			
FFN activation	SwiGLU	SwiGLU			
Per-expert FFN dimension	896	512	512	768	1,024
Experts (shared / routed / top-k)	2/64/6	1/384/6			
Parameter counts					
Total parameters	3.20B	10.05B	20.70B	46.00B	94.62B
Activated parameters	0.80B	0.41B	0.88B	1.56B	3.36B
Vision encoder (frozen)	—	—	—	—	0.41B
Visual projector	—	—	—	—	0.027B
Optimization and scheduler					
Peak learning rate	8.6×10^{-4}	5.7×10^{-4}	5.2×10^{-4}	4.7×10^{-4}	3.7×10^{-4}
Tokens per optimizer step	1.31M	1.57M	2.62M	4.19M	6.29M
Total optimizer steps	30,000	38,000	48,000	66,000	90,000
Nominal training tokens	39.3B	59.8B	125.8B	276.8B	566.2B
Warm-up steps	2,000	2,000			
Decay schedule[b]	2-stage	Cosine decay (final 30%)			
Final / peak learning rate	0.1	0.1			
Weight decay / grad. clipping	0.1 / none	0.1 / none			
Reference precision	BF16	MXFP8			

[a] GQA: grouped-query attention (Ainslie et al., 2023); SWA: sliding-window attention (Beltagy et al., 2020). MoE-3B uses 8 KV heads and learned softmax-denominator biases (OpenAI, 2025). CSA2 uses layer-local SWA and depth-adapted KV/index sharing.

[b] MoE-3B follows the two-stage multi-step learning-rate schedule of DeepSeek LLM (DeepSeek-AI et al., 2024).

* Other References: RoPE (Su et al., 2023); SwiGLU (Shazeer, 2020); MoE (Shazeer et al., 2017); MX formats (Rouhani et al., 2023).

B MXFP4 QUANTIZATION DETAILS

We provide details of the MXFP4 quantization used throughout this work. We follow the microscaling format of Rouhani et al. (2023) with the truncation-free scaling rule used in Chen et al. (2025).

Microscaling format. For an input matrix $\mathbf{X}$, MXFP4 partitions elements along the reduction dimension into groups of 32. For each grouped input $\mathbf{X}_g = \{X_{g,i}\}_{i=0}^{31}$, to be quantized, the quantized elements are represented in E2M1 and share a single power-of-two scale S_g represented in UE8M0. Here, $\mathrm{E}x\mathrm{M}y$ denotes a floating-point format with one sign bit, x exponent bits, and y mantissa bits, while $\mathrm{UE}x\mathrm{M}y$ denotes its unsigned counterpart.

Group-wise scale computation. Let

$$M_g := \max_{0 \le i < 32} |X_{g,i}|$$

denote the maximum absolute value of group g. We adopt the truncation-free MXFP4 scaling rule (Chen et al., 2025),

$$S_g = 2^{\left\lceil \log_2 \frac{M_g}{6} \right\rceil}, \tag{6}$$

where 6 is the largest finite magnitude representable by E2M1.

For a fixed nonzero scale S_g, Eq. (6) implies

$$3S_g < M_g \le 6S_g.$$

Therefore, the group absolute maximum remains within the representable E2M1 range after normalization by S_g. The scale changes only when $M_g/6$ crosses the boundary between two adjacent power-of-two intervals.

Quantization and dequantization. Given the shared scale S_g, each element is quantized as

$$\bar{X}_{g,i} = \mathrm{Round}_{\mathrm{E2M1}}\left(\frac{X_{g,i}}{S_g}\right), \qquad i = 0, \dots, 31.$$

The corresponding quantized-dequantized value is

$$\widehat{X}_{g,i} := Q(\mathbf{X})_{g,i} = S_g \bar{X}_{g,i}. \tag{7}$$

A group is therefore represented by its 32 E2M1 values together with a shared UE8M0 scale, $\left(\{\bar{X}_{g,i}\}_{i=0}^{31}, S_g\right)$. Throughout the paper, we use $\widehat{\mathbf{X}} = Q(\mathbf{X})$ to denote the quantized-dequantized tensor in Eq. (7) for mathematical convenience.

For the NVFP4 format in the ablation study (Sec. 4.4), we follow the details of Alvarez et al. (2025): we use E4M3 per-group scale factors and a global FP32 scale. We apply the gradient correction to the group-level E4M3 scales, treating the global FP32 scale as constant in the backward pass.

C DERIVATION OF THE TETRABACK GRADIENT

We provide the full derivation of the TetraBack correction in Eq. (4). We use the MXFP4 notation defined in App. B. For each group $\mathbf{X}_g = \{X_{g,i}\}_{i=0}^{31}$, let $\widehat{\mathbf{X}}_g = Q(\mathbf{X})_g$ denote its quantized-dequantized output, M_g denote its absolute maximum, and S_g denote its shared scale.

