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

Jungwoo Park[1,2], **Taewhoo Lee**[1,2], **Chanwoong Yoon**[1]
Hyeon Hwang[1], **Jaewoo Kang**[1,2*]
[1]Korea University, [2]AIGEN Sciences
{jungwoo-park, taewhoo, cwyoon99, hyeon-hwang, kangj}@korea.ac.kr

Abstract

Extreme activation outliers in Large Language Models (LLMs) critically degrade quantization performance, hindering efficient on-device deployment. While channel-wise operations and adaptive gradient scaling are recognized causes, practical mitigation remains challenging. We introduce **Outlier-Safe Pre-Training (OSP)**, a practical guideline that proactively prevents outlier formation, rather than relying on post-hoc mitigation. OSP combines three key innovations: (1) the Muon optimizer, eliminating privileged bases while maintaining training efficiency, (2) Single-Scale RMSNorm, preventing channel-wise amplification, and (3) a learnable embedding projection, redistributing activation magnitudes. We validate OSP by training a 1.4B-parameter model on 1 trillion tokens, which is the first production-scale LLM trained without such outliers. Under aggressive 4-bit quantization, our OSP model achieves a 35.7 average score across 10 benchmarks (versus 26.5 for an Adam-trained model), with only a 2% training overhead. Remarkably, OSP models exhibit near-zero excess kurtosis (0.04) compared to extreme values (1818.56) in standard models, fundamentally altering LLM quantization behavior. Our work demonstrates that outliers are not inherent to LLMs but are consequences of training strategies, paving the way for more efficient LLM deployment. The source code and pretrained checkpoints are available at https://github.com/dmis-lab/Outlier-Safe-Pre-Training.

1 Introduction

Quantization has emerged as a practical solution for deploying Large Language Models (LLMs) in resource-constrained environments (Dettmers et al., 2022). As modern LLMs often scale to hundreds of billions of parameters, reducing the bit-width of weights and activations can significantly decrease memory consumption during inference. However, the pervasive presence of outliers within LLM architectures presents a fundamental obstacle to effective low-bit quantization.

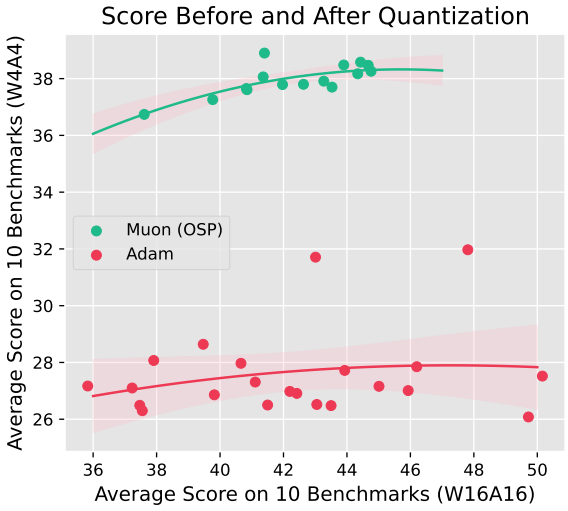


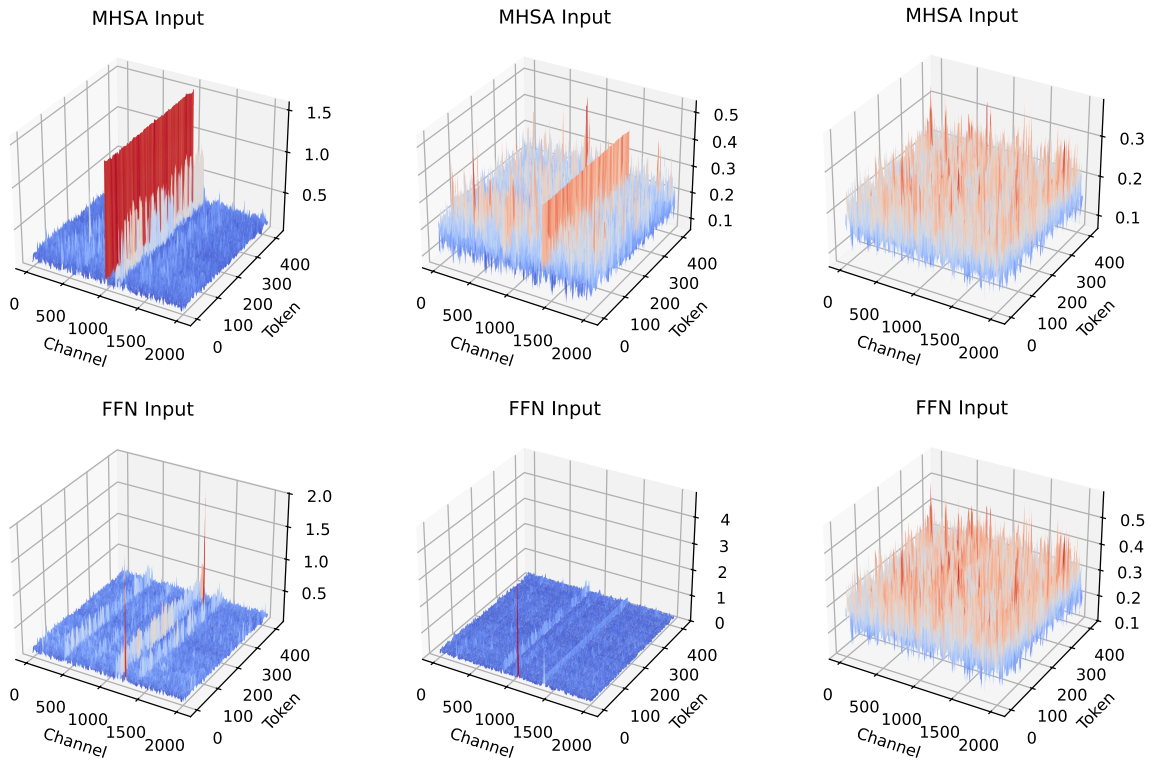
Figure 1: Comparison of performance degradation patterns under 4-bit quantization across different training methodologies. **Adam** represents various open-source LLMs trained with the Adam optimizer. Muon (OSP) represents checkpoints from our model trained within the **Outlier-Safe Pre-Training (OSP)** framework. Results demonstrate that our framework produces fundamentally different degradation characteristics compared to conventional Adam-trained models.

Contemporary LLMs trained with standard optimization techniques invariably develop outlier features during pre-training, posing persistent challenges for quantization (Bondarenko et al., 2023; He et al., 2024). Post-Training Quantization (PTQ) methods have gained substantial attention for mitigating these outliers at inference time, enabling immediate deployment without requiring costly retraining. While PTQ offers a pragmatic solution, it remains a reactive rather than preventive approach, implicitly accepting outliers as an intrinsic property of LLMs instead of addressing their root cause.

This raises a critical question: are outliers an

*Corresponding author

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12582–12600



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

Figure 2: Activation distribution analysis from the 20th layer input to Multi-Head Self-Attention (MHSA) and Feed-Forward Network (FFN) modules in 1.4 billion parameter models trained on 100 billion tokens. Three optimization strategies are examined: (a) standard Adam optimizer, (b) Muon optimizer without architectural modifications, and (c) the **Outlier-Safe Pre-Training (OSP)** framework. The histograms reveal that substituting Adam with Muon optimizer alone provides insufficient mitigation of activation outliers.

inevitable consequence of large-scale language model training? Returning to first principles, the ideal solution for robust quantization would be to train LLMs in a way that inherently prevent outlier formation during pre-training. Several recent studies have investigated the origins of outlier features in LLMs, suggesting three distinct perspectives on their root causes: channel-wise scaling factors in normalization layers (Kovaleva et al., 2021; Wei et al., 2022; He et al., 2024), the attention sink phenomena (Bondarenko et al., 2023; Guo et al., 2024; Gu et al., 2025), and diagonal optimizers such as Adam and AdaFactor (He et al., 2024; Guo et al., 2024). While each study has proposed methods to mitigate outliers at the pre-training stage based on these findings, existing approaches remain preliminary, lacking consideration for production-level scalability and practicality. Most are limited to models with fewer than one billion parameters or training corpora of under 100 billion tokens, and often overlook the computational overhead of alternative optimization strategies as well as compatibility issues introduced by architectural modifications.

