What Matters for NVFP4 Training? A Scaling Study of Low-Precision Pre-Training Recipes

Anonymous Authors[1]

Abstract

Training large language models directly in 4-bit floating-point (FP) formats promises substantial improvements in throughput and energy efficiency. While some recipe design choices have been validated at scale, many promising approaches remain untested beyond small models and short token horizons, leaving open the question of which trends will hold. We present a systematic comparison of recent NVFP4 recipes at medium-model scale, up to 8B dense and 30B-A3B MoE models, trained up to 1T tokens and focus on which ingredients are necessary for accuracy recovery. We propose a final recipe grounded in the principles behind established stable NVFP4 training at scale, incorporating state-of-the-art techniques such as unbiased gradient estimation with lower quantization error than stochastic rounding. To the best of our knowledge, this is the strongest FP4 training result demonstrated at this scale to date in loss gap to BF16. Through ablation studies, we find that: (i) each technique in the optimized recipe measurably improves loss trajectory, (ii) selective high-precision layers are necessary for recovering accuracy at scale, (iii) not all tensors in the backward pass benefit equally from de-biasing, leaving room to apply complementary error-reduction techniques to the remaining tensors.

1. Introduction

The computational cost of training state-of-the-art foundation models has been increasing rapidly (Amodei & Hernandez, 2018; Sevilla et al., 2022). Since pre-training modern Transformer-based models is dominated by dense matrix multiplications, reducing the precision of these operations is one of the most direct levers for improving training efficiency. This has motivated a progression from FP16/BF16 to FP8 training (Micikevicius et al., 2022), and, more recently, to hardware-supported 4-bit microscaling formats (Rouhani et al., 2023) such as MXFP4 and NVFP4 on NVIDIA Blackwell GPUs (NVIDIA, 2025).

Yet, quantized training introduces noise, and in the 4-bit regime this noise could affect convergence, even when operations may appear accurate. A recent line of work has therefore focused on recipes that recover most of the accuracy of FP8 or BF16 training while preserving the speedups of 4-bit tensor-core execution. Examples include fully-quantized FP4 training (Chmiel et al., 2025), the NVIDIA '25 NVFP4 recipe (NVIDIA et al., 2025), TetraJet-v2 (Chen et al., 2025), Four Over Six (Cook et al., 2025), Quartet (Castro et al., 2025), and Quartet II (Panferov et al., 2026). Despite this rapid progress, it is fair to say that there is still no definitive solution: lower precision training can still induce small but persistent accuracy degradation, and existing methods disagree about which techniques are essential.

At the same time, some principles are starting to emerge. On the forward pass, multiple methods implicitly or explicitly optimize representation quality, for instance by substituting the NVIDIA '25 NVFP4 recipe's two-dimensional weight quantization with one-dimensional quantization followed by re-quantization in the backward pass to increase scale factor granularity, or by minimizing quantization mean-square error (MSE) for weights and activations (Castro et al., 2025; Cook et al., 2025). On the backward pass, a separate line of evidence suggests that bias in the gradient estimator can accumulate over long training horizons, making unbiased or carefully controlled quantization important for stable convergence (Chmiel et al., 2025; Chen et al., 2025; Panferov et al., 2026). The difficulty is that these principles are not always clearly compatible. For instance, the NVIDIA '25 NVFP4 recipe uses two-dimensional quantization of weights to maintain consistency between forward and backward quantized weight representations, while the Quartet line of work formulates an unbiased backward-pass (gradient) estimation scheme that is most naturally applied alongside a one-dimensional weight quantization scheme. To approximate weight consistency, one-dimensional quantization schemes are paired with re-quantization in the backward

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

pass, which quantizes and dequantizes the forward pass tensor before transposing and quantizing in the backward pass. (Castro et al., 2025; Panferov et al., 2026).

This creates a practical question: which ingredients are non-negotiable, and which ones are merely useful at a given scale? The key problem we aim to address here is that the "right" answer is hard to infer from short ablations. Many accuracy effects in low-precision training only become visible late in training, or only after increasing model depth and token budget. As a result, as we show, recipes can appear close early on, while diverging later, in ways that are expensive to diagnose.