C.1 GRADIENT DECOMPOSITION

Define the incoming gradient

$$G_{g,i} := \frac{\partial \mathcal{L}}{\partial \widehat{X}_{g,i}}.$$

The total input gradient is

$$\frac{\partial \mathcal{L}}{\partial X_{g,j}} = \sum_{i=0}^{31} G_{g,i} \frac{\partial \widehat{X}_{g,i}}{\partial X_{g,j}}.$$

As the shared scale depends on the group input, $\widehat{X}_{g,i}$ depends on $X_{g,j}$ both directly and through S_g:

$$\frac{\partial \widehat{X}_{g,i}}{\partial X_{g,j}} = \left.\frac{\partial \widehat{X}_{g,i}}{\partial X_{g,j}}\right|_{S_g} + \frac{\partial \widehat{X}_{g,i}}{\partial S_g}\frac{\partial S_g}{\partial M_g}\frac{\partial M_g}{\partial X_{g,j}}. \tag{8}$$

Here the first term differentiates with respect to $X_{g,j}$ while treating S_g as a constant. Since the MXFP4 adopts the truncation-free scaling (Chen et al., 2025), applying the identity STE to the direct quantization path gives

$$\left.\frac{\partial \widehat{X}_{g,i}}{\partial X_{g,j}}\right|_{S_g} \approx \mathbf{1}\{i = j\}.$$

Substituting this into Eq. (8) yields

$$\frac{\partial \mathcal{L}}{\partial X_{g,j}} \approx G_{g,j} + \left[\sum_{i=0}^{31} G_{g,i}\frac{\partial \widehat{X}_{g,i}}{\partial S_g}\right]\frac{\partial S_g}{\partial M_g}\frac{\partial M_g}{\partial X_{g,j}}. \tag{9}$$

The first term is the conventional matrix-wise STE, while the second term is the scale-dependent path restored by TetraBack.

C.2 Derivatives Along the Scale Path

Derivative of the quantized output with respect to the scale. From App. B,

$$\widehat{X}_{g,i} = S_g \,\mathrm{Round}_{\mathrm{E2M1}}\left(\frac{X_{g,i}}{S_g}\right).$$

Applying an identity STE to E2M1 rounding while retaining the derivatives of normalization and rescaling gives

$$\frac{\partial \widehat{X}_{g,i}}{\partial S_g} \approx \bar{X}_{g,i} + S_g\left(-\frac{X_{g,i}}{S_g^2}\right) = \frac{\widehat{X}_{g,i} - X_{g,i}}{S_g}. \tag{10}$$

This has the same algebraic form as the step-size derivative in LSQ (Esser et al., 2020) in the non-clipped regime (in MXFP4, we adopt the truncation-free scaling, so there is no clipping-mask in STE): the scale gradient is determined by the quantization residual. Here, however, S_g is an input-dependent shared scale rather than a learnable quantization parameter, so this derivative *should* contribute to the upstream activation gradient through the scale path.

Surrogate derivative of the shared scale. The exact MXFP4 scale is a power-of-two staircase function of M_g and therefore has zero derivative almost everywhere. We instead use a surrogate $\widetilde{S}_g(M_g)$ only for the backward derivative:

$$\frac{\partial S_g}{\partial M_g} \approx \widetilde{S}'_g(M_g).$$

App. B shows that a fixed forward scale S_g corresponds to

$$3S_g < M_g \le 6S_g.$$

Our default surrogate passes linearly through the midpoint $(4.5S_g, S_g)$ of each interval:

$$\widetilde{S}_g(M_g) = \frac{M_g}{4.5}, \qquad \widetilde{S}'_g(M_g) = \frac{1}{4.5}. \tag{11}$$

Only the derivative in Eq. (11) is used during backward; the forward pass always uses the exact MXFP4 scale. Alternative surrogate choices are studied in App. D.

Derivative of the group absmax. Let

$$j_g^* := \arg\max_i |X_{g,i}|.$$

For a unique maximum,

$$\frac{\partial M_g}{\partial X_{g,j}} = \mathrm{sign}(X_{g,j})\mathbf{1}\{j = j_g^*\}. \tag{12}$$

At a nonzero tie, we select the first maximizing element, corresponding to one valid subgradient of the absmax operation.

C.3 Combined Input Gradient

Substituting Eqs. (10), (11), and (12) into Eq. (9) gives

$$\frac{\partial \mathcal{L}}{\partial X_{g,j}} \approx \frac{\partial \mathcal{L}}{\partial \widehat{X}_{g,j}} + \mathbf{1}\{j = j_g^*\} \frac{\operatorname{sign}(X_{g,j}) \widetilde{S}_g'(M_g)}{S_g} \sum_{i=0}^{31} \frac{\partial \mathcal{L}}{\partial \widehat{X}_{g,i}} \left(\widehat{X}_{g,i} - X_{g,i}\right), \tag{13}$$

which is Eq. (4) in the main text.

Group-wise form. The additional gradient can be written as

$$\Delta \mathrm{g}_{X_{g,j}} = \mathbf{1}\{j = j_g^*\} \frac{\operatorname{sign}(X_{g,j}) \widetilde{S}_g'(M_g)}{S_g} \left\langle \frac{\partial \mathcal{L}}{\partial \widehat{\mathbf{X}}_g}, \widehat{\mathbf{X}}_g - \mathbf{X}_g \right\rangle. \tag{14}$$

This form makes explicit that the correction requires only one group-wise length-32 inner product.