Therefore, we present the **Outlier-Safe Pre-Training (OSP)** framework, a practical guideline that integrates existing findings to enable the development of quantization-friendly models at industrial scale. Our approach prioritizes three key objectives: (1) scaling to production-level training requirements, (2) maintaining computational efficiency comparable to standard methods, and (3) ensuring complete architectural compatibility with existing inference pipelines.

Our empirical results demonstrate that the model trained with the **OSP** framework remains free of outlier features over the course of training on one trillion tokens. Our methodology reduces memory usage by 33% compared to standard approaches and maintains competitive performance, with only a 2% increase in training time. Under aggressive 4-bit weight and activation quantization, our model significantly outperforms comparable open-source alternatives. Moreover, our approach is orthogonal to existing PTQ techniques, enabling complementary performance gains when combined with post-training quantization.

Beyond quantization performance, our work provides new insights into outlier-free model behavior. Notably, our analysis reveals that attention sinks persist even in the absence of outliers, suggesting they are not inherently responsible for outlier formation. While Bondarenko et al. (2023) has attributed outliers to attention sinks implemented via extreme negative attention logits, our outlier-free models exhibit similar behavior via concentrated positive attention on specific tokens, without the emergence of massive activations. This finding challenges existing assumptions about the relationship between attention mechanisms and outliers. By publicly releasing the first production-scale outlier-free LLM, we enable further investigation into how the absence of outliers affects other emergent properties of large language models.

2 Related Works

2.1 Quantization

Quantization reduces the precision of weights and activations by mapping continuous floating-point values to discrete integers. For an n-bit quantization, floating-point value x is mapped to a discrete integer representation through:

$$\hat{x}_{\text{int}} = \text{clip}\left(\left\lfloor \frac{x}{s} \right\rceil + z, 0, 2^n - 1\right), \qquad (1)$$

where s denotes the scale factor and z represents the zero-point offset. The presence of outliers catastrophically inflates the scale factor, resulting in severe rounding errors and substantial information loss in low-bit settings.

To mitigate this challenge, the research community has pursued two primary directions. Quantization-Aware Training (QAT) enhances model robustness to quantization during the training process (Liu et al., 2024b; Chen et al., 2024). However, rather than eliminating outliers directly, QAT simulates quantization errors during training to improve tolerance to discretization errors. Moreover, this simulation significantly impedes training, making it impractical for large-scale deployment.

In contrast, Post-Training Quantization (PTQ) improves quantization performance of pre-trained models without additional training. PTQ encompasses various techniques: allocating higher precision exclusively to outlier channels (Dettmers et al., 2022; Kim et al., 2024), leveraging the Hessian of quantization error for optimal weight rounding (Frantar et al., 2023), and utilizing activation statistics (Xiao et al., 2023; Lin et al., 2024). A particularly notable advancement involves applying random rotation matrices, which effectively redistribute outliers across different channels for both activations and weights without requiring architectural modifications (Chee et al., 2023; Tseng et al., 2024; Ashkboos et al., 2024b).

Despite their sophistication, these PTQ methods fundamentally accept outliers as an intrinsic characteristic of LLMs, operating under the assumption that specific outlier channels persist across all token positions. Rather than preventing outlier formation, PTQ techniques represent sophisticated workarounds that manage the symptoms while leaving the underlying cause unaddressed.

2.2 Massive Activations and Attention Sinks

Meanwhile, recent studies have identified a distinct class of outliers in LLMs, referred to as *massive activations* (Sun et al., 2024). Unlike conventional outliers that appear in specific channels, massive activations exhibit sporadic behavior, emerging unpredictably in particular layers and predominantly within start tokens or delimiter tokens (Sun et al., 2024; Barbero et al., 2025).

Emerging evidence indicates a strong correlation between massive activations and *attention sinks* (Xiao et al., 2024), where attention layers excessively focus on specific tokens throughout the sequence. Bondarenko et al. (2023) posit that attention layers occasionally perform partial residual state updates by assigning attention weights to semantically unimportant tokens, effectively implementing a "no-op" operation. To achieve near-zero softmax values for most tokens, models drive attention logits toward negative infinity, inadvertently generating massive activations as a byproduct of this cancellation.

This understanding has motivated several research directions aimed at mitigating attention sinks to address quantization challenges. These approaches primarily target the attention mechanism through architectural modifications (Bondarenko et al., 2023; Guo et al., 2024; Gu et al., 2025) or adjustments to key-value caches (Liu et al., 2024a; Son et al., 2024). Nevertheless, empirical results demonstrate only limited performance gain, and growing evidence suggests that massive activations and outlier features underlie different mechanisms (Sun et al., 2024; Guo et al., 2024).

2.3 Privileged Bases and Diagonal Optimizers

An alternative interpretation of outlier emergence in neural networks centers on the concept of *privileged bases* (Elhage et al., 2023). Element-wise operations such as non-linear activation functions and channel-wise scaling factors in normalization layers induce a form of basis alignment wherein certain channels disproportionately accumulate magnitude. This observation has motivated previous efforts to manipulate normalization layers for outlier mitigation (Kovaleva et al., 2021; Wei et al., 2022; He et al., 2024).

Remarkably, recent studies suggest that the primary cause of outlier formation may not lie in architectural design, but rather in the adaptive gradient scaling mechanisms of optimizers (Elhage et al., 2023; Caples and Neuhaus, 2024; He et al., 2024; Guo et al., 2024). Optimizers such as RMSProp, Adam, and AdaFactor maintain running statistics of per-parameter gradient variance, applying element-wise standardization during parameter updates. This scaling strategy, often called diagonal preconditioning, introduces a preferential basis that can encourage outlier emergence.

Second-order optimizers that leverage Hessian matrix are proposed as alternatives to diagonal optimizers. By incorporating loss landscape curvature rather than rescaling gradient elements, these methods achieve faster convergence than first-order optimizers. K-FAC (Martens and Grosse, 2015) approximates the Hessian matrix using the Fisher information and constructs block-diagonal approximations layer-wise through Kronecker factorization instead of computing curvature for all parameters. Shampoo (Gupta et al., 2018) and SOAP (Vyas et al., 2024) further reduce computational complexity by introducing separate preconditioners for each tensor dimension.

3 Outlier-Safe Pre-Training

While recent investigations have proposed explanations for the emergence of outliers in LLMs, translating these insights into production-scale solutions remains challenging. Architectural modifications that eliminate privileged bases, including gating modules (Bondarenko et al., 2023) or the normalization layer removal (He et al., 2024), break compatibility with established inference frameworks.

Similarly, non-diagonal optimizers still impose prohibitive computational costs even with various approximation techniques. Our experiments show that second-order methods require three times the memory footprint of Adam and reduce training throughput by 25% (see Table 1). Such overheads make them impractical for trillion-token scale pre-training scenarios.