In this paper, we aim to isolate a set of principles that matter for NVFP4 training by systematically comparing variants of existing recipes at scale. We evaluate dense 8B models and 30B-A3B mixture-of-experts (MoE) models from the Nemotron model family, trained up to 1T tokens. We organize our comparison around the major design axes that separate current methods: whether to use selective high-precision layers, how to select scales in the forward and backward pass, whether to quantize weights in one or two dimensions, how to address bias in the backward pass, and how these choices interact.

Contributions. Our main finding is that the current recipes capture complementary pieces of the problem, we propose a "Mosaic" approach that blends the best qualities of each. In more detail:

1. We perform the first detailed and unified ablation on NVFP4 training techniques, examining how they measurably improve both training stability and model quality relative to BF16 training.

2. We show that keeping selected layers in higher precision is *necessary* for full accuracy recovery in NVFP4 training, even with an optimized recipe. Omitting this can cause training instabilities at larger model sizes and medium-scale token horizons.

3. We find that selectively applying the MS-EDEN (Panferov et al., 2026) unbiased estimator to gradients and Four Over Six (Cook et al., 2025) scale selection to the remaining backward tensors outperforms applying either technique uniformly across all backward tensors.

4. Based on these observations, we develop a combined NVFP4 training recipe, which we call "Mosaic NVFP4," with selective high-precision layers, Random Hadamard Transforms on both backward GEMMs, unbiased backward-pass via Quartet II on gradient tensors, and Four Over Six scale selection on remaining backward tensors and forward tensors. To the best of our knowledge, this is the strongest FP4 training result demonstrated at this scale in loss gap to BF16. This scheme remains compatible with efficient kernels.

2. Related Work

Lower-precision training. Low-precision training has a long history, beginning with reduced-precision and integer training for smaller neural networks (Courbariaux et al., 2015; Banner et al., 2018; Yang et al., 2019). For Transformer models, 8-bit training has become increasingly practical through methods such as SwitchBack, JetFire, HALO, and FP8 training systems (Wan et al., 2023; Xi et al., 2023; Shen et al., 2024; Panferov et al., 2025; Micikevicius et al., 2022). These methods established several recurring themes: the need for fine-grained scaling, special treatment of outlier values, and careful handling of the backward pass.

Training in FP4 formats. The introduction of hardware-supported FP4 microscaling formats has renewed interest in end-to-end 4-bit training. Early FP4 and INT4 approaches showed that stable training is possible in constrained regimes, but often required higher-precision fallbacks or accelerated only part of the training computation (Xi et al., 2023; Tseng et al., 2025). More recent work has directly targeted native FP4 training. Chmiel et al. (2025) studied fully quantized FP4 training with NVFP4-style scaling and identified stochastic rounding as important for the backward and update passes. NVIDIA et al. (2025) introduced a large-scale NVFP4 recipe based on random Hadamard transforms, two-dimensional weight quantization, stochastic rounding, and selective high-precision layers. TetraJet-v2 (Chen et al., 2025) further proposed oscillation suppression and outlier control mechanisms. Four Over Six (Cook et al., 2025) improved NVFP4 quantization accuracy by adaptively choosing between two scale factors.

Quartet and unbiased gradient estimation. Quartet (Castro et al., 2025) studied MXFP4 training through the lens of "low-precision scaling laws," linking forward compression error to parameter efficiency and backward-pass bias to data efficiency. Quartet II (Panferov et al., 2026) extended this line to NVFP4 by introducing MS-EDEN, an unbiased microscaling quantizer with lower error than standard stochastic rounding. Our work is closest in spirit to these studies, but asks a different question: given several plausible recipes, which principles remain necessary when the methods are compared under a common implementation and training budget?

3. Experimental Setup

All NVFP4 variants are implemented in a common training stack and trained on the same GPU architecture.