D Differentiable Surrogates for the Scale Factor in MXFP4

For a nonzero group $\mathbf{X}_g = \{X_{g,i}\}_{i=0}^{31}$, let $\widehat{\mathbf{X}}_g = Q(\mathbf{X})_g$ denote its quantized-dequantized output, M_g denote its absolute maximum, and S_g denote its shared scale. Recall that

$$S_g = 2^{\lceil \log_2(M_g/6) \rceil} =: \mathrm{Pow2Ceil}(M_g/6).$$

We propose three different backward surrogates for $\partial S_g / \partial M_g$:

- **FullSTE.** Treat the entire power-of-two scale selection as an identity STE applied to $M_g/6$:

$$S_g = \mathrm{Pow2Ceil}(M_g/6) \quad \Longrightarrow \quad \frac{\partial S_g}{\partial M_g} \approx \frac{1}{6}.$$

- **ExpSTE.** Apply the identity STE only to the *ceiling* operation, while retaining the derivatives of the logarithm and exponential:

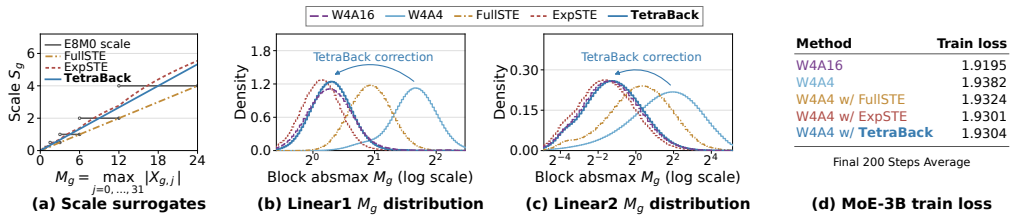
$$\frac{\partial S_g}{\partial M_g} \approx S_g \ln 2 \cdot \frac{\partial}{\partial M_g} \log_2(M_g/6) = \frac{S_g}{M_g}.$$

- **MidSTE (TetraBack).** For a fixed forward scale S_g, we have $3S_g < M_g \le 6S_g$. We approximate the ExpSTE derivative S_g/M_g by its value at the midpoint of this interval, $M_g = 4.5S_g$:

$$\frac{\partial S_g}{\partial M_g} \approx \left.\frac{S_g}{M_g}\right|_{M_g = 4.5 S_g} = \frac{1}{4.5}.$$

  Equivalently, this corresponds to the local linear surrogate $\widetilde{S}_g(M_g) = M_g/4.5$ passing through the midpoint of each scale interval.

Here, the corresponding $\widetilde{S}_g(M_g)$ functions are shown in Fig. 12.



(a) Scale surrogates **(b) Linear1 M_g distribution** **(c) Linear2 M_g distribution** **(d) MoE-3B train loss**

Figure 12: Scale-gradient estimators in MoE-3B training. **(a)** FullSTE, ExpSTE, and MidSTE (**TetraBack**) use backward surrogate derivatives $1/6$, S_g/M_g, and $1/4.5$, respectively. The curves show the corresponding surrogate functions. **(b,c)** Group-absolute-maximum distributions at the final checkpoints. Histograms use $\log_2 M_g$ bins. **(d)** Train loss averaged over the final 200 steps.

All three scale-gradient surrogates improve over standard W4A4 training (Fig. 12), showing that the benefit is *not specific to the particular surrogate* used by TetraBack, but primarily comes from restoring the missing scale-dependent gradient path.

The choice of surrogate nevertheless affects the correction *strength*. FullSTE uses the constant derivative $1/6$, which is no larger than the ExpSTE derivative $S_g/M_g \in [1/6, 1/3)$ within each scale interval, and accordingly only partially restores the shifted M_g distributions in Fig. 12(b,c). In contrast, ExpSTE and MidSTE bring the distributions substantially closer to W4A16 and achieve nearly identical training losses, with MidSTE providing a closer match to the W4A16 M_g distributions. We therefore use MidSTE as the default for its simple constant derivative. The comparable behavior of ExpSTE further suggests that the effectiveness of TetraBack does not rely on the exact choice of the MidSTE derivative.

E Why Apply TetraBack Only to Activation Quantization?

Activation-side versus weight-side correction. Although the missing scale-dependent gradient exists for both activation and weight microscaling, we apply TetraBack *only to activation quantization*. Empirically, the weight-side correction has only a marginal effect under W4A16 and does not further improve training for W4A4 (Tab. 3). This is also consistent with the observation in Sec. 2 that weight-only quantization *does not* induce the abnormal Norm and activation trajectories: W4A16 remains close to the unquantized-activation baseline, whereas the drift emerges only after activation quantization is introduced.

Table 3: Activation- and weight-side gradient corrections in MoE-3B training. Training loss is averaged over the final 200 steps.

Precision	Act. corr.	Weight corr.	Train loss ↓
W4A16	×	×	1.9195
W4A16	×	✓	1.9186
W4A4	×	×	1.9382
W4A4	✓	×	**1.9304**
W4A4	✓	✓	1.9315

Why does activation-side correction matter more? A key difference is their position in the computation graph. The input to an activation quantizer is an intermediate activation, so its input-gradient bias continues to backpropagate through the main computation path to upstream modules, including the Pre-FFN Norms analyzed in Sec. 4.5. Consequently, a small systematic bias in this gradient can propagate to persistent upstream parameters and accumulate over training. In contrast, a weight-quantizer correction acts directly on the corresponding master-weight gradient, without propagating through upstream network modules. This distinction may help explain the different empirical effects. We leave a causal characterization of this asymmetry to future work.