Optimizer	TPS (Relative)	Memory Usage	Build Time
Adam	4.07M (100%)	$O\left(36LD^2\right)$	2m 30s
Muon	3.99M (97.9%)	$O\left(24LD^2\right)$	3m 48s
Shampoo†	3.07M (75.5%)	$O\left(\frac{338}{3}LD^2\right)$	24m 24s
SOAP	–	$O\left(\frac{302}{3}LD^2\right)$	$\geq$ 3 hours

Table 1: Training throughput comparison across different optimizers estimated on TPU-v4 512 Pod Slice infrastructure. **TPS** denotes tokens processed per second during training operations. Theoretical memory usage requirements and code compilation time (**Build Time**) on JAX (Bradbury et al., 2018) framework are reported.

We present the **Outlier-Safe Pre-Training (OSP)** framework, which synthesizes prior insights while operating within production constraints. Our approach comprises three components that maintain the training costs and preserve the canonical transformer design, ensuring compatibility with existing inference systems and post-training techniques for complementary performance gains.

3.1 Integration of the Muon Optimizer

The foundation of our framework rests upon the integration of the Muon optimizer (Jordan et al., 2024), which represents a fundamental departure from diagonal preconditioning paradigms. Using the Newton-Schulz algorithm (Schulz, 1933; Higham, 2008), Muon iteratively transforms the gradient matrix to approximate orthogonalization according to:

$$G = U\Sigma V^T \mapsto UV^T, \tag{2}$$

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.

This algorithm distinguishes Muon from adaptive optimizers including Adam and AdaFactor. Muon maintains only gradient momentum without element-wise scaling and applies parameter updates through full-rank linear transformations. This algorithm eliminates privileged bases inherent to diagonal preconditioning, preventing systematic channel amplification and outlier occurrence.

Although technically classified as first-order optimization, Muon exhibits convergence properties comparable to second-order optimizers (Jordan

et al., 2024). Recent theoretical analyses (Bernstein and Newhouse, 2024b,a; Duvvuri et al., 2024) have established that Muon's update rule converges to that of Shampoo in the limiting case where preconditioner accumulation is disabled. This equivalence elucidates Muon's efficiency gains, which are achieved without explicit Hessian computation or gradient preconditioning, reaching 97.9% of the training throughput of standard Adam (see Table

Prior to our investigation, the scalability of Muon to billion-parameter architectures trained on trillion-token corpora has remained unexplored. We provide the first empirical validation demonstrating that Muon successfully scales to production-level training while maintaining its outlier-prevention properties throughout the pretraining process.

3.2 Single-Scale RMSNorm

Despite eliminating optimizer-induced privileged bases via the Muon optimizer, channel-wise scaling factors within normalization layers constitute an explicit basis alignment (Wei et al., 2022; He et al., 2024; Nrusimha et al., 2024), necessitating careful architectural modifications to achieve comprehensive outlier prevention.

Qin et al. (2023) proposed Simple RMSNorm (SRMSNorm), which removes the entire learnable parameters and instead divides vectors by $\sqrt{d}$, where d denotes the vector dimensionality. While this approach demonstrates effectiveness in preventing outlier emergence (He et al., 2024), our initial experiment reveals preliminary limitations that impede practical deployment. Specifically, dividing normalized vectors by $\sqrt{d}$ causes severe activation magnitude suppression during initial training phases, substantially slowing convergence. Conversely, fixing the scale to 1.0 leads to training instability as the model matures.

This phenomenon indicates that transformer architectures require dynamic scaling of activation magnitudes during training. To address these limitations, we propose **Single-Scale RMSNorm (SSNORM)**, which introduces to uniformly control activation magnitude across all dimensions. The SSNORM layer is defined as:

$$\text{SSNORM}(x) = \gamma \frac{x}{\|x\|_2}, \tag{3}$$

where $\gamma \in \mathbb{R}$ represents the scaling parameter. This design enables adaptive adjustment of activation scales while eliminating the channel-wise multiplication. By constraining all dimensions to share a single scaling factor, SSNORM fundamentally prevents the emergence of privileged coordinates while maintaining stable optimization dynamics.

3.3 Decoupled Embedding Optimization

The final component of our framework addresses the computational challenges posed by embedding layers in modern LLMs. As vocabulary sizes continue to expand, embedding matrices have grown to constitute a substantial fraction of model parameters. The high dimensionality of these matrices presents significant computational bottlenecks for non-diagonal optimizers. Our empirical analysis reveals that applying orthogonalization to embedding layers incurs an additional 6% throughput degradation beyond the baseline computational cost.

To address this computational challenge, our framework maintains Adam optimization exclusively for the embedding layers. This approach aligns with Jordan et al. (2024), who demonstrate that decoupling embedding matrices from Muon and training them with Adam achieves better convergence properties. For all experiments, we adopt decoupled optimization as the default configuration for embedding layers within the OSP framework.

Since this decoupling strategy potentially reintroduces outliers through the embeddings, our framework further introduces a learnable full-rank embedding projection, positioned after the embedding layer and before the unembedding layer, which we refer to as EMBPROJ. These matrices, inspired by PTQ methods (Chee et al., 2023; Ashkboos et al., 2024b; Liu et al., 2024c), redistribute any emerging outliers across different dimensions, preventing their concentration and propagation through other layers. The matrices can be absorbed into their adjacent embeddings after training, maintaining computational invariance (Ashkboos et al., 2024a). We employ orthogonal initialization to maintain the initial norm distribution of embedding vectors, thereby preserving training dynamics from initialization.

4 Experiments

4.1 Outlier Quantification

Prior to conducting our experimental evaluation, we establish a systematic approach for quantifying outlier emergence within model activations. Following established practices in the previous litera-

Optimizer	SSNorm	EmbProj	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	✗	41.5	11.4	39.7	21.6	39.7	21.6	26.5	1e5	26.8	8e4
				✓	41.5	11.4	40.2	22.3	40.3	22.3	27.2	3e4	26.9	3e4
Muon† (w/o Adam)	✗	✗	361.35	✗	41.0	11.7	38.4	14.8	38.3	14.8	26.3	1e6	26.3	8e5
				✓	41.0	11.7	37.5	15.4	37.5	15.4	33.3	24.5	33.1	24.8
Muon	✗	✗	1575.12	✗	41.5	11.4	40.0	13.8	40.0	13.8	29.4	934.3	29.0	1e4
				✓	41.5	11.4	40.6	12.9	40.6	12.9	38.6	15.7	38.4	15.8
Muon	✓	✗	66.69	✗	**41.8**	**11.2**	**41.0**	12.4	**40.9**	12.4	36.6	43.3	36.4	44.2
				✓	**41.8**	**11.2**	**40.8**	12.2	**40.8**	12.2	38.6	33.7	38.3	34.1
Muon	✗	✓	703.23	✗	40.0	12.3	38.4	14.8	38.4	14.8	31.0	99.7	30.4	114.6
				✓	40.0	12.3	39.2	13.9	39.3	13.9	36.3	22.1	36.2	22.3
Muon (OSP)	✓	✓	**0.04**	✗	41.4	**11.2**	40.6	**12.2**	40.6	**12.2**	**37.9**	**19.4**	**37.5**	**19.6**
				✓	41.4	**11.2**	40.5	**12.1**	40.5	**12.1**	**39.1**	**13.4**	**38.9**	**13.5**

Table 2: Ablation study examining models trained on 100 billion tokens. **SSNorm** indicates single-scale RMSNorm integration, while **EmbProj** denotes models incorporating learnable embedding projection layers. **Ex. Kurt** represents excess kurtosis measurements, and **Had.** indicates online Hadamard transformation applied to Feed-Forward Network (FFN). Bit-width configurations (e.g., 16-16-16, 4-8-16) specify quantization precision for weights, activations, and key-value cache respectively. **Avg.** presents average performance across 10 LLM benchmarks, while **PPL** reports WikiText-2 perplexity. †Model configuration that disables decoupled embedding optimization by training with Muon optimizer without Adam optimization on embedding layers (Section 3.3).

ture (He et al., 2024; Caples and Neuhaus, 2024), we employ *excess kurtosis* to quantify the degree of outlier concentration:

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

where μ and σ denote the mean and standard deviation of activation X, respectively. This metric effectively captures the heavy-tailed nature of distributions containing outliers, with higher values indicating greater outlier presence.