Models and data. We evaluate both dense and sparse hy-

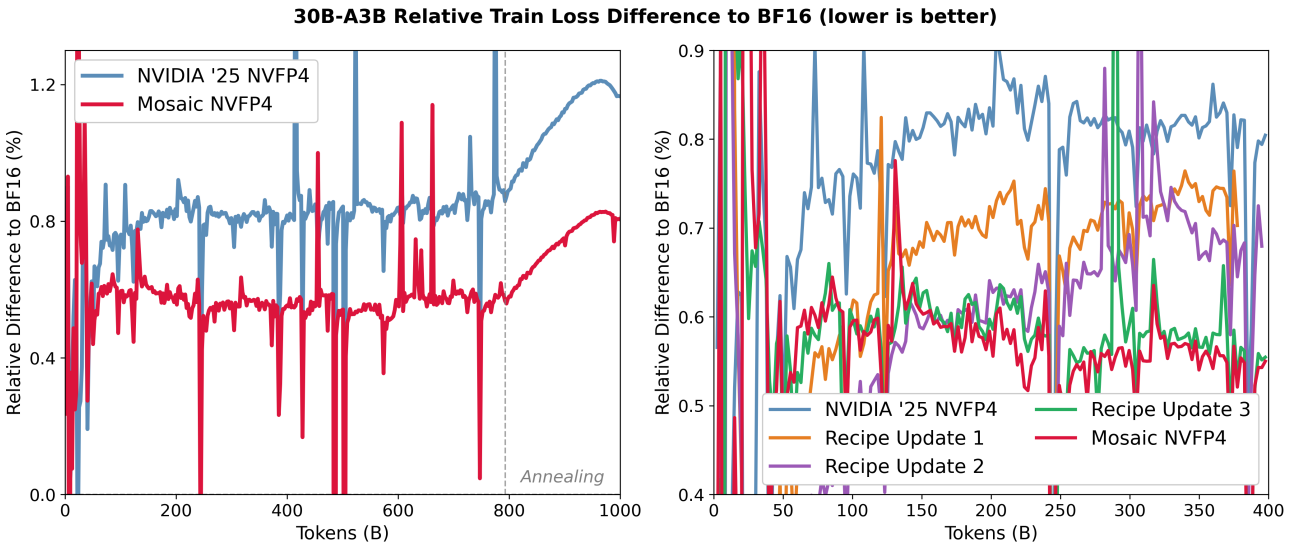

Figure 1. Left: Relative train loss difference between NVFP4 recipe and BF16 (lower is better), trained with NVIDIA '25 NVFP4 and Mosaic NVFP4 recipes to 1T tokens on Nemotron 3 Nano. Right: Cumulative recipe updates, each adding one Mosaic NVFP4 component, trained to 400B tokens on Nemotron 3 Nano. All recipes keep the last 15% of layers in BF16. Recipe updates described in Table 1.

brid Mamba-Transformer architectures. The dense setting uses the 8B variant of the NVIDIA Nemotron-H Family of Hybrid Mamba-Transformer Models (NVIDIA, 2025). This model has 24 Mamba-2 layers, 4 Attention layers, and 24 FFN layers. We train on the same data distribution as (NVIDIA, 2025), following their phased data-blending approach (Feng et al., 2024) with diverse data for the first 600B tokens and high-quality data for the remainder of the 1T token horizon. We additionally validated NVFP4 pretraining recipes on the Nemotron 3 Nano Hybrid Mamba-Transformer MoE architecture (30B-A3B) using the first 1T tokens of the 25T token dataset described here (NVIDIA, 2025). Like Nemotron-H, this model has a high ratio of Mamba-2 to attention layers, with 128 total routed experts, top-k of 6, and 2 shared experts per MoE FFN layer. We report training and validation loss alongside downstream evaluations spanning knowledge, reasoning, math, and coding benchmarks.

Quantization method terminology. Quantization is applied on the network's linear layers and grouped linear layers. To describe quantization recipes, we refer to three generalized-matrix-matrix product (GEMM) operations and the quantized operands to these products. Assuming GEMM contracts along the last dimension of the two inputs, we formalize them as follows:

1. **Y=GEMM(X, W)** — the product between the input X and the weight W.

2. **DGRAD=GEMM(G, W.T)** — the product between the gradient G w.r.t Y and the transposed weight W.T.

3. **WGRAD=GEMM(G.T, X.T)** — the product between the transposed gradient G and the transposed input X.T.