F More Analysis on the Effect of TetraBack on W4A4 Training

F.1 Does the Optimization Shift Strengthen with Quantization Error?

As shown in Sec. 4.5, the gradient correction at the Linear layer input quantizers back-propagates upstream and contributes to the gradients at the same-layer Pre-FFN Norm; this propagated correction mainly *counteracts* the abnormal scale-growth tendency observed in standard W4A4 training, as described in Sec. 2.

This contribution is stronger for `Linear2`, whose activation quantization also induces a larger Norm shift and training-loss gap, as shown in Sec. 4.5. We therefore vary the quantization error at the `Linear2` input to examine how the optimization shift depends on the retained error magnitude:

$$\widehat{\mathbf{X}}_\lambda = (1-\lambda)\mathbf{X} + \lambda Q(\mathbf{X}), \qquad \lambda \in \{0, 0.5, 0.75, 0.9, 1\}.$$

Here λ controls the fraction of quantization error retained. Weights remain MXFP4, and `Linear1` activations remain BF16. Thus $\lambda = 0$ recovers W4A16, whereas $\lambda = 1$ quantizes only the `Linear2` activation input, rather than both Linear inputs as in the full W4A4 setting. The identity backward estimator is retained, and scale-path correction is disabled in this experiment.

As λ increases, both the Pre-FFN Norm shift and the training-loss gap increase (Fig. 13). This demonstrates that the strength of the optimization bias grows with the activation quantization error.

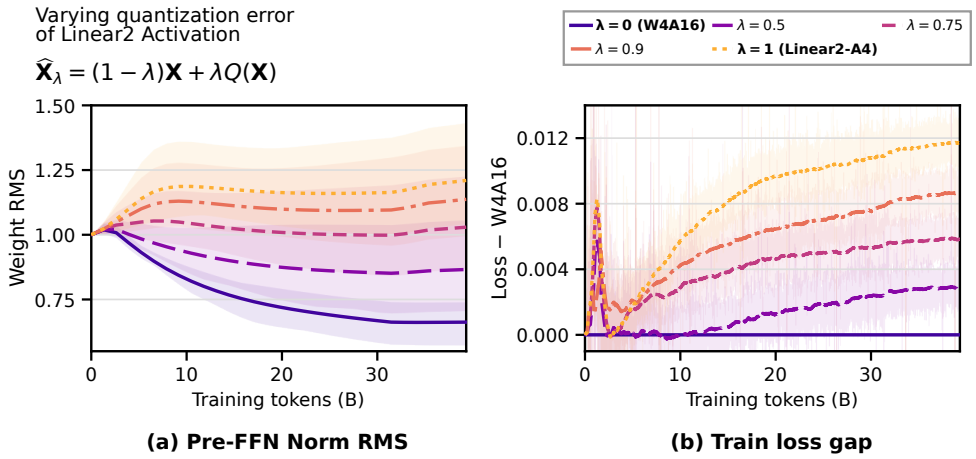


Figure 13: Varying the `Linear2` activation-quantization error in MoE-3B without TetraBack correction. Larger error produces greater Pre-FFN Norm scale elevation and a larger training-loss gap relative to $\lambda = 0$. Linear1 activations and all other precision settings are unchanged.

F.2 Does the Effect Persist without SwiGLU Clamping?

Since the model architectures in this paper adopt the SwiGLU Clamp from DeepSeek-V4 (DeepSeek-AI et al., 2026b) which clamps the SwiGLU operands to ensure the `Linear2` inputs are bounded, we conduct additional experiments on MoE-3B models with ordinary, non-clamped SwiGLU. The data, optimizer, and 30k-step schedule setting are unchanged. Quantization remains confined to routed experts, with BF16 backward computation.

The optimization bias is *even worse* without clamping (Fig. 14) compared with enabling clamping (in Sec. 2). Without gradient correction, standard W4A4 reaches $\sim 10^2\times$ the MoE output RMS and $\sim 6\times$ the pre-FFN Norm weight RMS of the W4A16 and BF16 baselines. With TetraBack, this abnormal growth is curbed and the W4A4 training trajectories are recovered close to the baselines. TetraBack also lowers the training loss, reducing the gap to the W4A16 and BF16 baselines. These results extend its effectiveness to architectures with ordinary, unclamped SwiGLU.

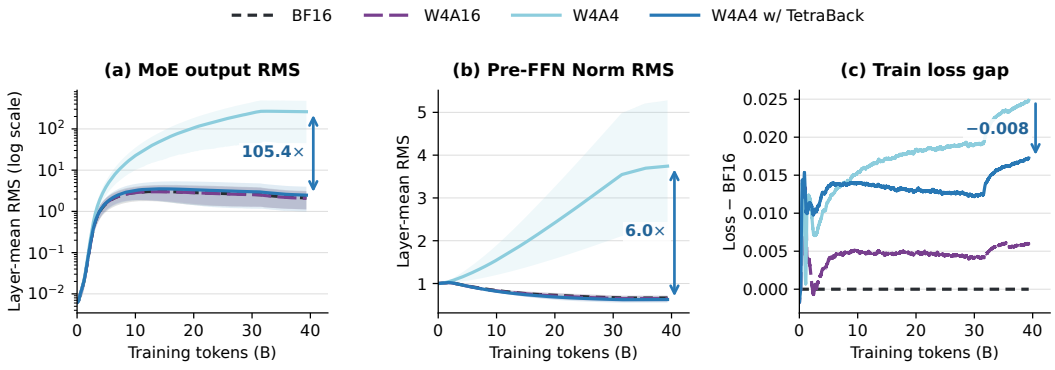


Figure 14: MoE-3B training with ordinary, unclamped SwiGLU. (a) MoE output RMS; (b) Pre-FFN Norm weight RMS; (c) loss gap to the unclamped BF16 run. Shading shows ± 1 layer-wise SD.