4.2 Experimental Setup

We evaluate our framework through comprehensive experiments on 1.4B-parameter LLaMA (Touvron et al., 2023) architecture. Our experimental design examines quantization performance across different optimizers and architectural modifications to validate the effectiveness of our approach at scale.

We assess model performance using perplexity on WikiText-2 (Merity et al., 2016) and accuracy across 10 downstream benchmarks: ARC (Clark et al., 2018), CommonsenseQA (Talmor et al., 2019), GSM8k (8-shot) (Cobbe et al., 2021), 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 (5-shot) (Joshi et al., 2017), and WinoGrande (Sakaguchi et al., 2020). This extensive evaluation protocol ensures thorough assessment of both the quantization robustness and the general capabilities of models trained with the OSP framework.

4.3 Ablation Study

To systematically evaluate the contribution of each component within the OSP framework, we conduct comprehensive ablation studies on a subset of the training corpus comprising 100 billion tokens. This controlled experimental setting enables detailed analysis of how various architectural and optimization choices impact quantization robustness while maintaining computational feasibility.

We assess quantization robustness using two complementary approaches: round-to-nearest (RTN) quantization, and online Hadamard transformations within feed-forward network layers (Chee et al., 2023; Liu et al., 2024c) that rotate hidden states to address inherent quantization challenges. Each experimental configuration undergoes evaluation across four quantization scenarios (Table 2).

The results presented in Table 2 demonstrate that our pre-training strategy achieves the highest quantization resilience. This configuration attains near-zero excess kurtosis in activation distributions and maintains model performance under both RTN and Hadamard rotation-based quantization. The activation histograms displayed in Figure 2 particularly highlight the effectiveness of our approach, and the training dynamics illustrated in Figure 3 show that the OSP framework uniquely maintains low kur-

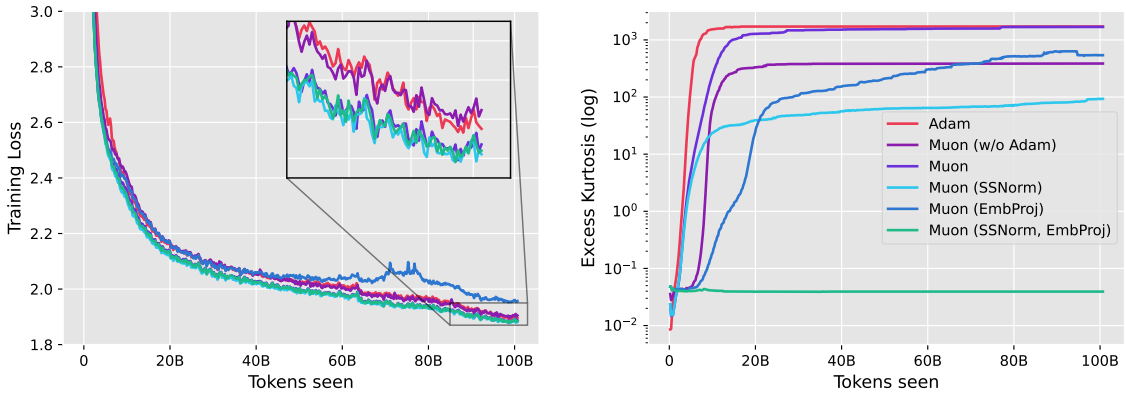


Figure 3: Training dynamics comparison showing loss (left) and excess kurtosis evolution (right) across 100 billion tokens for ablation study examining various configurations of OSP components. The excess kurtosis demonstrate that only when all OSP components are simultaneously enabled does the kurtosis remain near zero throughout training, indicating complete elimination of outlier formation. Partial implementation of OSP components results in insufficient outlier suppression, as evidenced by elevated kurtosis values that persist across the training duration.

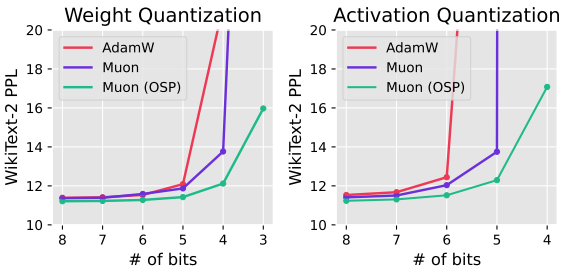


Figure 4: WikiText-2 perplexity under varying weight and activation quantization bit-widths for models trained on 100B tokens. Three configurations are compared: standard Adam, Muon, and the OSP framework.

tosis throughout training, providing evidence that our methodology fundamentally eliminates outliers rather than merely reducing them.

Table 2 also presents ablation studies that examine the individual contributions of OSP components: the Muon optimizer, SSNorm, and EmbProj. When using the Muon optimizer as the baseline, switching to the Adam optimizer results in a further increase in kurtosis. While both SSNorm and EmbProj independently reduce kurtosis substantially compared to Muon alone, the training logs in Figure 3 indicate that neither component using Muon alone is sufficient to prevent outlier emergence. This observation suggests that once outliers begin to form at any point in the network, they propagate throughout the entire architecture, ultimately degrading quantization performance.

Figure 4 provides comprehensive visualization of perplexity degradation on WikiText-2 across various quantization bit-widths. Our proposed method consistently preserves model performance across all quantization levels, with particularly exceptional performance in the challenging 4-bit regime. Notably, the stability of the performance degradation curve for weights down to 3-bit quantization suggests that our approach fundamentally transforms the model's quantization characteristics rather than merely shifting the performance degradation baseline.

4.4 Scaling Up to 1 Trillion Tokens

Having demonstrated the preliminary effectiveness of the OSP framework, we scale training to one trillion tokens, matching the scale commonly employed in production environments. While our 100-billion token experiments have confirmed the absence of outlier emergence, model behavior at substantially larger scales remains uncertain. Therefore, validating the effectiveness of each component at production scale constitutes a critical step in verifying the robustness of our framework.

Consistent with our preliminary findings, the kurtosis trajectory demonstrates that the OSP framework successfully prevents divergence throughout the entire training process. This result is particularly significant, as it confirms OSP's effectiveness at the corpus scale in actual production deployments. Unlike previous research limited to preliminary experimental scales, our OSP framework provides practical guidelines capable of preventing outlier formation at production scale. Further details are provided in Figure 7.

