Outlier-Safe Pre-Training for Robust 4-Bit Quantization of Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have experienced remarkable performance gains through increased parameter counts and training data, but this growth poses significant challenges for on-device deployment. Quantization has emerged as a critical technique to reduce compute and memory overhead in resource-constrained environments. Unfortunately, traditional quantization approaches are hampered by outliers–rare but extreme activation values that stretch quantization ranges and degrade performance. Recent work suggests that the Adam optimizer itself may contribute to outlier formation through its element-wise gradient normalization. In this paper, we introduce Muon as a practical alternative to Adam for large-scale LLM training. By employing efficient gradient orthogonalization via Newton-Schulz iterations, Muon avoids the heavy overhead common in second-order methods like Shampoo. We further propose an **Outlier-Safe Pre-Training (OSP)** framework that incorporates learnable embedding rotations and single-scale RMSNorm, suppressing outliers without architectural modifications at inference. Our ablation study on a 100 billion token corpus demonstrates that these components effectively mitigate outliers while maintaining model quality. We validate our approach by training a 1.4B-parameter LLM on 1 trillion tokens–to our knowledge, the first production-scale model trained without Adam. The resulting model exhibits distinct quantization behavior under 4-bit weight and activation (W4A4) quantization compared to existing open-source LLMs, suggesting new possibilities for robust low-bit pre-training in LLM development.

1 Introduction

Large Language Models (LLMs) have grown at an unprecedented pace, resulting in escalating compute, memory, and energy costs. Such growth poses significant challenges for on-device LLM deployment, especially on devices with limited resources.

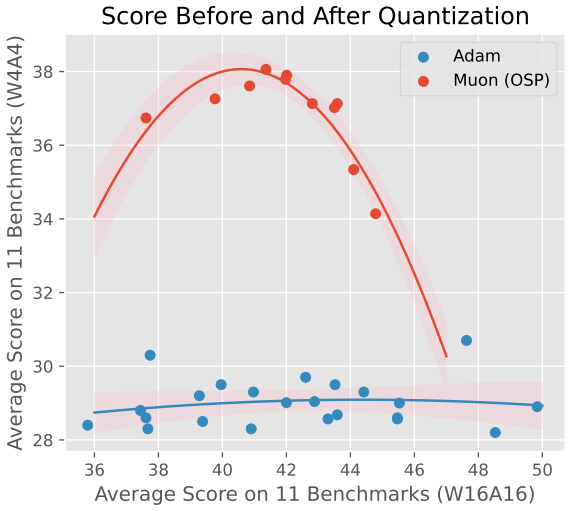


Figure 1: Comparison of Quantization-induced Degradation (QiD) patterns under 4-bit weight and activation (W4A4) quantization. In the plot, **Adam** refers to various open-source LLMs trained with the Adam optimizer, which exhibit one characteristic pattern of performance under quantization. **Muon (OSP)** represents checkpoints from our model trained with the Muon optimizer over **OSP** framework, revealing a distinctly different QiD pattern. Performance is evaluated across standard LLM benchmarks detailed in Section 4.1.

As a practical solution, various quantization methods (Frantar et al., 2023; Shao et al., 2024; Ashkboos et al., 2024b; Liu et al., 2024b) have been developed to reduce model size and computational overheads by representing weights and activations in lower-precision formats.

However, activation outliers in LLMs (Bondarenko et al., 2021; Wei et al., 2022; Liu et al., 2024a; Dettmers et al., 2022; Xiao et al., 2023; Ashkboos et al., 2024b; Liu et al., 2024b) present a persistent obstacle: these rare but extreme values expand the dynamic range, rendering simple methods like Round-To-Nearest ineffective. To accommodate these outliers, the quantization range must be widened, which results in a loss of pre-

cision for the majority of non-outlier values. To address this, prior work has explored strategies such as Quantization-Aware Training (QAT) (Liu et al., 2024a; Chen et al., 2024; Nrusimha et al., 2024), Post-Training Quantization (PTQ) that handles outliers in a mixed-precision manner (Zhao et al., 2024; Dettmers et al., 2022), shifting outliers between weights and activations (Xiao et al., 2023; Lin et al., 2024), or applying a random rotation matrix to mitigate outliers through computational invariance (Chee et al., 2023; Tseng et al., 2024; Ashkboos et al., 2024b; Liu et al., 2024b). Despite these efforts, existing methods often require additional training or calibration and have yet to fully explain why outliers arise.

Recent findings (Elhage et al., 2023; He et al., 2024; Caples and rrenaud, 2024) suspect that the Adam optimizer itself may be responsible for the occurrence of activation outliers. Adam's element-wise gradient normalization privileges the coordinate-wise basis of model parameters over arbitrary directions in parameter space (Elhage et al., 2023), leading to basis-aligned outliers. When models are trained with pure SGD, SOAP (Vyas et al., 2024), or other non-Adam optimizers, outliers appear less frequently.

In this work, we introduce Muon (Jordan et al., 2024; Bernstein and Newhouse, 2024a) as a practical, outlier-suppressing alternative to Adam for large-scale LLM training. Although Muon draws inspiration from second-order methods such as Shampoo (Gupta et al., 2018; Anil et al., 2020) and SOAP (Vyas et al., 2024), it circumvents the high overhead associated with full matrix preconditioning (as Table 2). By orthogonalizing gradients with a momentum buffer, essentially a simplified Shampoo (Bernstein and Newhouse, 2024b,a; Morwani et al., 2024), Muon achieves efficiency far superior to existing second-order methods. While Shampoo and SOAP often suffer from slow performance and large memory footprints, Muon matches the convergence speed of these second-order optimization methods while requiring less memory than Adam (Jordan et al., 2024). Our contributions can be summarized as follows:

- **Outlier-Safe Pre-Training (OSP).** We propose training LLMs without introducing outliers from the start by replacing Adam with Muon. We observe that Muon alone does not entirely eliminate outliers, contradicting some prior suggestions (He et al., 2024). Hence, we propose an **OSP** framework, adding an embedding rotation matrix and a single-scale RMSNorm during training. These modifications remove element-wise scaling while retaining architectural equivalence at inference (Ashkboos et al., 2024a,b; Liu et al., 2024b).

- **Scaling Up to 1 Trillion Tokens.** While most open-source LLMs still rely on Adam, we demonstrate that Muon maintains stable training and our framework can reduce outliers in a 1.4 billion parameter model trained on 1 trillion tokens. To our knowledge, this is the first production-level LLM of this scale trained without Adam. We open-source the trained model to facilitate further study of how different optimizers affect LLM behavior.