When both operands of a GEMM are quantized to NVFP4 along their last dimension, the multiplication can be performed using specialized TensorCores on Blackwell GPUs. As all operands across the three GEMMs are unique, we will use the notation above to discriminate between each tensor in a computation scheme.

Compared methods. We instantiate the main design choices from prior recipes in a common framework:

- **NVIDIA '25 NVFP4 :** two-dimensional quantization of weights, backward-pass stochastic rounding, Hadamard transforms on WGRAD, and selective high-precision layers (NVIDIA et al., 2025).
- **Quartet-II-style NVFP4:** unbiased backward-pass quantization via MS-EDEN (Panferov et al., 2026).
- **Four Over Six:** adaptive block scaling for certain operations (Cook et al., 2025).

Questions. Our experiments are organized around the following questions:

- Do these new recipe techniques provide incremental benefits to model quality at scale?
- Are selective high-precision layers necessary for long-horizon accuracy?

- Is unbiased backward-pass quantization still necessary once forward-pass representation is improved and selected layers are protected?
- Do trends with NVFP4 quantization reproduce across dense and MoE architectures?

4. Experimental Results

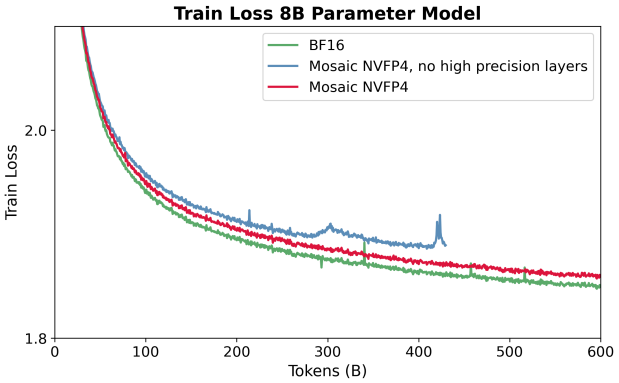

Figure 2. Stability study of the Mosaic NVFP4 recipe with last 15% of the network kept in BF16 vs. all layers quantized on the Nemotron-H 8B model.

4.1. Ablation Studies on Recipe Modifications

We ablate each recipe modification on Nemotron 3 Nano, starting from the NVIDIA et al. (2025) recipe and stacking techniques sequentially (Table 1). We refer to the final stacked recipe as Mosaic NVFP4 . The first update replaces two-dimensional weight quantization with one-dimensional quantization, re-quantizing in the backward pass to keep forward and backward weight representations consistent. The second adds a dimension-128 tiled Random Hadamard Transform to DGRAD and updates the WGRAD transform to use a consistent tile shape and a resampled sign vector throughout training. The third applies MS-EDEN to gradient tensors in place of stochastic rounding and adds Four Over Six scale selection to the remaining backward-pass tensors. The fourth adds Four Over Six to the forward-pass tensors. Each modification improves training loss (Figure 1), yielding a 0.36 pp total improvement once fully stacked. The largest single gain comes from replacing stochastic rounding with MS-EDEN and adding Four Over Six on the backward tensors, which we attribute to MS-EDEN's ability to achieve unbiasedness at lower quantization-error cost (Panferov et al., 2026).

4.2. Selective High-Precision Layers is Required

The need for high precision layers to maintain stability is evident at larger model scales and longer token horizons, even with an optimized recipe. Figure 2 shows that the Mosaic NVFP4 recipe diverges on the Nemotron-H 8B model past 300B tokens when all layers are quantized to NVFP4.

4.3. Backward-Pass Unbiasedness is Required

We studied backward pass quantization strategies on Nemotron 3 Nano. Recipes in this set of ablations informed Update 3 of Table 1. Each use 1D weight quantization with re-quantization in the backward pass, dimension-128 tiled Random Hadamard Transforms on both WGRAD and DGRAD. We vary the use of MS-EDEN and Four Over Six on backward tensors, applying: i) MS-EDEN on all backward tensors, ii) MS-EDEN on gradients only, iii) Four Over Six on all backward tensors, and iv) MS-EDEN on gradients and Four Over Six on the remaining backward tensors (X.T, W.T). These recipes are tested with and without the last 15% of the network kept in BF16. Shown in Figure 3, applying MS-EDEN to all tensors in backward GEMMs provides no clear benefit over applying MS-EDEN only to gradients. We found that the best configuration, observed when all layers are quantized, applies MS-EDEN to G and G.T and Four Over Six to X.T and W.T. We posit that this is because the bias correction is most critical for the gradient tensors.