Subsequently, Table 3 presents comprehensive benchmark results for 4-bit quantization across 12 open-source LLMs of comparable scale. The majority of baseline models experience severe accuracy degradation under low-bit quantization, with

Model	Params.	Tokens	ARC	CSQA	GSM8K	HS	MMLU	OBQA	PIQA	SIQA	TQA	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	49.0	26.5
TinyLlama	1.1B	2T	28.3	22.9	0.0	26.6	26.2	21.2	48.7	40.7	0.0	49.0	26.4
OPT	1.3B	0.3T	25.0	21.6	0.0	26.5	25.6	28.2	49.6	36.9	0.0	49.5	26.3
OLMo	1.2B	3T	27.7	25.8	0.0	27.0	26.1	25.8	54.1	37.4	0.0	**51.9**	27.6
MobileLLaMA	1.4B	1.3T	27.4	23.5	0.0	26.7	26.0	22.4	49.6	38.3	0.0	49.6	26.4
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	49.3	27.4
Qwen 2	1.5B	7T	30.9	27.7	0.4	35.7	26.2	28.4	56.5	38.3	0.8	48.6	29.3
Qwen 2.5	1.5B	–	27.7	25.0	0.0	26.9	25.7	24.0	52.2	38.4	0.0	47.5	26.7
LLaMA 3.2	1.2B	–	29.3	24.7	**0.5**	30.1	25.8	27.4	53.3	39.5	0.1	50.3	28.1
Stable LM 2	1.6B	2T	26.2	24.0	0.0	27.0	24.6	27.0	51.1	37.8	0.0	51.1	26.9
SmolLM	1.7B	1T	28.4	25.7	0.0	27.0	26.1	28.0	51.0	38.8	0.0	48.4	27.3
SmolLM 2	1.7B	11T	25.8	22.4	0.0	25.9	24.2	26.6	51.5	36.0	0.0	50.0	26.2
From Scratch													
Adam	1.4B	1T	25.7	25.3	0.0	26.8	25.4	26.0	49.0	37.2	0.0	49.9	26.5
Muon (OSP)	1.4B	1T	**45.9**	**36.2**	**0.5**	**44.9**	**31.1**	**34.0**	**65.6**	**41.3**	**7.8**	49.8	**35.7**

Table 3: Performance evaluation of 4-bit quantization across 12 open-source large language models and our two models trained from scratch, assessed across 10 benchmark tasks. Model parameters (**Params.**) represent total trainable parameters, while **Tokens** indicate training dataset size. Evaluation benchmarks include CommonsenseQA (CSQA), HellaSwag (HS), OpenBookQA (OBQA), TriviaQA (TQA), and WinoGrande (WG), among others. Results demonstrate the impact of extreme quantization on model performance, with the model trained using the OSP framework exhibiting superior quantization robustness across the comprehensive benchmark suite.

Quantization	Adam	Muon (OSP)
RTN	14475.51	**38.92**
+ Hadamard†	4794.00	**18.71**
+ GPTQ	3723.46	**14.14**
+ SpinQuant	14.94	**13.66**

Table 4: WikiText-2 perplexity after applying various PTQ methods under 4-bit quantization. Minimal methods (Hadamard and GPTQ) show limited effectiveness on Adam. Models trained with OSP demonstrate consistently superior performance across all scenarios. †Only applies Hadamard transform to FFN hidden states.

performance on multiple-choice benchmarks such as ARC and CommonsenseQA deteriorating to near-random baselines (25%). In contrast, our approach demonstrates substantially stronger performance retention, indicating that our framework preserve quantization resilience more effectively.

5 Analysis

5.1 Complementary Benefits with Post-Training Quantization

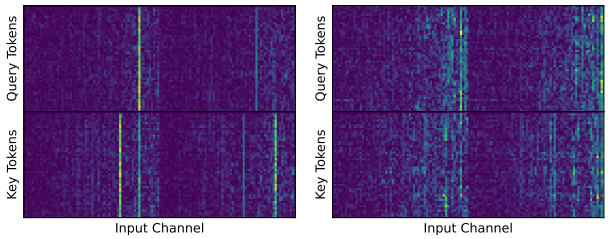
To further investigate the practical implications of our framework, we examine how models trained with OSP perform when combined with existing PTQ techniques. Unlike architectural modifications that alter model structure, our approach maintains computational invariance while exhibiting fundamentally different quantization characteristics. This raises an important question: does the absence of outliers eliminate the need for PTQ?

Table 4 demonstrates that our model trained under OSP achieves complementary performance gains when combined with PTQ methods. This finding is particularly significant because, unlike Quantization-Aware Training (QAT), our framework does not incorporate explicit quantization objectives during pre-training. Rather, our primary goal centers on eliminating the outlier features that fundamentally impede quantization, thereby creating models with inherently superior quantization robustness. While our framework demonstrates consistent perplexity improvements over RTN quantization across various PTQ scenarios, the gains are less pronounced than those achieved with Adam. Crucially, however, our approach consistently outperforms Adam across all evaluation settings. This pattern suggests that our approach provides a better foundation for subsequent PTQ calibration.

5.2 Attention Sinks without Outliers

We conduct a qualitative analysis to examine the internal dynamics of models when outliers are eliminated. Following previous studies that have conceptualized outliers as byproducts of attention sinks, we investigate whether the attention sink phenomenon disappears when massive activations are mitigated.

Following Bondarenko et al. (2021), we initially identify activations that exceed 6 standard deviations from the mean. As a result, models trained



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

Figure 5: Activation magnitudes of query and key tokens within attention sink heads comparing Adam and OSP models. Adam models exhibit concentrated activation patterns with sparse high-magnitude channels, while OSP demonstrates broadly distributed activation patterns across multiple channels.

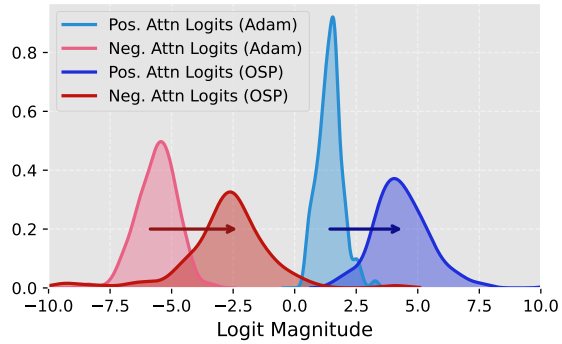


Figure 6: Attention logit distributions at sink token positions comparing Adam and OSP training. Adam models exhibit skewed distributions with predominantly negative logits, while OSP models demonstrate balanced distributions between sink tokens and other positions.

with Adam exhibit massive activations concentrated in delimiter tokens and similar positions, consistent with prior findings. In contrast, our framework demonstrates complete absence of such massive activations across all examined components.

Despite the elimination of massive activations, attention sink patterns still persist in our models. Following the analytical approach of Gu et al. (2025), we apply threshold-based filtering to identify cases where attention concentrates heavily on initial tokens. This analysis confirms that OSP-trained models continue to exhibit substantial attention sink behavior, raising important questions about the underlying mechanisms linking attention patterns to outlier formation.

To address this apparent contradiction, we conduct a preliminary analysis by examining the distributional characteristics of attention mechanisms in both training paradigms. Figure 5 visualizes the query and key activation magnitudes within attention heads that exhibit sink behavior. Models trained with Adam demonstrate a concentration of massive values within a small number of outlier channels, while the OSP-trained model distributes these magnitudes broadly across multiple channels.

This distributional difference is directly related to the attention logit patterns illustrated in Figure 6, which compares the logit distributions between sink tokens and other sequence positions. For models prone to massive activations, the concentration of large values in overlapping channels generates predominantly negative logits across most tokens. Conversely, OSP-trained models achieve more uniform distributions, resulting in balanced logits between sink tokens and the broader token sequence.

Since softmax normalization operates on relative logit differences rather than absolute values, models can achieve effective "no-op" operations without driving attention logits toward negative infinity. Our analysis suggests that attention sinks do not inherently cause massive activations. Rather, models prone to outlier formation tend to the negative infinity strategy as a computational solution for implementing "no-op" operations within training dynamics that favor concentrated channel activations.

6 Conclusion

We present the Outlier-Safe Pre-Training (OSP) framework, which prevents emergence of activation outliers during LLM training by replacing Adam with the Muon optimizer, adopting Single-Scale RMSNorm, and incorporating learnable embedding projections. Our approach achieves production-scale efficiency by training a 1.4B model on 1 trillion tokens with only 2% overhead compared to Adam, while fundamentally improving quantization robustness. The resulting model maintains strong performance under aggressive 4-bit quantization where comparable models fail catastrophically. By preventing outliers instead of mitigating them post-hoc, OSP enables robust low-bit deployment without architectural modifications at inference time or costly quantization-aware training. Our publicly released model provides the first demonstration that outlier-free training is both feasible and practical at scale, opening new possibilities for efficient LLM deployment.