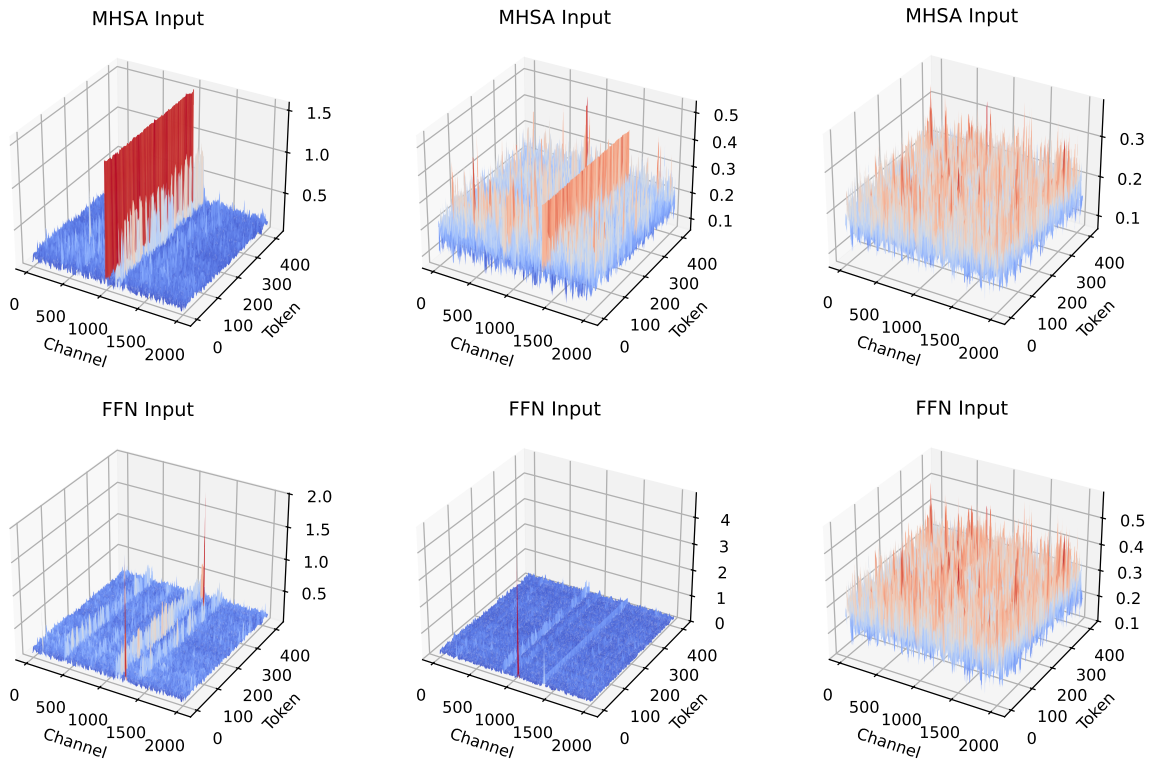
- **State-of-the-Art Low-Bit Robustness.** Combining Muon training with post-training quantization (PTQ) achieves the best 4-bit weight and activation over Round-To-Nearest (RTN) quantization performance among open-source LLMs, substantially improving *robustness to low-bit quantization*.

2 Preliminary

Several prior works have investigated activation outliers in LLMs. For instance, Nrusimha et al. (2024) explored how outliers emerge during pre-training and proposed a combination of Quantization-Aware Training (QAT) and kurtosis regularization to suppress them. While this approach reduces extreme activation values, it still does not eliminate them entirely and incurs a significant training slowdown (over 10%) due to QAT's overhead. Likewise, He et al. (2024) proposed an Outlier-Protected (OP) Transformer block, but this architectural modification complicates existing inference pipelines and requires entropy regularization to prevent issues like entropy collapse. These approaches highlight two main challenges: (1) modifications to the training pipeline (e.g., slow QAT or custom CUDA kernels) and (2) architectural changes that complicate deployment.

3 Method

Meanwhile, He et al. (2024) also explored to replace Adam optimizer with SOAP variants, suppressing outlier emergence while pre-training LLMs. This aligns with findings that Adam's



(a) Adam **(b) Muon** **(c) Muon (OSP)**

Figure 2: Activation histograms from the 20th layer input of Multi-Head Self-Attention (MHSA) and Feed-Forward Network (FFN) in 1.4B-parameter models trained on 100B tokens. Three optimization approaches are compared: (a) standard Adam optimizer, (b) Muon optimizer without modifications, and (c) Muon with learnable embedding rotations and single-scale RMSNorm (SSNorm). The histograms demonstrate that replacing Adam with Muon alone is insufficient to fully mitigate activation outliers.

element-wise gradient normalization may contribute to outlier emergence (Elhage et al., 2023; Caples and rrenaud, 2024).

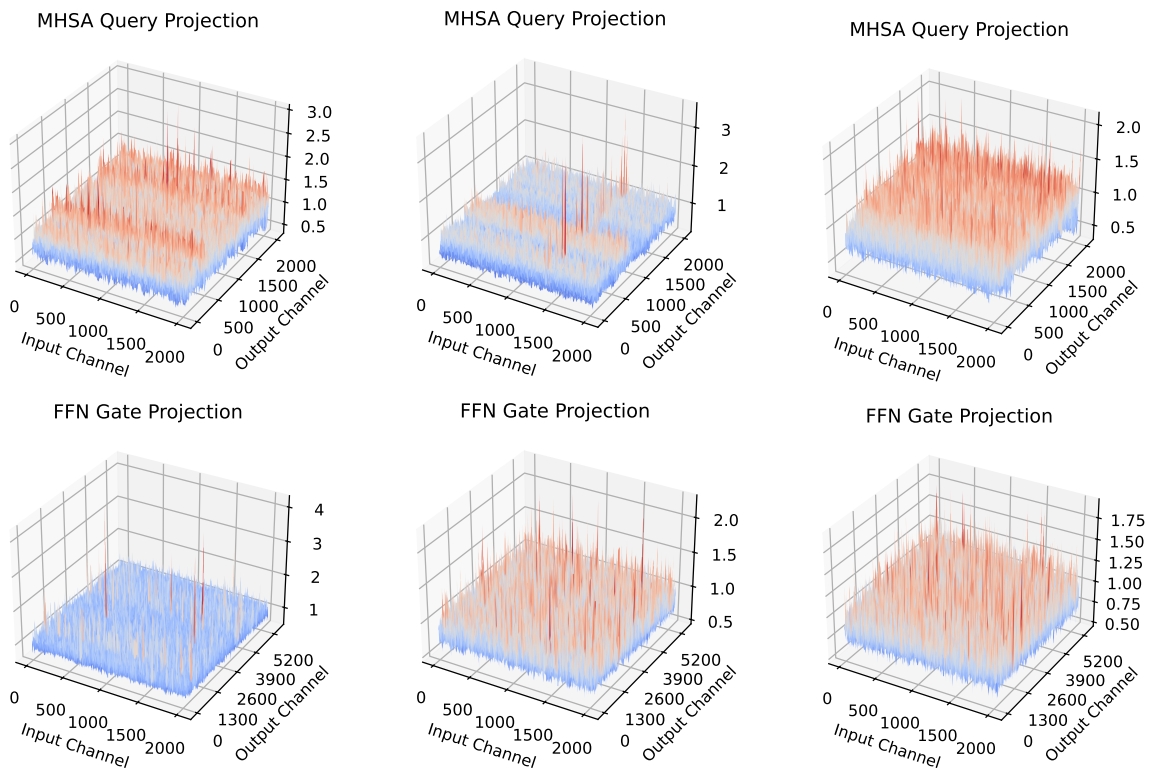
Building on these insights, We introduce a training pipeline for Large Language Models (LLMs) that inherently avoids the activation outliers impeding low-bit quantization. In contrast to approaches that require architectural modifications or additional training overhead like Quantization-Aware Training, our method employs (1) a simpler yet effective second-order-inspired optimizer called Muon, (2) learnable embedding rotations, and (3) a single-scale RMSNorm. Below, we detail the motivation behind these design choices and describe how each component helps mitigate outliers without incurring substantial computational or implementation costs.

3.1 Learnable Embedding Rotations

Contrary to He et al. (2024), we observe that non-diagonal preconditioning alone fails to remove all outliers, often leaving residual extremes in the embedding and unembedding layers. This is partially because large embedding matrices (e.g., vocab size $V\times$ hidden dimension m) are often handled differently by standard second-order routines. For instance, Shampoo or SOAP might skip the vocabulary dimension for the preconditioner factorization, and Muon may revert to Adam for embeddings.

Drawing on the concept of *computational invariance* (Section 5.3), we insert learnable rotation matrices around the embedding layers. Let $E = [e_1, e_2, \ldots, e_V]^T \in \mathbb{R}^{V\times m}$ be the embedding matrix and $Q \in \mathbb{R}^{m\times m}$ be a learnable matrix initialized to be orthogonal. Each embedding vector e_i is then mapped to $\hat{e}_i = e_i Q$, effectively rotating the embedding space. A similar rotation is applied before the unembedding layer.

Since those rotations are simply linear transformations, they can be merged into single effective matrices for inference, i.e. $\hat{V} = VQ$, preserving the original architecture and no additional overhead. Empirically, we observed that training additional matrices reduces outliers within the embedding



(a) Adam **(b) Muon** **(c) Muon (OSP)**

Figure 3: Weight histograms from the 20th layer input of Multi-Head Self-Attention (MHSA) and Feed-Forward Network (FFN) in 1.4B-parameter models trained on 100B tokens. Three optimization approaches are compared: (a) standard Adam optimizer, (b) Muon optimizer without modifications, and (c) Muon with learnable embedding rotations and single-scale RMSNorm (SSNorm). The histograms reveal that weight matrices, like activations, exhibit outliers that complicate quantization.

space and the resulting model size increases by only about 0.6%, making it a practical solution for mitigating outliers in large-scale LLMs.