F.3 What is the Backpropagated Effect of the Gradient Correction?

We extend Sec. 4.5 to both Pre-FFN and Pre-Attention Norms. On an uncorrected W4A4 training trajectory, we enable the correction at one Linear input at a time and trace its effect on upstream Norm gradients. Following notation in Sec. 4.5, $\mathbf{X}_i$ is the selected Linear input in source layer i,

and $\boldsymbol{\gamma}_j$ is the selected Norm weight in target layer j. The local correction propagates as

$$\Delta \mathbf{g}_{j \leftarrow i} = \left(\frac{\partial \operatorname{vec}(\mathbf{X}_i)}{\partial \boldsymbol{\gamma}_j^\top} \right)^\top \operatorname{vec}(\Delta \mathbf{g}_{\mathbf{X}_i}).$$

where $\Delta \mathbf{g}_{\mathbf{X}_i}$ is the correction term described in Sec. 4.5. We then use the same metric in Sec. 4.5 to measure the correction effect:

$$H_{j \leftarrow i} = -\mathbb{E}_t \left[\frac{\langle \Delta \mathbf{g}_{j \leftarrow i, t}, \boldsymbol{\gamma}_{j,t} \rangle}{\|\boldsymbol{\gamma}_{j,t}\|_2^2} \right]. \tag{15}$$

Here $\mathbb{E}_t$ averages over steps 2000–2499 of the 30k-step training schedule. Negative H indicates a tendency toward Norm scale *shrinkage* under gradient descent.

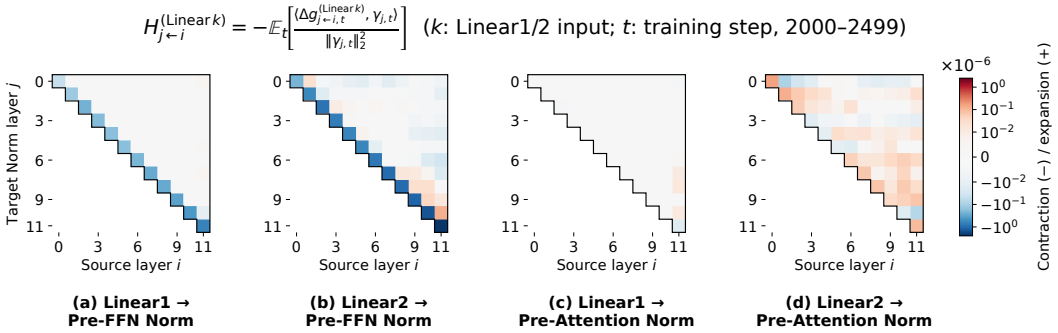


Figure 15: Layer-wise propagation of a single-source correction in MoE-3B, averaged over 500 steps of one training trajectory. The effect is measured by Eq. (15).

Both Linear inputs contribute shrinkage-directed gradients to their same-layer Pre-FFN Norm, counteracting the abnormal scale growth observed in Sec. 2. The contribution is larger for `Linear2`. Pre-Attention Norm responses are weaker and have mixed signs, consistent with the absence of scale growth in these Norms under W4A4 (Sec. 2). The strongest effect is therefore localized to the Norm immediately preceding the quantized FFN.

F.4 The Complete Group-wise Chain Rule for Microscaling

In this subsection, we expand the comparison in Sec. 3 between the activation backward in Hu et al. (2025) and the complete group-wise chain rule used by TetraBack. We focus on activation quantization; the distinction between activation- and weight-side corrections is discussed in App. E.

The incomplete group-wise chain rule in Hu et al. (2025). In Sec. 2 ("A common framework for FP4 training") of Hu et al. (2025), they derive $\partial \widehat{X}_{g,j} / \partial X_{g,j}$ in Proposition 1, expressed here in our notation. Their loss-gradient formula after that proposition corresponds to a per-group form:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{X}_g} \approx \frac{\partial \mathcal{L}}{\partial \widehat{\mathbf{X}}_g} \odot \left(\frac{\partial \widehat{X}_{g,j}}{\partial X_{g,j}} \right)_{j=0}^{31}. \tag{16}$$

The error is the Hadamard product: it pairs each output gradient only with the derivative of that output with respect to its corresponding input. An input also affects other outputs through the shared scale, and those contributions are missing. Thus, the issue is the loss-gradient expression after Proposition 1, rather than the individual derivative in the proposition.