Limitations

Our study focuses primarily on Muon without extensive comparisons to other second-order methods like Shampoo or SOAP. This limitation stems from

practical constraints: TPU compilation time for training pipelines often exceeds 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.

Acknowledgments

This work was supported in part by the National Research Foundation of Korea [NRF-2023R1A2C3004176, RS-2023-00262002], the Ministry of SMEs and Startups [RS-2024-00523644], the Ministry of Health & Welfare, Republic of Korea [HR20C002103], the ICT Creative Consilience program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the MSIT [IITP-2025-RS-2020-II201819], and Cloud TPUs from Google's TPU Research Cloud (TRC).

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.

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*.

Federico Barbero, Alvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Michael Bronstein, Petar Veličković, and Razvan Pascanu. 2025. Why do llms attend to the first token? *arXiv preprint arXiv:2504.02732*.

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.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. 2023. Quantizable transformers: Removing outliers by helping attention heads do nothing. In *Advances in Neural Information Processing Systems*, volume 36, pages 75067–75096. Curran Associates, Inc.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. JAX: composable transformations of Python+NumPy programs.

Diego Caples and Rob Neuhaus. 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.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. 2025. When attention sink emerges in language models: An empirical view. In *The Thirteenth International Conference on Learning Representations*.

Tianyu Guo, Druv Pai, Yu Bai, Jiantao Jiao, Michael I Jordan, and Song Mei. 2024. Active-dormant attention heads: Mechanistically demystifying extreme-token phenomena in llms. *arXiv preprint arXiv:2410.13835*.

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.

Sehoon Kim, Coleman Richard Charles Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. 2024. SqueezeLLM: Dense-and-sparse quantization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 23901–23923. PMLR.

Olga Kovaleva, Saurabh Kulshreshtha, Anna Rogers, and Anna Rumshisky. 2021. BERT busters: Outlier dimensions that disrupt transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3392–3405, Online. Association for Computational Linguistics.

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.

Ruikang Liu, Haoli Bai, Haokun Lin, Yuening Li, Han Gao, Zhengzhuo Xu, Lu Hou, Jun Yao, and Chun Yuan. 2024a. IntactKV: Improving large language model quantization by keeping pivot tokens intact. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7716–7741, Bangkok, Thailand. Association for Computational Linguistics.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2024b. 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. 2024c. 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.

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*.

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.

Seungwoo Son, Wonpyo Park, Woohyun Han, Kyuyeun Kim, and Jaeho Lee. 2024. Prefixing attention sinks can mitigate activation outliers for large language model quantization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2242–2252, Miami, Florida, USA. Association for Computational Linguistics.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. 2024. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*.

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.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.

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.

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*.

A Appendix

A.1 Additional Training Details

For training data, we adopt a corpus composition similar to Allal et al. (2025), leveraging carefully validated mixture of high-quality datasets. This corpus comprises FineWeb-Edu (Lozhkov et al., 2024), FineMath (Allal et al., 2025), Cosmopedia (Ben Allal et al., 2024), and Python codes sampled from the StarCoder (Li et al., 2023) training set. The selection of this data mixture enables direct comparison with existing benchmarks while ensuring robust evaluation across various downstream tasks.

Our training infrastructure utilizes a TPU v4-512 Pod Slice, enabling efficient distributed training at scale. We conduct comparative experiments between the standard Adam optimizer with a learning rate of 5×10^{-3} and the Muon optimizer with a learning rate of 5×10^{-4}. The training configuration maintains a batch size of 4 million tokens with a sequence length of 2048 tokens, applying weight decay of 0.01 across all experiments. We adopt trapezoidal learning rate scheduling (Hägele et al., 2024; Wen et al., 2024), wherein the learning rate increases linearly from zero to its maximum value over the first 5 billion tokens, maintains this peak throughout the majority of training, and subsequently decays to zero during the final 20% steps.

To achieve optimal training throughput, we implement Fully-Sharded Data-Parallel (FSDP) (Xu et al., 2020) with parameters distributed across 16 accelerator cores. For Muon optimization, we develop a distributed variant that partitions gradients across 8 dedicated optimizer-parallel ranks, where Newton-Schulz iterations are performed independently on each rank. This parallelization strategy enables efficient orthogonalization of large gradient matrices without communication bottlenecks.

A.2 Comprehensive Benchmark Results for Open-Source LLMs

Table 5 presents the performance across 10 benchmarks without quantization. In particular, the model trained under our framework achieves comparable performance to the open-source models trained with Adam optimizer, confirming the successful application to trillion-token scale training.

A.3 Training Dynamics Over 1T Token Scale

Figure 7 illustrates the evolution of training loss and excess kurtosis throughout the one-trillion token training process, mirroring our ablation study

Model	Params.	Tokens	ARC	CSQA	GSM8K	HS	MMLU	OBQA	PIQA	SIQA	TQA	WG	Avg.
Pythia	1.4B	0.3T	41.3	35.4	2.4	50.8	31.3	34.6	71.1	43.5	9.2	55.2	37.5
TinyLlama	1.1B	2T	36.5	25.4	1.7	54.0	32.6	23.0	70.3	41.3	23.5	50.0	35.8
OPT	1.3B	0.3T	39.3	40.0	0.9	52.2	29.6	35.8	71.0	42.3	11.1	53.3	37.6
OLMo	1.2B	3T	44.2	40.4	1.7	60.4	31.9	37.8	75.2	44.1	17.6	53.4	40.7
MobileLLAMA	1.4B	1.3T	42.7	37.0	2.0	54.2	31.8	34.4	73.3	43.0	24.5	55.4	39.8
Qwen 1.5	1.8B	2.4T	46.9	32.9	34.2	59.5	33.1	37.2	74.3	44.5	18.8	57.9	43.9
Qwen 2	1.5B	7T	48.2	31.0	58.1	63.9	37.4	36.8	75.4	44.2	24.0	59.2	47.8
Qwen 2.5	1.5B	–	58.8	34.3	**61.6**	66.5	40.3	39.6	75.7	**44.9**	20.6	59.4	**50.2**
LLAMA 3.2	1.2B	–	49.2	41.1	6.0	61.3	36.3	39.0	74.9	43.5	20.7	58.1	43.0
Stable LM 2	1.6B	2T	53.5	34.6	19.3	66.7	36.0	37.0	76.8	43.5	**35.6**	59.2	46.2
SmolLM	1.7B	1T	59.7	38.0	6.8	63.0	39.4	**42.8**	76.0	44.1	25.8	54.6	45.0
SmolLM 2	1.7B	11T	**60.4**	**43.6**	32.6	**68.7**	**41.3**	42.4	**77.6**	43.4	27.1	**60.1**	49.7
From Scratch													
Adam	1.4B	1T	59.5	40.6	14.5	64.0	39.5	41.0	76.1	43.6	23.9	56.6	45.9
Muon (**OSP**)	1.4B	1T	57.5	37.6	10.5	61.3	38.5	40.4	75.5	44.4	22.4	55.8	44.4

Table 5: Performance evaluation of 12 open-source large language models and our two models trained from scratch, assessed without quantization across 10 benchmark tasks. Model parameters (**Params.**) represent total trainable parameters, while **Tokens** indicate training dataset size. Evaluation benchmarks include CommonsenseQA (CSQA), HellaSwag (HS), OpenBookQA (OBQA), TriviaQA (TQA), and WinoGrande (WG), among others. Results provide comprehensive baseline performance metrics across diverse reasoning and knowledge tasks, demonstrating the comparative performance of our trained models against established baselines.