4.4. The Best Stacked Recipe

Both the Mosaic NVFP4 and NVIDIA '25 NVFP4 recipes were trained on Nemotron 3 Nano (30B-A3B) (Figure 1) and Nemotron-H 8B (Figure 4) to 1T tokens, keeping the last 15% of the network in BF16 for the entirety of training. On the dense model, the Mosaic NVFP4 finishes training with less than a 1.2% relative train loss gap from BF16 and improves upon the previous loss gap by 0.24 percentage points. The benefit of Mosaic NVFP4 is most pronounced in the beginning of training and then during the annealing stage. Train and validation loss gap dynamics differ slightly due to dataset differences, but the gap between recipes is consistent across both. The switch from Phase 1 to Phase 2 data introduces a transiently higher loss gap for the Mosaic NVFP4 recipe, though sustained improvements are visible again after training has progressed in Phase 2. Both recipes show an increased loss gap from BF16 during the learning rate annealing phase. On the MoE model, the Mosaic NVFP4 recipe shows sustained improvement throughout the token horizon, starting after the first 75B tokens (Figure 1). The Mosaic NVFP4 recipe finishes training with less than a 0.81% relative train loss gap from BF16 and improves upon the previous loss gap by 0.36 percentage points. The training curriculum is restricted to a single phase of data so the only change in loss gap trajectory occurs during the learning rate annealing, which impacts both recipes equally (Figure 1).

Table 1. Recipe improvements over the prior state-of-the art at this model and token scale. All recipes keep last 15% of layers in BF16 unless otherwise noted.

Recipe	Weight Quant	Weight Re-quant	Tiled Hadamard Transforms	Hadamard Sign Vectors	Technique to Address Bias in Gradients	Four Over Six Tensors
NVIDIA '25 NVFP4	2D	No	Dim-16 WGRAD	Static	SR	None
Update 1	1D	Yes	Dim-16 WGRAD	Static	SR	None
Update 2	1D	Yes	Dim-128 WGRAD, DGRAD	Random	SR	None
Update 3	1D	Yes	Dim-128 WGRAD, DGRAD	Random	MS-EDEN	W.T, X.T
Mosaic NVFP4	1D	Yes	Dim-128 WGRAD, DGRAD	Random	MS-EDEN	W, X, W.T, X.T