3.2 Single-Scale RMSNorm

Normalization layers also play a critical role in outlier formation (Wei et al., 2022; He et al., 2024; Nrusimha et al., 2024). He et al. (2024) employ Simple RMSNorm (SRMSNorm) (Qin et al., 2023), which rescale the output vector by $\sqrt{m}$ without learnable parameters, where m is a dimensionality of input activation vector.

Inspired by SRMSNorm, we propose a single-scale RMSNorm (SSNorm), which use a single learnable scalar to control the scale of activation norm. SSNorm uses only a single scalar $\gamma \in \mathbb{R}$ to scale the normalized activations:

$$\text{SSNorm}(x) = \gamma \frac{x}{||x||_2}. \quad (1)$$

This design removes the channel-wise multiplicative degrees of freedom that can inflate specific coordinates and cause outliers. By controlling all dimensions uniformly with one parameter, SSNorm helps stabilize activations and reduces extreme values across the dimensions.

4 Experiments

4.1 Experimental Setup

We train 1.4B-parameter LLAMA (Touvron et al., 2023) models to evaluate quantization performance under different optimizers and architectural configurations. Training data is drawn from a mixture of FineWeb-Edu (Lozhkov et al., 2024), FineMath (Allal et al., 2025), Cosmopedia (Ben Allal et al., 2024), and Python code samples from StarCoder's training set (Li et al., 2023).

Training is conducted on TPU v4-512 Pod Slice, utilizing the Adam optimizer with a learning rate of 5×10^{-3} and Muon optimizer with a learning rate of 5×10^{-4}. We use a batch size of 4M tokens with sequence length of 2048 tokens. We use weight decay of 0.01 and trapezoidal learning rate scheduling (Hägele et al., 2024; Wen et al., 2024), of which learning rates are increased from 0 to maximum

Optimizer	EmbRot	SSNorm	Ex. Kurt.	Had.	16-16-16		4-8-16		4-8-8		4-4-16		4-4-4	
					Avg.	PPL	Avg.	PPL	Avg.	PPL	Avg.	PPL	Avg.	PPL
Adam	✗	✗	1818.56	✗	40.9	11.4	38.4	21.6	38.3	21.6	28.3	1e5	28.3	8e4
				✓	40.9	11.4	39.9	22.3	40.0	22.3	29.2	3e4	28.9	3e4
Muon† (w/o Adam)	✗	✗	361.35	✗	40.9	11.7	38.4	14.8	38.3	14.8	28.3	1e6	28.3	8e5
				✓	40.9	11.7	37.4	15.4	37.4	15.4	33.9	24.5	33.7	24.8
Muon	✗	✗	1575.12	✗	41.3	11.4	39.8	13.8	39.8	13.8	30.8	934.3	30.5	1e4
				✓	41.3	11.4	40.4	12.9	40.5	12.9	38.4	15.7	38.3	15.8
Muon	✓	✗	703.23	✗	39.9	12.3	38.4	14.8	38.4	14.8	32.2	99.7	31.9	114.6
				✓	39.9	12.3	39.2	13.9	39.3	13.9	36.6	22.1	36.7	22.3
Muon	✗	✓	66.69	✗	**41.4**	**11.2**	**40.8**	12.4	**40.8**	12.4	36.7	43.3	36.6	44.2
				✓	**41.4**	**11.2**	**40.6**	12.2	**40.6**	12.2	38.5	33.7	38.3	34.1
Muon	✓	✓	**0.04**	✗	41.2	**11.2**	40.4	**12.2**	40.5	**12.2**	**38.1**	**19.4**	**37.8**	**19.6**
				✓	41.2	**11.2**	40.4	**12.1**	40.4	**12.1**	**39.1**	**13.4**	**39.0**	**13.5**

Table 1: Ablation study of models trained on 100B tokens. **EmbRot** denotes models with learnable embedding rotations, and **SSNorm** indicates single-scale RMSNorm. **Ex. Kurt** represents excess kurtosis (Section 5.1), while **Had.** indicates online Hadamard transformation from QuaRot Stage 1. Bit-width configurations (e.g., 16-16-16, 4-8-16) specify quantization precision for weights, activations, and key-value cache respectively. **Avg.** shows mean performance across 11 LLM benchmarks, and **PPL** reports WikiText-2 perplexity. †Model trained with Muon without Adam on embedding layers, applying Newton-Schulz orthogonalization to embedding gradients.

value for 5 billion tokens, and decayed to zero at the last 20% of the training steps.

To achieve faster training throughput, we adopt Fully-Sharded Data-Parallel (Xu et al., 2020) with parameters sharded across 16 accelerator cores. We further implement a distributed variant of Muon, distributing the Newton–Schulz iterations across 8 *optimizer-parallel* ranks for orthogonalization. As shown in Table 2, our distributed Muon incurs only about 2% overhead relative to Adam, whereas Distributed Shampoo (Anil et al., 2020) degrades performance by over 30% even with an relaxed update frequency (every 32 steps).

We evaluate model performance using perplexity on WikiText-2 (Merity et al., 2016) and accuracy on 11 standard LLM benchmarks: ARC (Clark et al., 2018), CommonsenseQA (Talmor et al., 2019), GSM8k (Cobbe et al., 2021) (8-shot), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), TriviaQA (Joshi et al., 2017) (5-shot), TruthfulQA (Lin et al., 2022), and WinoGrande (Sakaguchi et al., 2020).

4.2 Ablation Study

To analyze each component of our proposed method, we conduct an ablation study on a 100B-token subset of our training corpus. By default, we optimize embedding matrices using Adam for two reasons: orthogonalizing gradients for large vocabularies is computationally expensive, and Adam achieves better final performance than Muon alone (Jordan et al., 2024). This aligns with other second-order optimizers, which typically exclude the vocabulary dimension from their preconditioners due to the computational constraints of handling such large matrices. While we use Adam for embeddings by default, we also evaluate full Muon optimization as part of our ablation study.

Optimizer	TPS	Relative Speed
Adam	4.07M	–
Muon	3.99M	97.9%
Shampoo†	3.07M	75.5%

Table 2: Performance comparison of different optimizers on TPU-v4 512 Pod Slice. **TPS** denotes tokens processed per second during training, and **Relative Speed** indicates training speed relative to Adam. Muon demonstrates significantly better throughput than Distributed Shampoo. †Distributed Shampoo results shown with update frequency of 32 steps.

Quantization robustness is first evaluated through a simple Round-To-Nearest (RTN) approach. We also test QuaRot (Ashkboos et al., 2024b) Stage 1 by applying an online Hadamard transform to rotate hidden states within the FFN layers. Each experimental setting involves four distinct quantization scenarios summarized in Table 1.