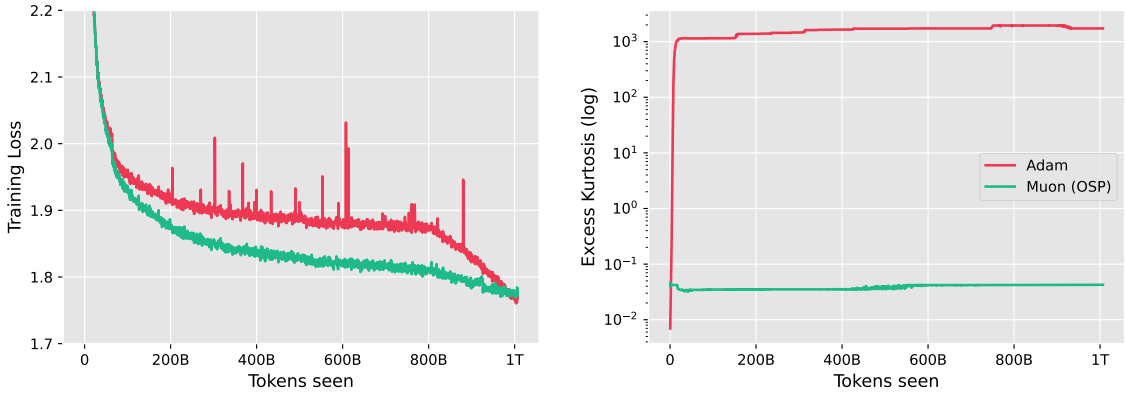


Figure 7: Training dynamics over 1 trillion tokens demonstrating production-scale viability of our framework. The loss (left) and excess kurtosis evolution (right) compare Adam baseline against complete OSP implementation. Results confirm that the OSP maintains consistently low kurtosis values throughout extended training, validating the framework's effectiveness at production scale while achieving competitive convergence characteristics.

methodology. To ensure fair comparison, we trained a standard Adam-based model under identical conditions.

A.4 Detailed Weight and Activation Distributions

For a comprehensive view of activation and weight distributions, we provide detailed histograms in Figures 8, 9, 10, and 11.

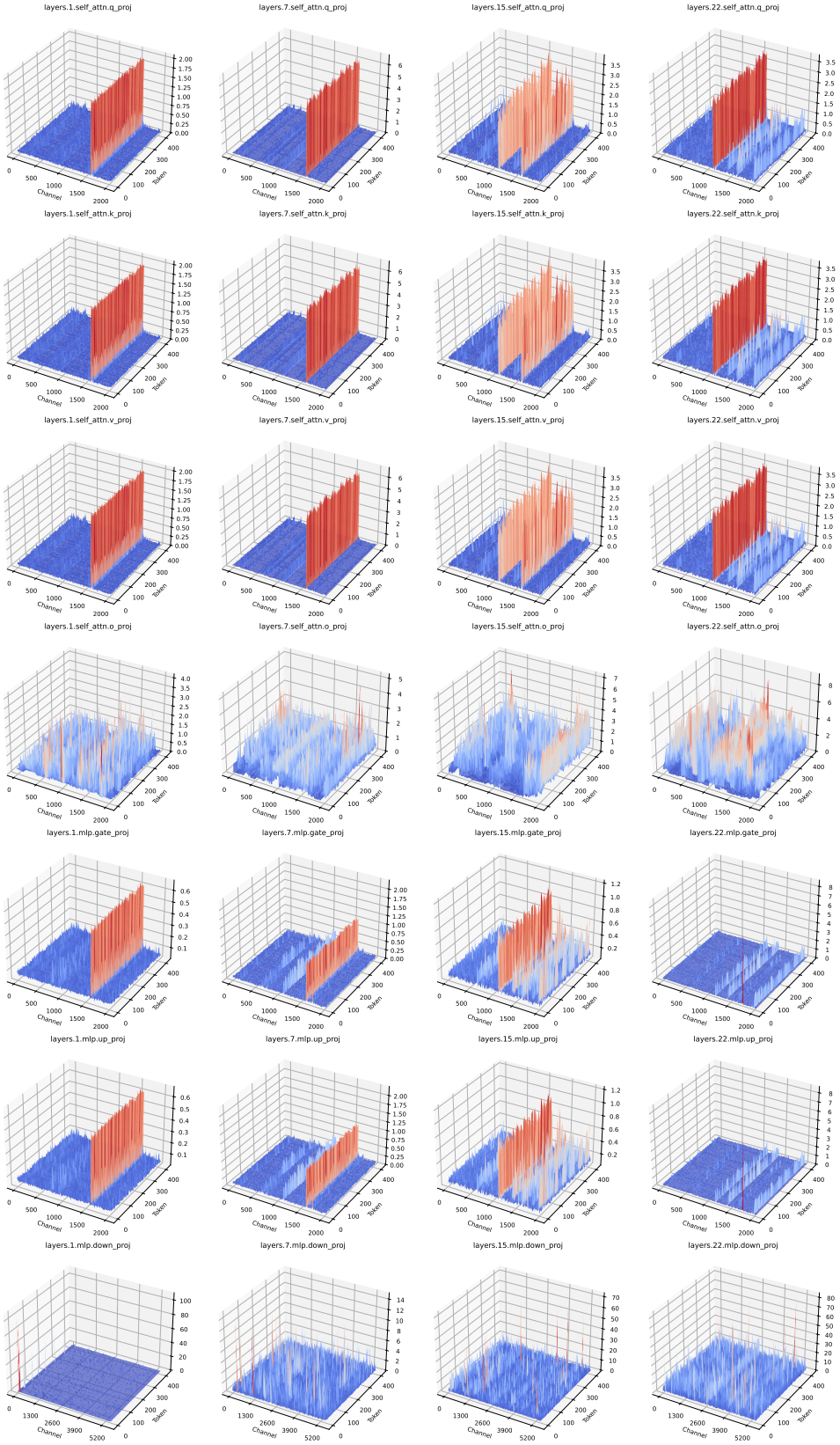


Figure 8: Activation distribution visualization of models trained with Adam optimizer across 1 trillion training tokens. The histograms display input activation distributions to Multi-Head Self-Attention (MHSA) and Feed-Forward Network (FFN) layers at four transformer block depths: 1st, 7th, 15th, and 22nd layers. Results demonstrate the evolution of activation patterns across network depth and illustrate the characteristic input distribution behavior produced by standard Adam optimization in both attention and feed-forward layers.