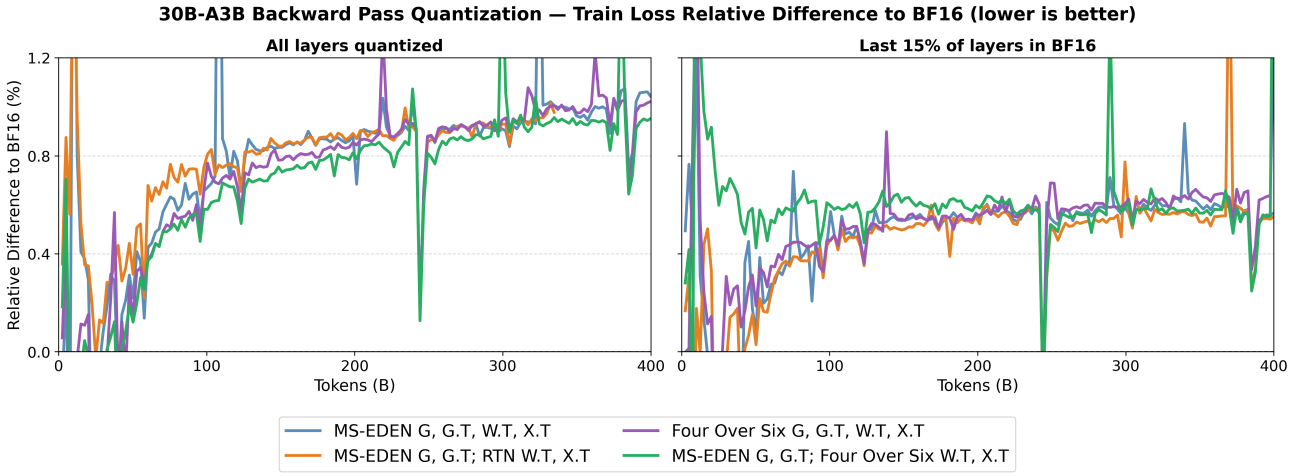

Figure 3. Different schemes for backward quantization shown on Nemotron 3 Nano shown with all layers are quantized in each of these recipes (left) and last 15% of layers in the network left in BF16 (right). Best configuration with all layers quantized uses MS-EDEN on gradient tensors and Four Over Six rounding on remaining backward tensors (X.T and W.T). No clear separation in backward techniques at this model size and token horizon when the last 15% of layers in the network are left in BF16.

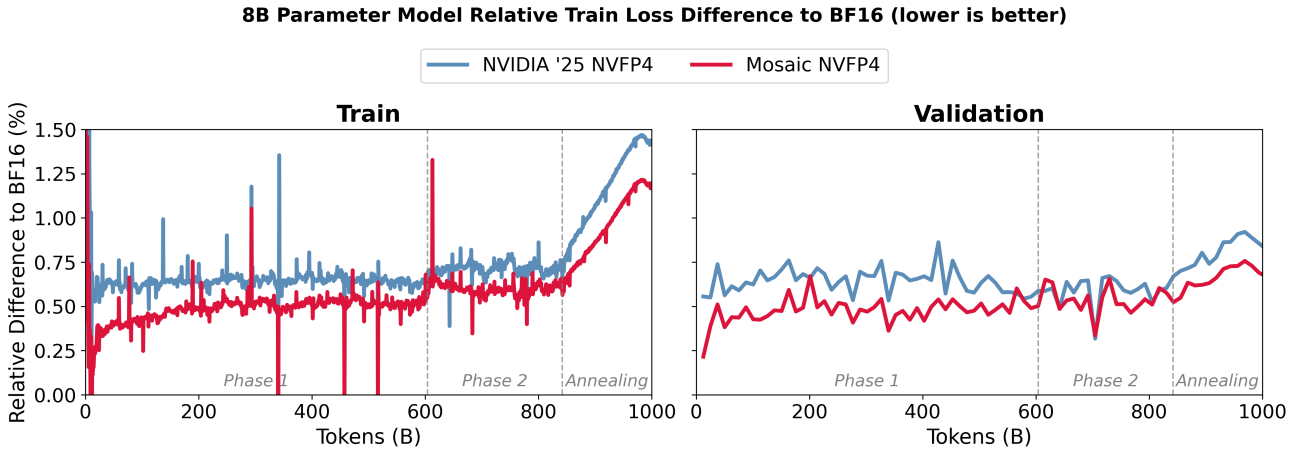

Figure 4. Percent relative train (left) and validation (right) loss difference compared to BF16 (lower is better) evaluated on the Nemotron-H 8B model to 1T tokens. Mosaic NVFP4 recipe has better training and validation loss throughout training than NVIDIA '25 NVFP4 .

5. Discussion

Prior work has shown that a close variant of the Mosaic NVFP4 recipe can converge on small models over short token horizons without keeping any layers in high precision (Panferov et al., 2026). However, at scale these NVFP4 recipes still require high-precision layers, suggesting a representation failure that the error-reduction techniques studied in this work cannot address. This is shown most directly on the 8B parameter model trained to 300B tokens in Figure 2, and further supported by Figure 3, where all recipes, regardless of the backward-tensor technique used, show a loss gap that grows throughout training.