As shown in Table 1, using *both* learnable embedding rotations and single-scale RMSNorm (SS-

Model	Params.	Tokens	ARC	CSQA	GSM8K	HS	MMLU	OBQA	PIQA	SIQA	TQA	TFQA	WG	Avg.
Pythia	1.4B	0.3T	27.2	21.5	0.0	25.8	26.2	24.8	53.2	37.2	0.0	50.1	49.0	28.6
TinyLlama	1.1B	2T	28.3	22.9	0.0	26.6	26.2	21.2	48.7	**40.7**	0.0	49.1	49.0	28.4
OPT	1.3B	0.3T	25.0	21.6	0.0	26.5	25.6	28.2	49.6	36.9	0.0	48.0	49.5	28.3
OLMo	1.2B	3T	27.7	25.8	0.0	27.0	26.1	25.8	54.1	37.4	0.0	49.0	**51.9**	29.5
MobileLLaMA	1.4B	1.3T	27.4	23.5	0.0	26.7	26.0	22.4	49.6	38.3	0.0	49.4	49.6	28.5
Qwen 1.5	1.8B	2.4T	27.2	25.4	0.0	28.3	25.7	25.0	54.1	39.1	0.0	50.1	49.3	29.5
Qwen 2	1.5B	7T	30.9	27.7	0.4	35.7	26.2	28.4	56.5	38.3	0.8	44.0	48.6	30.7
Qwen 2.5	1.5B	–	27.7	25.0	0.0	26.9	25.7	24.0	52.2	38.4	0.0	49.9	47.5	28.9
LLaMA 3.2	1.2B	–	29.3	24.7	**0.5**	30.1	25.8	27.4	53.3	39.5	0.1	45.3	50.3	29.7
Stable LM 2	1.6B	2T	26.2	24.0	0.0	27.0	24.6	27.0	51.1	37.8	0.0	**50.2**	51.1	29.0
SmolLM	1.7B	1T	28.4	25.7	0.0	27.0	26.1	28.0	51.0	38.8	0.0	48.9	48.4	29.3
SmolLM 2	1.7B	11T	25.8	22.4	0.0	25.9	24.2	26.6	51.5	36.0	0.0	48.0	50.0	28.2
Trained from scratch														
Adam	1.4B	1T	25.7	25.3	0.0	26.8	25.4	26.0	49.0	37.2	0.0	49.1	49.9	28.6
Muon (OSP)	1.4B	1T	**42.1**	**32.4**	0.0	**41.5**	**28.5**	**32.0**	**61.2**	38.8	**2.8**	45.5	50.8	**34.1**

Table 3: W4A4 quantization performance comparison across 12 open-source LLMs and our implementations. **Params.** indicates model size in parameters, and **Tokens** shows training dataset size. Benchmarks include CSQA (CommonsenseQA), HS (HellaSwag), OBQA (OpenBookQA), TQA (TriviaQA), TFQA (TruthfulQA), and WG (WinoGrande). Models trained with **Outlier-Safe Pre-Training** exhibit distinct quantization behavior across the benchmark suite.

Norm) yields minimal excess kurtosis (Section 5.1) and preserves performance under both RTN and Hadamard rotation. Histograms of activations and weights in Figures 2 and 3 further confirm that outliers are substantially mitigated.

Notably, simply switching from Adam to a non-diagonal optimizer for all parameters is insufficient to remove outliers; skipping the vocabulary dimension or partially applying Muon can still leave outlier issues. Moreover, applying Newton–Schulz updates to the embedding matrices raises training costs by about 2.2% overall, yet still underperforms our final approach on the 11 LLM benchmarks and WikiText-2 perplexity.

Figure 4 visualizes perplexities on WikiText-2 for weight and activation quantization ranging from 8 bits down to 4 bits (via RTN). Our method consistently preserves performance more effectively across all bit configurations.

4.3 Scaling Up to 1 Trillion Tokens

We scale our training to 1 trillion tokens, a dataset size commonly used for commercial LLMs but rarely explored in academic research on outlier mitigation (He et al., 2024; Nrusimha et al., 2024). While previous works limited their experiments to hundreds of billions of tokens, we demonstrate that our **OSP** framework maintains its effectiveness at trillion-token scale.

Table 3 compares W4A4 quantization performance for 12 open-source small LMs, each trained on a large corpus. Most baseline models suffer heavy accuracy drops under W4A4, and scores on

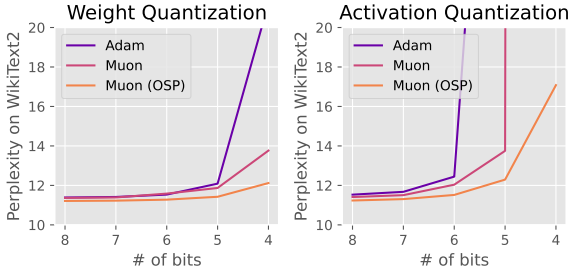


Figure 4: WikiText-2 perplexity under varying weight and activation quantization bit-widths for models trained on 100B tokens. Three configurations are compared: standard Adam, Muon, and Muon with **Outlier-Safe Pre-Training (OSP)**. The results demonstrate different quantization robustness patterns, particularly in low-bit scenarios.

multiple-choice benchmarks like ARC and CommonsenseQA degrade to near-random baselines (around 25%). By contrast, our approach exhibits notably stronger retention of performance, suggesting that both Muon and our architectural modifications better maintain quantization resilience.

4.4 Analysis of Quantization Robustness

Finally, we investigate how models trained with our method respond to standard Post-Training Quantization (PTQ) techniques. Since our approach focuses on pre-training, it can be combined with any PTQ method. As shown in Table 4, Adam-trained models suffer significant degradation under 4-bit weight and activation quantization when using minimal PTQ methods like QuaRot and GPTQ, which perform limited calibration. In contrast, models

Quantization	Adam	Muon (OSP)
RTN	14475.51	162.81
+ QuaRot†	4794.00	88.40
+ GPTQ	3723.46	18.72
+ SpinQuant	14.94	17.44

Table 4: WikiText-2 perplexity after applying various Post-Training Quantization (PTQ) methods to models trained with Adam versus **Outlier-Safe Pre-Training (OSP)**. The minimal PTQ methods (QuaRot and GPTQ) show limited effectiveness in mitigating quantization error for Adam-trained models. Models trained with **OSP** maintain consistent performance when combined with SpinQuant. †Only applies online Hadamard transform to Feed-Forward Network (FFN) hidden states.

trained with Muon and OSP show remarkable robustness to quantization, particularly on WikiText-2 where perplexity remains nearly unchanged after applying the more advanced SpinQuant method. This demonstrates that SpinQuant's learnable rotations serve a similar function to non-diagonal preconditioner optimizers in preserving model quality under quantization, which aligns to the previous observation (He et al., 2024).

Recent work (Ouyang et al., 2024) indicates that more extensively trained models are prone to severe Quantization-induced Degradation (QiD) in the low-bit setting. In Table 3, for instance, Qwen 2.5 and SmolLM 2 are trained on significantly more data than earlier versions, thereby achieving higher benchmark scores but suffering worse drops under W4A4 quantization.

To analyze quantization error patterns, we collect 11 checkpoints from our OSP model during its training and compare their performance before and after W4A4 RTN quantization against various Adam-trained open-source LLMs. As shown in Figure 1, models trained with our OSP approach follow a distinctly different Quality-in-Distribution (QiD) pattern than Adam-trained models, achieving higher absolute maximum performance. This discovery of a different quantization behavior pattern highlights the importance of exploring and analyzing non-Adam-trained LLMs for robust low-bit compression.

5 Related Works

5.1 Quantization

Quantization reduces the precision of weights and activations by mapping continuous floating-point values to discrete integers. While standard floating-point formats typically require 32 or 16 bits, quantization can reduce each value to 8 bits or even fewer than 4 bits. Let N be the target number of bits. A common baseline, Round-To-Nearest (RTN), is expressed as:

$$\hat{X} = \alpha \left\lfloor \frac{X - \beta}{\alpha} \right\rceil + \beta, \tag{2}$$

where $\alpha = \frac{\max(|X|)}{2^{N-1}-1}, \beta = 0$ for symmetric quantization, or $\alpha = \frac{\max(X)-\min(X)}{2^{N-1}-1}, \beta = \min(X)$ for asymmetric quantization. In both cases, the quantization scale α is heavily influenced by the extreme values in X, so outliers can severely degrade the reconstruction error $||X - \hat{X}||$. As a workaround, some prior works keep outliers in higher precision while compressing the remainder (Xiao et al., 2023; Lin et al., 2024), or they mitigate outliers by applying additional training (Liu et al., 2024a; Chen et al., 2024; Nrusimha et al., 2024).

To quantify those activation outliers, researchers commonly use *kurtosis*, specifically *excess kurtosis* (shkolnik et al., 2020; He et al., 2024; Caples and rrenaud, 2024; **?**). Excess kurtosis is defined as:

$$\text{Kurt}[X] - 3 = \mathbb{E}\left[\left(\frac{X - \mu}{\sigma}\right)^4\right] - 3, \tag{3}$$

where μ and σ are the mean and standard deviation of activation X, respectively.