For a nonzero group, the quantized-dequantized output is $\widehat{X}_{g,i} = S_g \operatorname{Round}_{\text{E2M1}}(X_{g,i}/S_g)$, where S_g depends on $M_g = \max_i |X_{g,i}|$. Their elementwise loss-gradient expression retains only the first of the following scale paths:

$$\textit{Hu et al. (2025):} \quad X_{g,j} \to M_g \to S_g \to \widehat{X}_{g,j},$$
$$\textbf{\textit{TetraBack} (ours):} \quad X_{g,j} \to M_g \to S_g \to \widehat{\mathbf{X}}_g = \{\widehat{X}_{g,i}\}_{i=0}^{31}.$$

The complete backward must include every output reached through S_g. The standard matrix-wise STE, commonly used in MXFP4/NVFP4 training recipes (NVIDIA et al., 2026; Cim et al., 2026; Chen et al., 2026b; Panferov et al., 2026), omits the scale path entirely. Keeping the fixed-scale derivative explicit, the three backward estimates are

$$\left(\frac{\partial \mathcal{L}}{\partial X_{g,j}}\right)_{\text{STE}} \approx \frac{\partial \mathcal{L}}{\partial \widehat{X}_{g,j}} \left.\frac{\partial \widehat{X}_{g,j}}{\partial X_{g,j}}\right|_{S_g} \qquad \times\times \text{ No scale path}$$

$$\left(\frac{\partial \mathcal{L}}{\partial X_{g,j}}\right)_{\text{diag}} \approx \frac{\partial \mathcal{L}}{\partial \widehat{X}_{g,j}} \left.\frac{\partial \widehat{X}_{g,j}}{\partial X_{g,j}}\right|_{S_g} + \frac{\partial \mathcal{L}}{\partial \widehat{X}_{g,j}} \frac{\partial \widehat{X}_{g,j}}{\partial S_g} \frac{\partial S_g}{\partial X_{g,j}} \qquad \times \text{ Incomplete}$$

$$\left(\frac{\partial \mathcal{L}}{\partial X_{g,j}}\right)_{\text{full}} \approx \frac{\partial \mathcal{L}}{\partial \widehat{X}_{g,j}} \left.\frac{\partial \widehat{X}_{g,j}}{\partial X_{g,j}}\right|_{S_g} + \left(\sum_{i=0}^{31} \frac{\partial \mathcal{L}}{\partial \widehat{X}_{g,i}} \frac{\partial \widehat{X}_{g,i}}{\partial S_g}\right) \frac{\partial S_g}{\partial X_{g,j}} \qquad \checkmark \text{ Complete (ours)} \tag{17}$$

The notation $\left.\partial \widehat{X}_{g,j}/\partial X_{g,j}\right|_{S_g}$ means differentiating with respect to $X_{g,j}$ while holding S_g fixed at its forward value. It isolates the direct path, which is common to all three expressions; the additional terms account for the input dependence of S_g. Here, STE, diag, and full denote the standard STE, the diagonal-only correction corresponding to Hu et al. (2025), and TetraBack, respectively.

Backward rules used in the comparison. Following App. C, we compare all three variants using an identity STE for E2M1 rounding under truncation-free scaling:

$$\left.\frac{\partial \widehat{X}_{g,j}}{\partial X_{g,j}}\right|_{S_g} = \text{Round}'_{\text{E2M1}}\left(\frac{X_{g,j}}{S_g}\right) \approx 1, \qquad \frac{\partial \widehat{X}_{g,i}}{\partial S_g} \approx \frac{\widehat{X}_{g,i} - X_{g,i}}{S_g}.$$

Following Sec. 3, both correction variants also use $\widetilde{S}'_g(M_g) = 1/4.5$ and $j_g^* = \arg\max_i |X_{g,i}|$, selecting the first maximizing element at a tie. The full backward in Eq. (17) then recovers Eq. (4), whereas the diagonal-only backward omits

$$\left(\frac{\partial \mathcal{L}}{\partial X_{g,j}}\right)_{\text{full}} - \left(\frac{\partial \mathcal{L}}{\partial X_{g,j}}\right)_{\text{diag}} = \frac{\mathbf{1}\{j = j_g^*\}\,\text{sign}(X_{g,j})}{4.5 S_g} \sum_{\substack{i=0 \\ i \neq j}}^{31} \frac{\partial \mathcal{L}}{\partial \widehat{X}_{g,i}} (\widehat{X}_{g,i} - X_{g,i}).$$

Both corrections act only on the abs-max input, but their values generally differ: the full correction aggregates contributions from all outputs in the group, whereas the diagonal-only correction uses only the corresponding output.

Controlled ablation and training loss. We compare the three backward variants in MoE-3B training, using identical MXFP4 forward quantization and training settings. Weight quantization retains the standard STE. The corrections use the matched backward rules above; W4A16 provides the activation-unquantized reference. This isolates cross-element aggregation, rather than reproducing the full recipe of Hu et al. (2025). Tab. 4 shows that TetraBack achieves the lowest training loss among the three W4A4 variants.

Table 4: Final-200-step averaged training loss on MoE-3B.