Figure 1 shows that the relative ranking of recipes does not stabilize until well past 150B tokens of training. Notably, weaker recipes often initially appear better than the strongest ones. These early changes in ordering can be attributed to differences in the randomly sampled sign vectors used by the Random Hadamard Transforms, noise in training, or convergence properties of individual recipes. Only after enough tokens have been seen do the rankings settle. This view suggests that short NVFP4 pretraining ablations are useful for detecting catastrophic instability, but they are unfortunately not sufficient for deciding whether a new training algorithm will preserve accuracy at scale. For FP4 pretraining methods, where the intended benefit is to make long training runs cheaper, this late-emerging failure mode is especially important.

The separation between techniques applied to backward tensors is more visible in a less stable training regime. Figure 3 shows a slight separation when all layers are quantized, but indistinguishable curves once the last 15% of the network is kept in BF16. The winning recipe addresses bias in the gradients and reduces quantization error in the subsequent backward tensors, suggesting that both interventions matter, each on the specific tensors they target. We hypothesize that this insight informs better recipe choices for large-scale training.

The Mosaic NVFP4 recipe is an incremental improvement over the NVIDIA '25 NVFP4 recipe, robust across the two model sizes, datasets, and architectures explored in this work, and partially closing the train loss gap to BF16 across two scales.

6. Limitations

This paper focuses on medium-scale dense and MoE language models and on NVFP4 training. We do not yet establish whether the same conclusions hold for even larger production runs, other modalities, post-training, or other FP4 formats. We leave this as future work.

References

Amodei, D. and Hernandez, D. Ai and compute. https://openai.com/research/ai-and-compute, 2018.

Banner, R., Hubara, I., Hoffer, E., and Soudry, D. Scalable methods for 8-bit training of neural networks. In *Advances in Neural Information Processing Systems*, 2018.

Castro, R. L., Panferov, A., Tabesh, S., Sieberling, O., Chen, J., Nikdan, M., Ashkboos, S., and Alistarh, D. Quartet: Native FP4 training can be optimal for large language models. *arXiv preprint*, 2025.

Chen, Y., Xu, X., Zhang, P., Beyer, M., Rapp, M., Zhu, J., and Chen, J. Tetrajet-v2: Accurate NVFP4 training for large language models with oscillation suppression and outlier control. *arXiv preprint arXiv:2510.27527*, 2025.

Chmiel, B., Fishman, M., Banner, R., and Soudry, D. FP4 all the way: Fully quantized training of LLMs. *arXiv preprint arXiv:2505.19115*, 2025.

Cook, J., Guo, J., Xiao, G., Lin, Y., and Han, S. Four over six: More accurate NVFP4 quantization with adaptive block scaling. *arXiv preprint arXiv:2512.02010*, 2025.

Courbariaux, M., Bengio, Y., and David, J.-P. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems*, 2015.

Feng, S., Prabhumoye, S., Kong, K., Su, D., Patwary, M., Shoeybi, M., and Catanzaro, B. Maximize your data's potential: Enhancing llm accuracy with two-phase pretraining, 2024. URL https://arxiv.org/abs/2412.15285.

Micikevicius, P., Stosic, D., Burgess, N., Cornea, M., Dubey, P., Grisenthwaite, R., Ha, S., Heinecke, A., Judd, P., Kamalu, J., et al. FP8 formats for deep learning. *arXiv preprint arXiv:2209.05433*, 2022.

NVIDIA. Nemotron-h: A family of accurate and efficient hybrid mamba-transformer models. 2025. URL https://arxiv.org/abs/2504.03624.

NVIDIA. NVFP4 trains with precision of 16-bit and speed and efficiency of 4-bit. NVIDIA Technical Blog, 2025.

NVIDIA. Nvidia nemotron 3: Efficient and open intelligence, 2025. URL https://arxiv.org/abs/2512.20856.

NVIDIA, Abecassis, F., Agrusa, A., Ahn, D., Alben, J., et al. Pretraining large language models with NVFP4. *arXiv preprint arXiv:2509.25149*, 2025.

Panferov, A., Schultheis, E., Tabesh, S., and Alistarh, D. Quartet II: Accurate LLM pre-training in NVFP4 by improved unbiased gradient estimation. *arXiv preprint arXiv:2601.22813*, 2026.

Panferov, A. et al. HALO: Hadamard-assisted low-precision optimization for llm training. *arXiv preprint*, 2025.