5.2 Second-Order Optimization

Early second-order methods in deep learning were hindered by large computation and memory overhead, relying on relying on quasi-Newton or Hessian-based approximations (Nocedal and Wright, 1999). Kronecker-Factored Approximate Curvature (K-FAC) (Martens and Grosse, 2015) alleviated these costs by approximating the Fisher information matrix with Kronecker-factored statistics. Shampoo (Gupta et al., 2018) leverages the multi-dimensional (tensor) structure of parameters, maintaining and updating factorized preconditioners along each dimension of the gradient tensor, thereby improving scalability; its distributed version (Anil et al., 2020) demonstrated strong wall-clock performance.

Formally, let $W, G \in \mathbb{R}^{m \times n}$ denote a weight matrix and its gradient. Let $L \in \mathbb{R}^{m \times m}$ and $R \in \mathbb{R}^{n \times n}$ accumulate the statistics GG^T and G^TG,

respectively. Shampoo updates W as follows:

$$W \leftarrow W - \eta L^{-1/4} G R^{-1/4}, \tag{4}$$

where $\eta \in \mathbb{R}$ is the learning rate. Vyas et al. (2024) showed that Shampoo is equivalent to Adafactor (Shazeer and Stern, 2018) in the eigenbasis of its preconditioner, and introduced SOAP to adapt that eigenbasis to Adam.

More recently, Muon (Jordan et al., 2024) was proposed to orthogonalize gradients via the Newton–Schulz algorithm (Higham, 2008; Schulz, 1933), effectively approximating the Singular Value Decomposition (SVD) of gradients. Specifically, the Newton–Schulz iteration transforms

$$G = U\Sigma V^T \mapsto UV^T, \tag{5}$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are the singular vectors, and $\Sigma \in \mathbb{R}^{m\times n}$ is the diagonal matrix of singular values. Unlike Shampoo and SOAP, which require momentum buffers for both the preconditioner and its inverse, Muon maintains only a momentum buffer for the gradients. Subsequent works (Bernstein and Newhouse, 2024b,a; Duvvuri et al., 2024) has shown that Muon coincides with Shampoo in the absence of preconditioner accumulation.

5.3 Computational Invariance

Ashkboos et al. (2024a) introduced *computational invariance* in Transformer architectures, whereby inserting an orthogonal rotation into the residual stream and removing it later yields identical outputs. Modern large language models usually employ RMSNorm (Zhang and Sennrich, 2019) as a pre-normalization layer. Let $X \in \mathbb{R}^{m\times n}$ be an activation matrix, and $Q \in \mathbb{R}^{n\times n}$ be orthogonal. By definition of RMSNorm, the following holds:

$$\text{RMSNorm}(XQ)Q^T = \text{RMSNorm}(X). \tag{6}$$

Additionally, when linear layers follow RMSNorm, multiplication by Q^T can be merged with these layers via the associative property of matrix multiplication. By carefully placing Q and Q^T across normalization, projection, and embedding layers, one can preserve network-level invariance.

Building on this idea, QuaRot (Ashkboos et al., 2024b) applies a random Hadamard rotation to the Transformer's embedding space, while SpinQuant (Liu et al., 2024b) employs a learnable rotation matrix which is optimized via *Cayley SGD* (Li et al., 2020).

6 Conclusion

In this paper, we presented a pre-training pipeline that substantially mitigates activation outliers by replacing the Adam optimizer with Muon, introducing learnable embedding rotations, and adopting a single-scale RMSNorm. Unlike many previously proposed solutions, our method achieves almost outlier-free training without expensive quantization-aware fine-tuning or specialized architectural blocks that complicate model deployment. The transition away from Adam alleviates the basis-alignment issue that underpins outlier formation, while our framework eliminates residual extremes without negatively impacting convergence.

Extensive experiments on a 1.4B-parameter LLM trained over 1 trillion tokens confirm the reliability and efficiency of our approach. In particular, the resulting model remains robust under 4-bit quantization, outperforming comparable open-source models that rely on Adam. These findings motivate further exploration of how optimizer choices influence model behavior and quantization readiness. We believe our work on **Outlier-Safe Pre-Training** will encourage broader adoption of outlier-aware training practices as the field continues to develop larger, more efficiently deployable language models.

Limitations

Our study focused primarily on Muon without extensive comparisons to other second-order methods like Shampoo or SOAP. This limitation stems from practical constraints: TPU compilation times for training pipelines often exceed one hour, making comprehensive optimizer ablation studies prohibitively time-consuming given our available computational resources.

Additionally, while our experiments demonstrate effectiveness on a 1.4B-parameter model, we have not yet explored the impact across a range of model sizes, particularly the 3B and 7B parameter scales commonly targeted for mobile deployment. Looking ahead, we plan to extend our analysis to these larger models. Our distributed implementation of Muon in JAX achieves comparable efficiency to Adam, making such broader experiments computationally feasible.

References

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. 2025. Smollm2: When smol goes big – data-centric training of a small language model. *Preprint*, arXiv:2502.02737.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Leandro von Werra, and Thomas Wolf. 2024. Smollm - blazingly fast and remarkably powerful.

Rohan Anil, Vineet Gupta, Tomer Koren, Kevin Regan, and Yoram Singer. 2020. Scalable second order optimization for deep learning. *arXiv preprint arXiv:2002.09018*.

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. 2024a. SliceGPT: Compress large language models by deleting rows and columns. In *The Twelfth International Conference on Learning Representations*.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. 2024b. Quarot: Outlier-free 4-bit inference in rotated llms. In *Advances in Neural Information Processing Systems*, volume 37, pages 100213–100240. Curran Associates, Inc.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Marco Bellagente, Jonathan Tow, Dakota Mahan, Duy Phung, Maksym Zhuravinskyi, Reshinth Adithyan, James Baicoianu, Ben Brooks, Nathan Cooper, Ashish Datta, et al. 2024. Stable lm 2 1.6 b technical report. *arXiv preprint arXiv:2402.17834*.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. 2024. Cosmopedia.

Jeremy Bernstein and Laker Newhouse. 2024a. Modular duality in deep learning. *arXiv preprint arXiv:2410.21265*.

Jeremy Bernstein and Laker Newhouse. 2024b. Old optimizer, new norm: An anthology. *arXiv preprint arXiv:2409.20325*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. 2021. Understanding and overcoming the challenges of efficient transformer quantization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7947–7969, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Diego Caples and rrenaud. 2024. Adam optimizer causes privileged basis in transformer lm residual stream. *LessWrong*.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M De Sa. 2023. Quip: 2-bit quantization of large language models with guarantees. In *Advances in Neural Information Processing Systems*, volume 36, pages 4396–4429. Curran Associates, Inc.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. 2024. Efficientqat: Efficient quantization-aware training for large language models. *arXiv preprint arXiv:2407.11062*.

Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, et al. 2023. Mobilevlm: A fast, reproducible and strong vision language assistant for mobile devices. *arXiv preprint arXiv:2312.16886*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

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. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Gpt3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, pages 30318–30332. Curran Associates, Inc.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Sai Surya Duvvuri, Fnu Devvrit, Rohan Anil, Cho-Jui Hsieh, and Inderjit S Dhillon. 2024. Combining axes preconditioners through kronecker approximation for deep learning. In *The Twelfth International Conference on Learning Representations*.

Nelson Elhage, Robert Lasenby, and Christopher Olah. 2023. Privileged bases in the transformer residual stream. *Transformer Circuits Thread*.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. 2024. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.

Vineet Gupta, Tomer Koren, and Yoram Singer. 2018. Shampoo: Preconditioned stochastic tensor optimization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1842–1850. PMLR.

Alex Hägele, Elie Bakouch, Atli Kosson, Loubna Ben allal, Leandro Von Werra, and Martin Jaggi. 2024. Scaling laws and compute-optimal training beyond fixed training durations. In *Advances in Neural Information Processing Systems*, volume 37, pages 76232–76264. Curran Associates, Inc.