Setting	$\mathcal{L}_{\text{train}} \downarrow$	$\mathcal{L} - \mathcal{L}_{\text{W4A16}}$
W4A16	1.9195	—
W4A4 (standard STE)	1.9382	+0.0187
W4A4 w/ Diag. Corr.	1.9361	+0.0166
W4A4 w/ **TetraBack**	**1.9304**	**+0.0109**

Norm gradients and update dynamics. Following Sec. 3, for Norm weights $\boldsymbol{\gamma}_{\ell,t}$ and their gradients $\mathbf{g}_{\ell,t}$ at layer ℓ and step t, we measure

$$r_{\ell,t} = -\frac{\langle \mathbf{g}_{\ell,t}, \boldsymbol{\gamma}_{\ell,t} \rangle}{\|\boldsymbol{\gamma}_{\ell,t}\|_2^2}, \qquad u_{\ell,t} = \log \frac{\text{RMS}(\boldsymbol{\gamma}_{\ell,t+1})}{\text{RMS}(\boldsymbol{\gamma}_{\ell,t})}.$$

Positive $r_{\ell,t}$ indicates a gradient-descent tendency toward Norm weight growth; positive $u_{\ell,t}$ records actual growth after the optimizer step. Negative values indicate the corresponding shrinkage.

Fig. 16 tracks these metrics and Norm scales throughout the training of MoE-3B (39.32B tokens). Both the standard STE and diagonal-only correction retain outward gradient tendencies and elevated Pre-FFN Norm scales across layers. TetraBack brings the gradients, actual updates, and Norm scales closer to W4A16, while Pre-Attention Norm scales remain similar across variants, which is consistent with the phenomenon in Sec. F.3. Thus, the incomplete scale correction leaves substantial Norm drift in this setting.

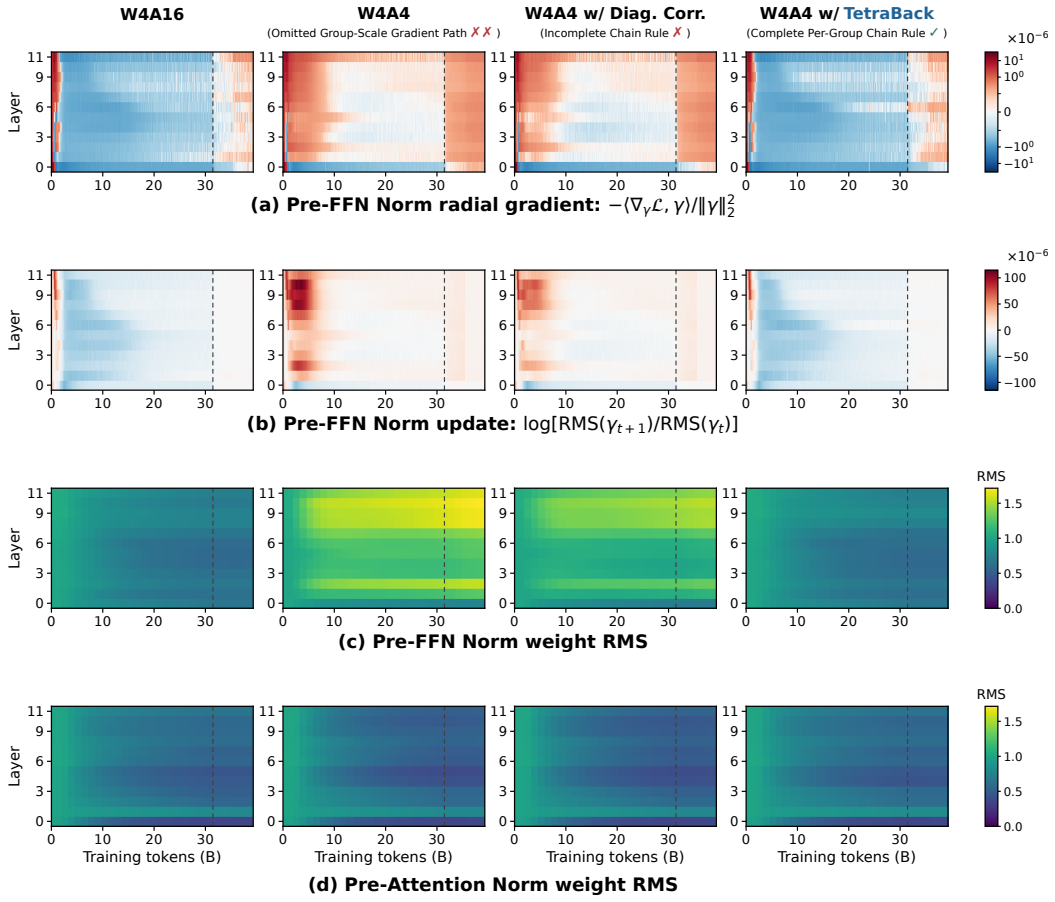
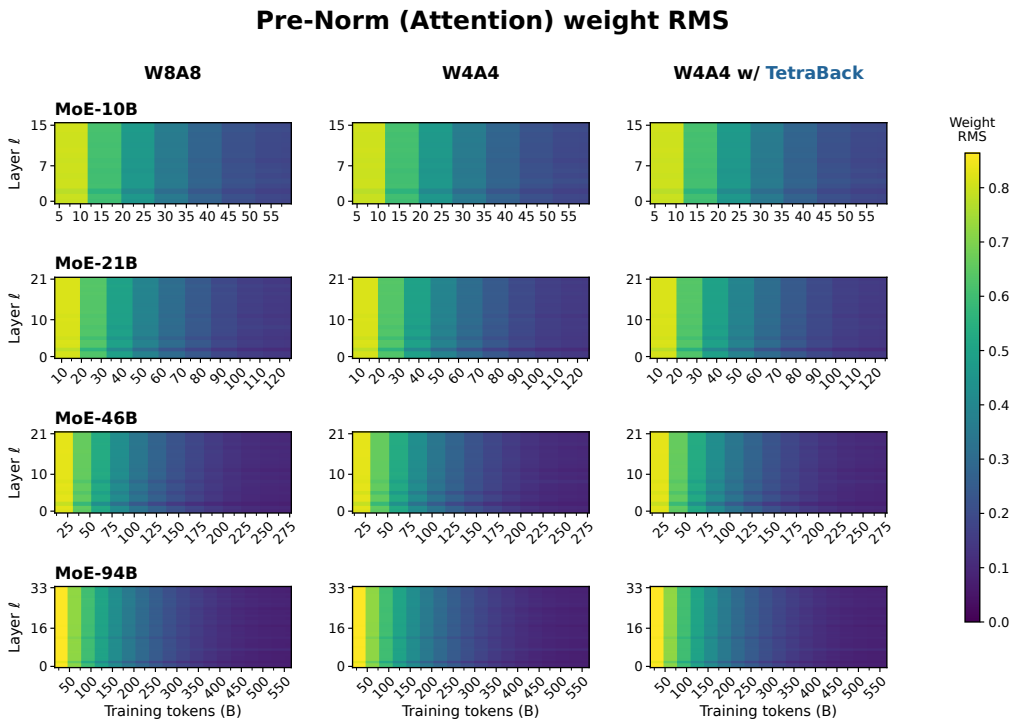