Rouhani, B. D., Zhao, R., More, A., Hall, M., Khodamoradi, A., Deng, S., Choudhary, D., Cornea, M., Dellinger, E., Denolf, K., Dusan, S., Elango, V., Golub, M., Heinecke, A., James-Roxby, P., Jani, D., Kolhe, G., Langhammer, M., Li, A., Melnick, L., Mesmakhosroshahi, M., Rodriguez, A., Schulte, M., Shafipour, R., Shao, L., Siu, M., Dubey, P., Micikevicius, P., Naumov, M., Verrilli, C., Wittig, R., Burger, D., and Chung, E. Microscaling data formats for deep learning, 2023. URL https://arxiv.org/abs/2310.10537.

Sevilla, J., Heim, L., Ho, A., Besiroglu, T., Hobbhahn, M., and Villalobos, P. Compute trends across three eras of machine learning. *International Joint Conference on Neural Networks*, 2022.

Shen, H. et al. Jetfire: Efficient and accurate transformer pretraining with INT8 data flow and per-block quantization. *arXiv preprint arXiv:2403.12422*, 2024.

Tseng, A. et al. Training with MXFP4 for large language models. *arXiv preprint*, 2025.

Wan, Y. et al. Switchback: Accelerating deep learning training via quantized activation gradients. *arXiv preprint arXiv:2304.13013*, 2023.

Xi, H., Chen, Y., Zhao, K., Zheng, K., Chen, J., and Zhu, J. INT4 training for neural networks. *arXiv preprint arXiv:2306.11987*, 2023.

Yang, G.-Y. et al. Training deep neural networks with 8-bit floating point numbers. In *Advances in Neural Information Processing Systems*, 2019.

A. Appendix

A.1. Model and Training Parameters

Model sizes follow definitions in Tables 2 and 3. All models were trained using AdamW with the Warmup Stable Decay learning rate schedule, and annealed during the last 20% of the token horizon for Nemotron 3 Nano and the last 15% of the token horizon for Nemotron-H 8B. Each model was trained to a maximum of 1T tokens, some ablations were stopped early in the schedule. Weight decay of 0.1 was used for all models. Attention layers used Grouped Query Attention. All models have a sequence length of 8192.

Table 2. Model parameters for each experimental setup. †FFN layers are MoE for Nemotron 3 Nano.

Model	No. Mamba-2 Layers	No. Attention Layers	No. FFN† Layers	No. KV Heads	Total Parameters
Nemotron 3 Nano	23	6	23	2	3B (30B total)
Nemotron-H	24	4	24	8	8B

Table 3. Training parameters for each experimental setup.

Model	Batch Size	Min LR	LR
Nemotron 3 Nano	3072	1e-5	1e-3
Nemotron-H	768	8e-6	8e-4

A.2. Downstream Task Accuracy

The results in Table 4 show that Mosaic NVFP4 recipe shows improvements in downstream task accuracy in some cases, for example a 0.9 pt MMLU improvement on Nemotron 3 Nano. On the Nemotron-H 8B model, improvements in loss trajectory do not always translate to downstream task accuracy.

Task	Nemotron 3 Nano		Nemotron-H 8B	
	NVIDIA '25 NVFP4	**Mosaic NVFP4**	**NVIDIA '25 NVFP4**	**Mosaic NVFP4**
General Knowledge				
MMLU, 5-shot	67.5	**68.4**	61.0	60.3
MMLU-Pro, 5-shot CoT	45.0	**47.1**	29.7	27.6
Math Average	79.4	79.2	51.3	**52.3**
(Math 500, 4-shot; GSM8K, 8-shot)				
Common Sense Average	77.6	77.1	76.5	75.9
(ARC Challenge, 25-shot; WinoGrande, 0-shot; Hellaswag, 0-shot)				
Code Average	57.1	56.6	43.4	43.0
(HumanEval, Avg Pass@1; MBPP Sanitized, 3-shot)				
Reading Comprehension				
RACE, 0-shot	80.8	**81.1**	77.7	**78.8**

Table 4. Downstream task accuracy on both Nemotron 3 Nano and Nemotron-H 8B. Models are evaluated in BF16.