Bobby He, Lorenzo Noci, Daniele Paliotta, Imanol Schlag, and Thomas Hofmann. 2024. Understanding and minimising outlier features in transformer training. In *Advances in Neural Information Processing Systems*, volume 37, pages 83786–83846. Curran Associates, Inc.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

NJ Higham. 2008. Functions of matrices: Theory and computation.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cecista, Laker Newhouse, and Jeremy Bernstein. 2024. Muon: An optimizer for hidden layers in neural networks.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Jun Li, Fuxin Li, and Sinisa Todorovic. 2020. Efficient riemannian optimization on the stiefel manifold via the cayley transform. In *International Conference on Learning Representations*.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2023. Starcoder: may the source be with you! *arXiv preprint arXiv:2305.06161*.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. In *Proceedings of Machine Learning and Systems*, volume 6, pages 87–100.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human

falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2024a. LLM-QAT: Data-free quantization aware training for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 467–484, Bangkok, Thailand. Association for Computational Linguistics.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. 2024b. Spinquant–llm quantization with learned rotations. *arXiv preprint arXiv:2405.16406*.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. Fineweb-edu: the finest collection of educational content.

James Martens and Roger Grosse. 2015. Optimizing neural networks with kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2408–2417, Lille, France. PMLR.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Depen Morwani, Itai Shapira, Nikhil Vyas, Eran Malach, Sham Kakade, and Lucas Janson. 2024. A new perspective on shampoo's preconditioner. *arXiv preprint arXiv:2406.17748*.

Jorge Nocedal and Stephen J Wright. 1999. *Numerical optimization*. Springer.

Aniruddha Nrusimha, Mayank Mishra, Naigang Wang, Dan Alistarh, Rameswar Panda, and Yoon Kim. 2024. Mitigating the impact of outlier channels for language model quantization with activation regularization. *arXiv preprint arXiv:2404.03605*.

Xu Ouyang, Tao Ge, Thomas Hartvigsen, Zhisong Zhang, Haitao Mi, and Dong Yu. 2024. Low-bit quantization favors undertrained llms: Scaling laws for quantized llms with 100t training tokens. *arXiv preprint arXiv:2411.17691*.

Zhen Qin, Dong Li, Weigao Sun, Weixuan Sun, Xuyang Shen, Xiaodong Han, Yunshen Wei, Baohong Lv, Fei Yuan, Xiao Luo, et al. 2023. Scaling transnormer to 175 billion parameters. *arXiv preprint arXiv:2307.14995*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Günther Schulz. 1933. Iterative berechung der reziproken matrix. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 13(1):57–59.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. 2024. Omniquant: Omnidirectionally calibrated quantization for large language models. In *The Twelfth International Conference on Learning Representations*.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

moran shkolnik, Brian Chmiel, Ron Banner, Gil Shomron, Yury Nahshan, Alex Bronstein, and Uri Weiser. 2020. Robust quantization: One model to rule them all. In *Advances in Neural Information Processing Systems*, volume 33, pages 5308–5317. Curran Associates, Inc.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *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)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. 2024. QuIP: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 48630–48656. PMLR.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. 2024. Soap: Improving and stabilizing shampoo using adam. *arXiv preprint arXiv:2409.11321*.

Xiuying Wei, Yunchen Zhang, Xiangguo Zhang, Ruihao Gong, Shanghang Zhang, Qi Zhang, Fengwei Yu, and Xianglong Liu. 2022. Outlier suppression: Pushing the limit of low-bit transformer language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 17402–17414. Curran Associates, Inc.

Kaiyue Wen, Zhiyuan Li, Jason Wang, David Hall, Percy Liang, and Tengyu Ma. 2024. Understanding warmup-stable-decay learning rates: A river valley loss landscape perspective. *arXiv preprint arXiv:2410.05192*.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 38087–38099. PMLR.

Yuanzhong Xu, HyoukJoong Lee, Dehao Chen, Hongjun Choi, Blake Hechtman, and Shibo Wang. 2020. Automatic cross-replica sharding of weight update in data-parallel training. *arXiv preprint arXiv:2004.13336*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024b. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *Preprint*, arXiv:2401.02385.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Yilong Zhao, Chien-Yu Lin, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind Krishnamurthy, Tianqi Chen, and Baris Kasikci. 2024. Atom: Low-bit quantization for efficient and accurate llm serving. In *Proceedings of Machine Learning and Systems*, volume 6, pages 196–209.

A Appendix

A.1 Comprehensive Benchmark Results for Open-Source LLMs

Table 5 presents the performance of various open-source LLMs across 11 benchmarks using W16A16 precision (16-bit weights and activations, without quantization). Notably, our Muon-trained model achieves comparable performance to Adam-trained models, marking the first successful application of Muon to trillion-token scale training.

A.2 Detailed Distribution Analysis

For a comprehensive view of activation and weight distributions, we provide detailed histograms in Figures 5, 6, 7, and 8. These visualizations allow direct comparison between Adam and Muon training approaches.

Model	Params.	Tokens	ARC	CSQA	GSM8K	HS	MMLU	OBQA	PIQA	SIQA	TQA	TFQA	WG	Avg.
Pythia	1.4B	0.3T	41.27	35.38	2.43	50.81	31.33	34.6	71.06	43.45	9.22	38.99	55.17	37.61
TinyLlama	1.1B	2T	36.51	25.39	1.74	53.98	32.64	23.0	70.29	41.30	23.48	35.44	49.96	35.79
OPT	1.3B	0.3T	39.33	39.97	0.91	52.22	29.58	35.8	71.00	42.27	11.14	38.86	53.28	37.67
OLMo	1.2B	3T	44.21	40.38	1.67	60.36	31.93	37.8	75.19	44.11	17.57	32.90	53.43	39.96
MobileLLAMA	1.4B	1.3T	42.65	37.02	1.97	54.18	31.78	34.4	73.29	43.04	24.48	34.95	55.41	39.38
Qwen 1.5	1.8B	2.4T	46.85	32.92	34.19	59.52	33.14	37.2	74.32	44.47	18.76	39.37	57.93	43.52
Qwen 2	1.5B	7T	48.18	30.96	58.07	63.89	37.42	36.8	75.35	44.22	23.99	45.85	59.19	47.63
Qwen 2.5	1.5B	–	58.77	34.32	61.56	66.47	40.26	39.6	75.68	44.88	20.59	46.67	59.43	49.84
LLAMA 3.2	1.2B	–	49.24	41.11	5.99	61.31	36.26	39.0	74.92	43.45	20.72	38.51	58.09	42.60
Stable LM 2	1.6B	2T	53.46	34.56	19.26	66.67	35.98	37.0	76.82	43.50	35.59	38.76	59.19	45.53
SmolLM	1.7B	1T	59.69	38.00	6.75	63.02	39.36	42.8	75.95	44.11	25.84	38.55	54.54	44.42
SmolLM 2	1.7B	11T	60.38	43.57	32.60	68.70	41.30	42.4	77.58	43.40	27.08	36.70	60.14	48.53
Trained from scratch														
Adam	1.4B	1T	59.50	40.62	14.48	63.97	39.52	41.0	76.06	43.60	23.91	40.88	56.59	45.47
Muon (OSP)	1.4B	1T	57.68	38.74	14.25	62.43	38.91	41.0	75.35	44.73	24.46	39.54	55.56	44.79

Table 5: W16A16 quantization performance comparison across 12 open-source LLMs and our implementations. **Params.** indicates model size in parameters, and **Tokens** shows training dataset size. Benchmarks include CSQA (CommonsenseQA), HS (HellaSwag), OBQA (OpenBookQA), TQA (TriviaQA), TFQA (TruthfulQA), and WG (WinoGrande).