Figure 16: Layer-wise Norm dynamics in MoE-3B under W4A16, standard W4A4, diagonal-only correction according to the chain rule in Hu et al. (2025), and TetraBack. (a) Pre-FFN Norm radial gradients; (b) actual log-RMS updates; (c) Pre-FFN Norm weight RMS; (d) Pre-Attention Norm weight RMS. Each cell in (a,b) averages 100 consecutive steps at one layer. Dashed lines mark the start of learning-rate decay.

F.5 Layer-wise Norm Dynamics Across Model Scales

We extend the Norm-scale analysis in Fig. 16 from MoE-3B to the text-only 10B, 21B, and 46B models and the multimodal 94B model. Fig. 17 compares both Pre-Attention and Pre-FFN Norm weight RMS across layers and training checkpoints. W8A8 serves as a higher-precision reference, while the two W4A4 settings isolate the effect of the backward correction.

Across model sizes, Pre-Attention Norm weight RMS follows similar trajectories in all three settings. In contrast, standard W4A4 produces persistently elevated Pre-FFN Norm weight RMS across layers, while TetraBack brings these trajectories closer to W8A8. This extends the pattern observed in MoE-3B: the abnormal Norm-scale growth remains concentrated before the quantized FFNs, and the correction is effective across model scales.



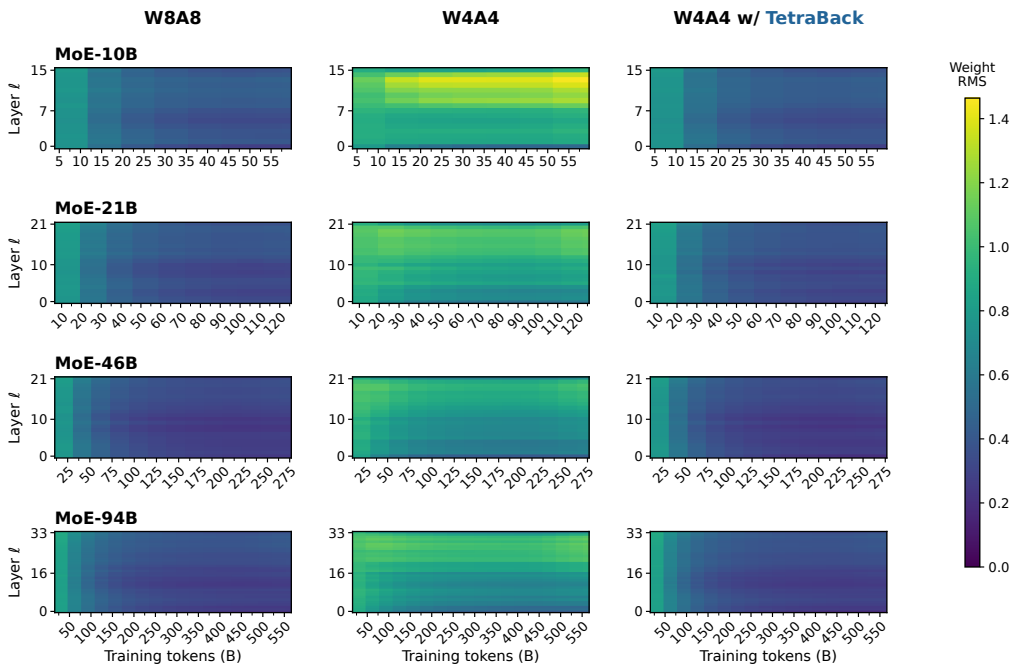


Figure 17: Pre-Attention Norm and Pre-FFN Norm weight RMS across layers and training tokens. Rows show 10B, 21B, 46B, and 94B; columns show W8A8, W4A4, and W4A4 with TetraBack. Color scales are linear and shared within each block. Cells show recorded checkpoints, without value interpolation.