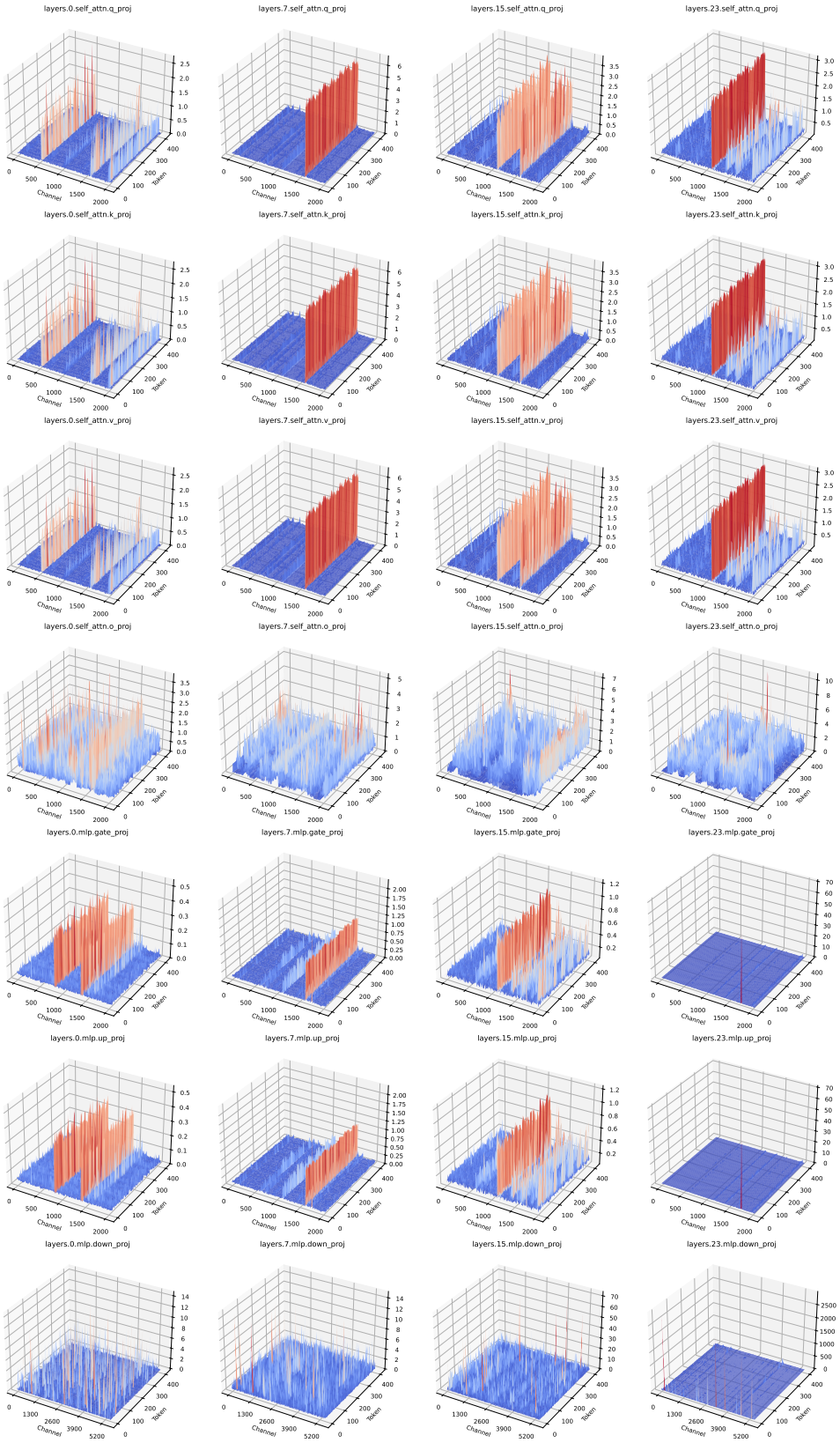


Figure 5: Activation histograms from Adam-trained models on 1 trillion tokens. The plots show input distributions to Multi-Head Self-Attention (MHSA) and Feed-Forward Network (FFN) layers at four different depths: 0th, 7th, 15th, and 23rd transformer blocks.

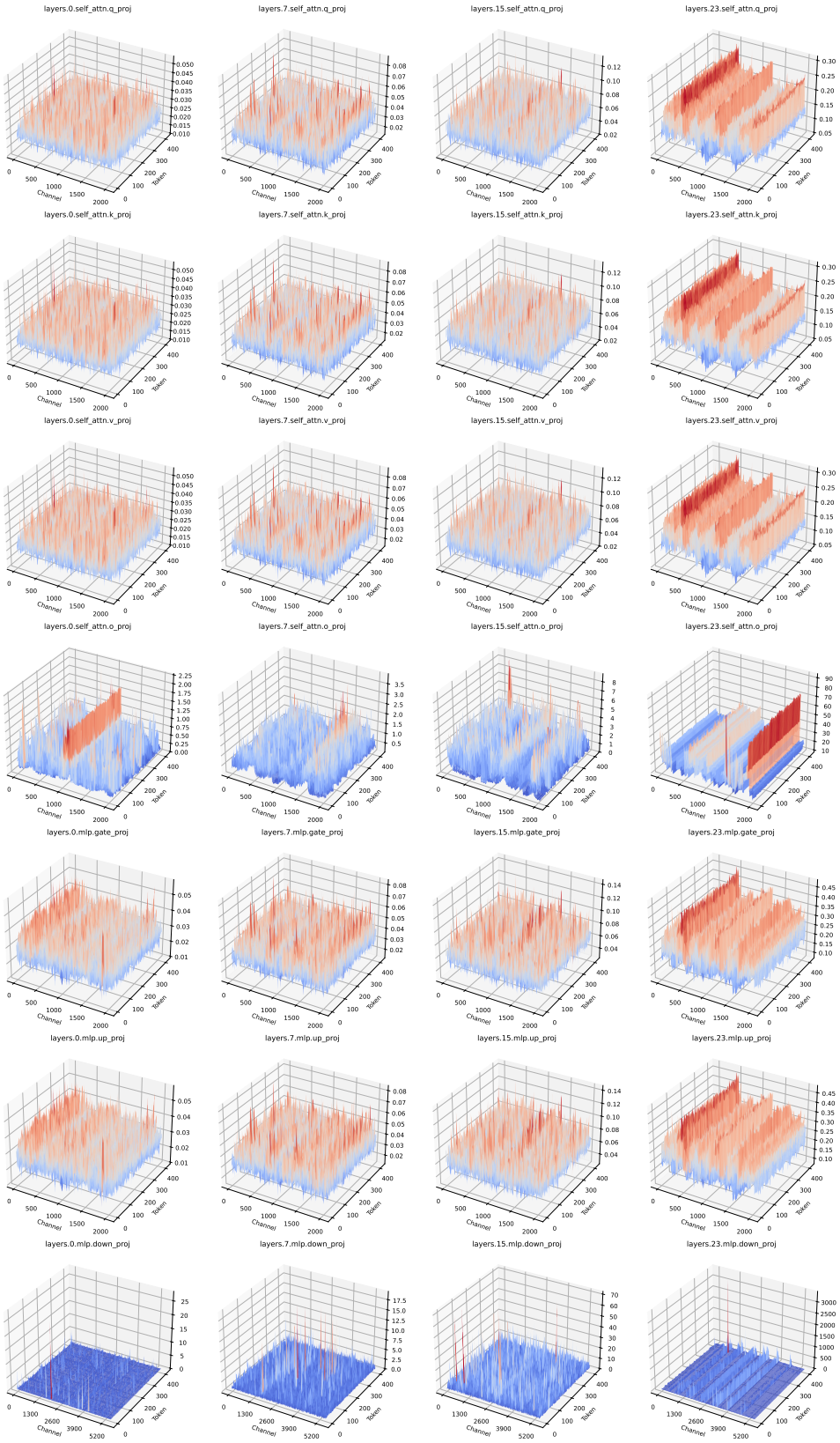


Figure 6: Activation histograms from Muon-trained models with **OSP** on 1 trillion tokens. The plots show input distributions to Multi-Head Self-Attention (MHSA) and Feed-Forward Network (FFN) layers at four different depths: 0th, 7th, 15th, and 23rd transformer blocks.

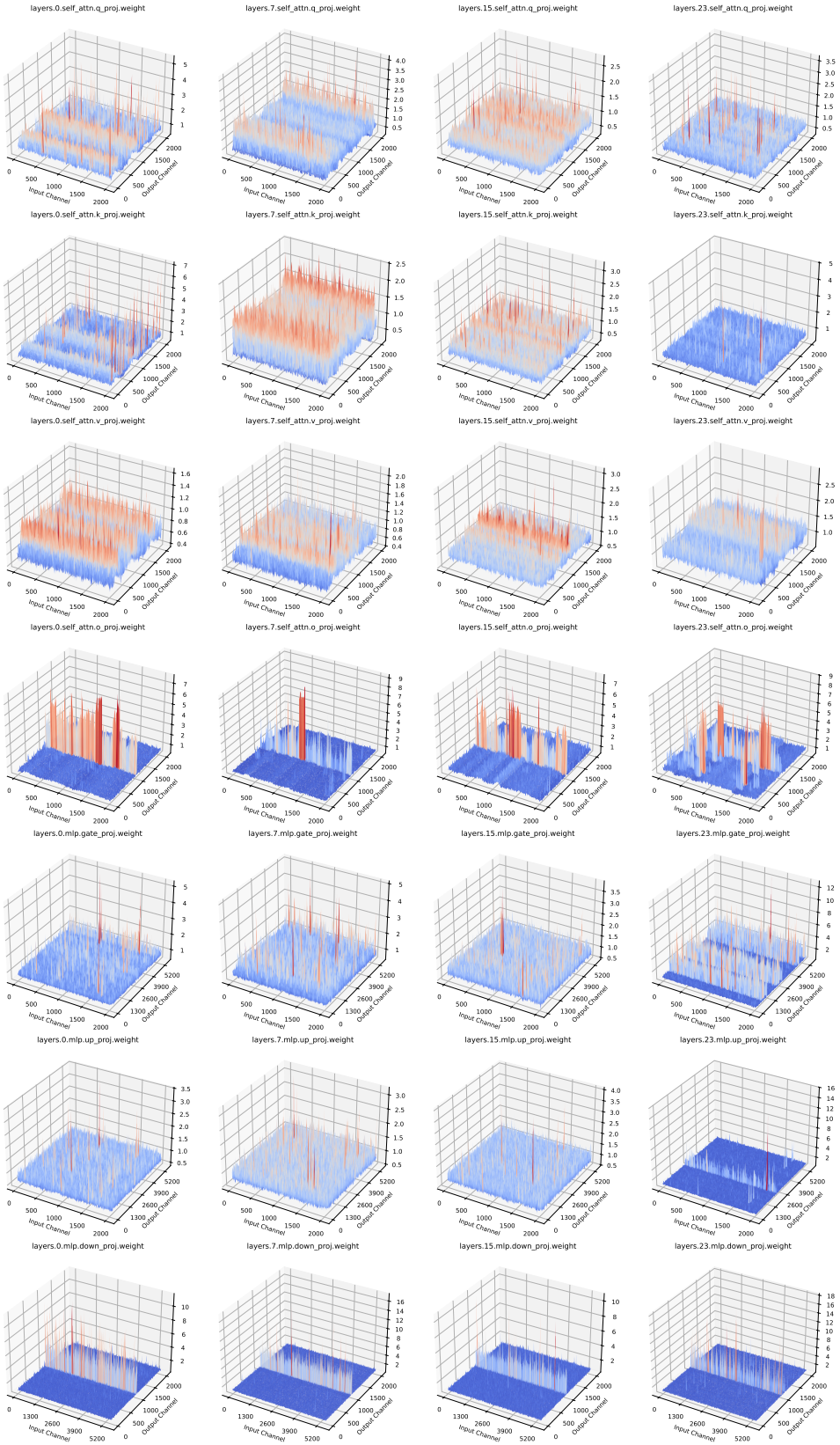


Figure 7: Weight histograms from Adam-trained models on 1 trillion tokens. The plots show weight distributions in Multi-Head Self-Attention (MHSA) and Feed-Forward Network (FFN) layers across four different depths: 0th, 7th, 15th, and 23rd transformer blocks.

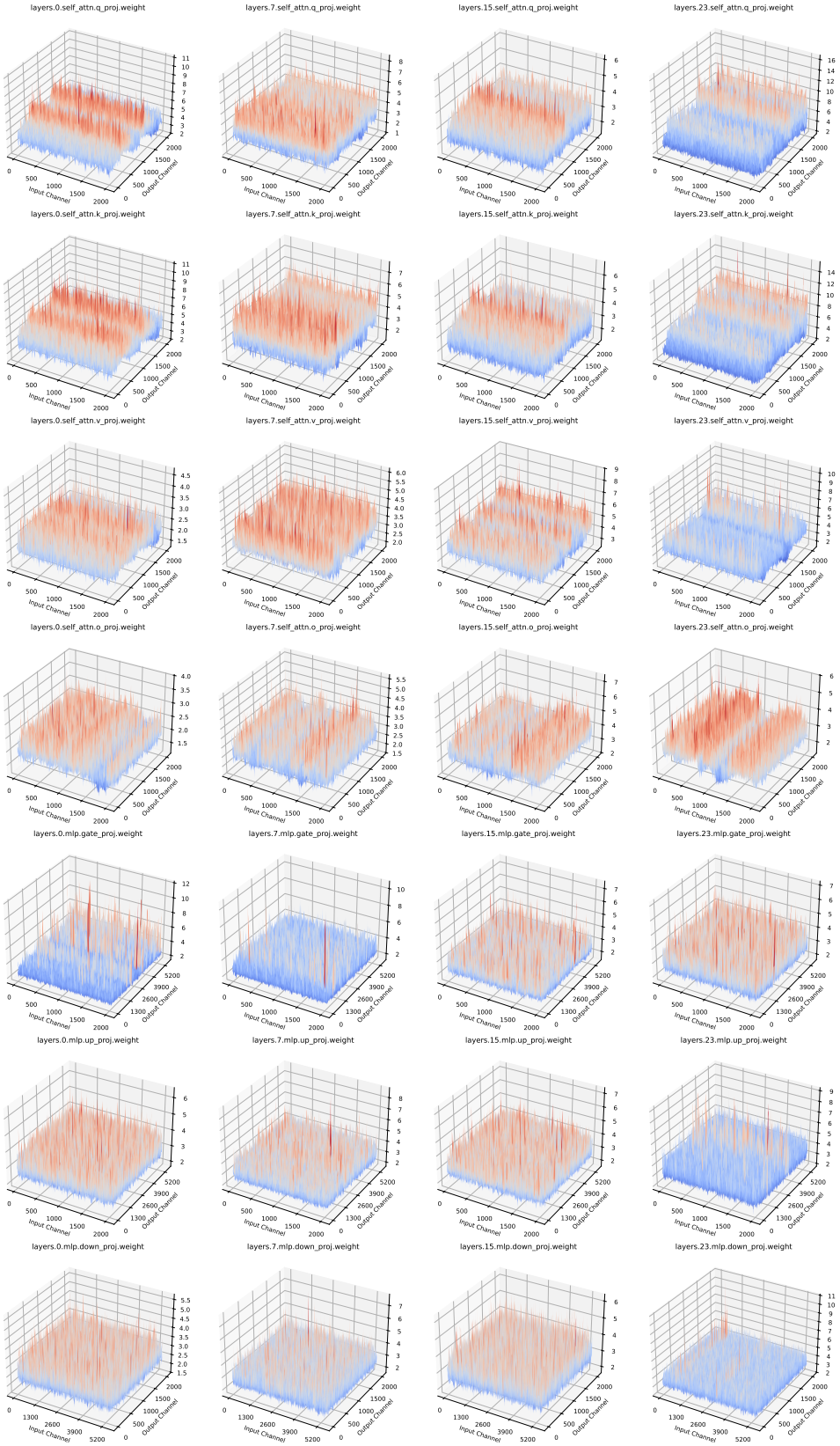


Figure 8: Weight histograms from Muon-trained models with **OSP** on 1 trillion tokens. The plots show weight distributions in Multi-Head Self-Attention (MHSA) and Feed-Forward Network (FFN) layers across four different depths: 0th, 7th, 15th, and 23rd transformer blocks.