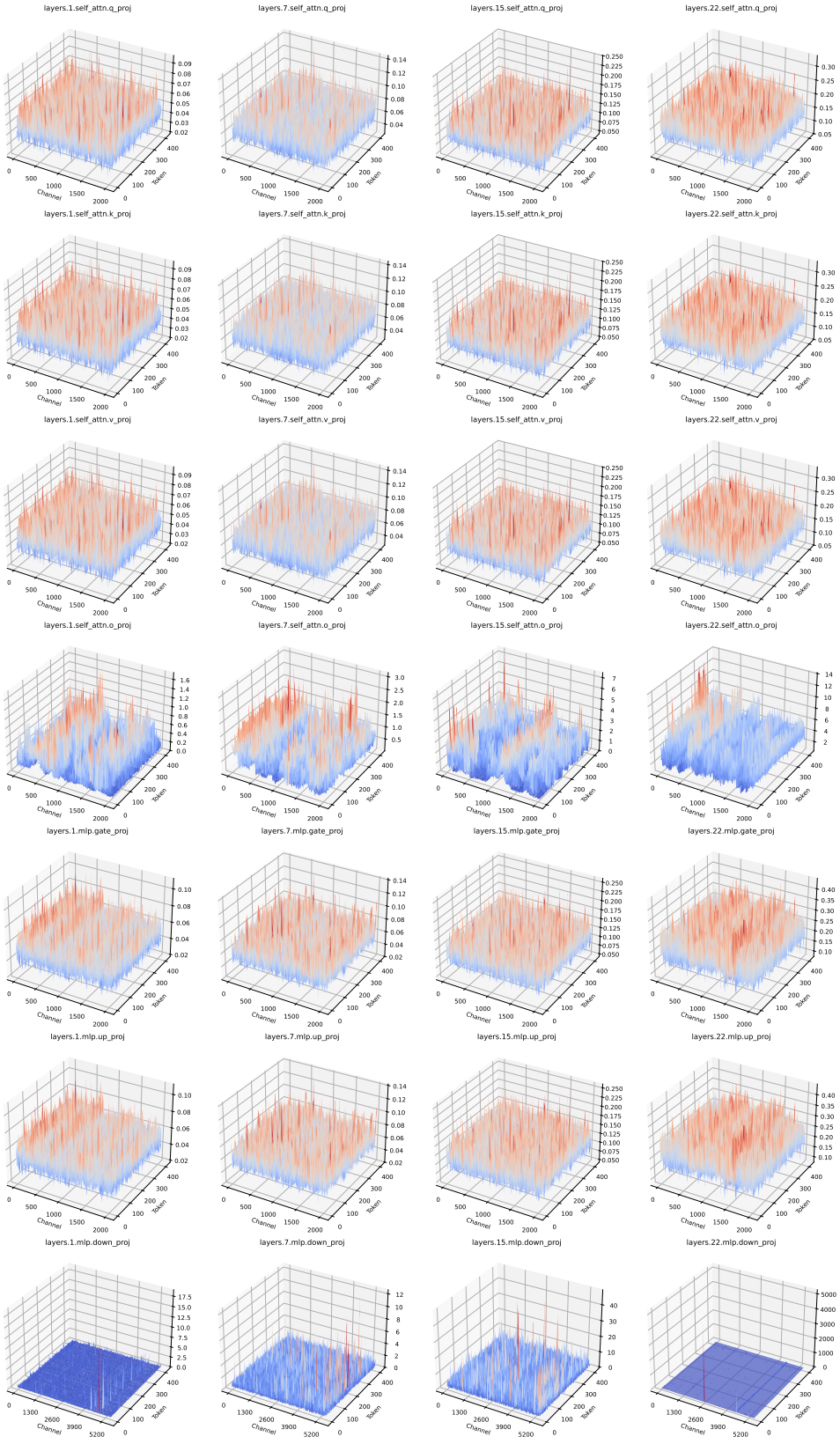


Figure 9: Activation distribution visualization of models trained with **OSP** across 1 trillion training tokens. The histograms display input activation distributions to Multi-Head Self-Attention (MHSA) and Feed-Forward Network (FFN) layers at four transformer block depths: 1st, 7th, 15th, and 22nd layers. Results demonstrate the evolution of activation patterns across network depth and illustrate the distinctive input distribution characteristics achieved through the **OSP** framework in both attention and feed-forward layers.

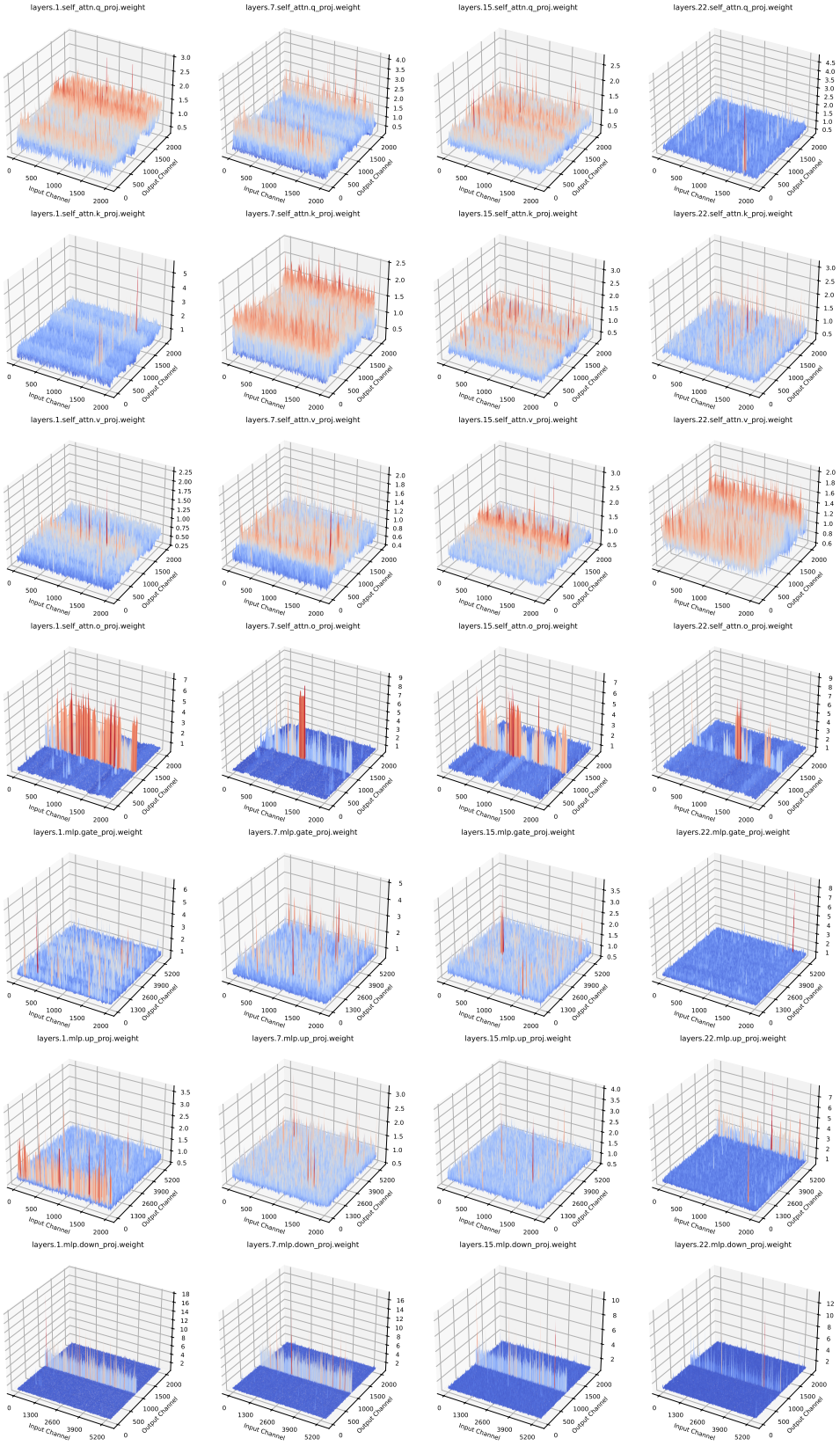


Figure 10: Weight distribution visualization of models trained with Adam optimizer across 1 trillion training tokens. The histograms display weight distributions within Multi-Head Self-Attention (MHSA) and Feed-Forward Network (FFN) layers at four transformer block depths: 1st, 7th, 15th, and 22nd layers. Results illustrate the evolution of weight distributions across network depth and demonstrate the characteristic patterns produced by standard Adam optimization in both attention and feed-forward layers.

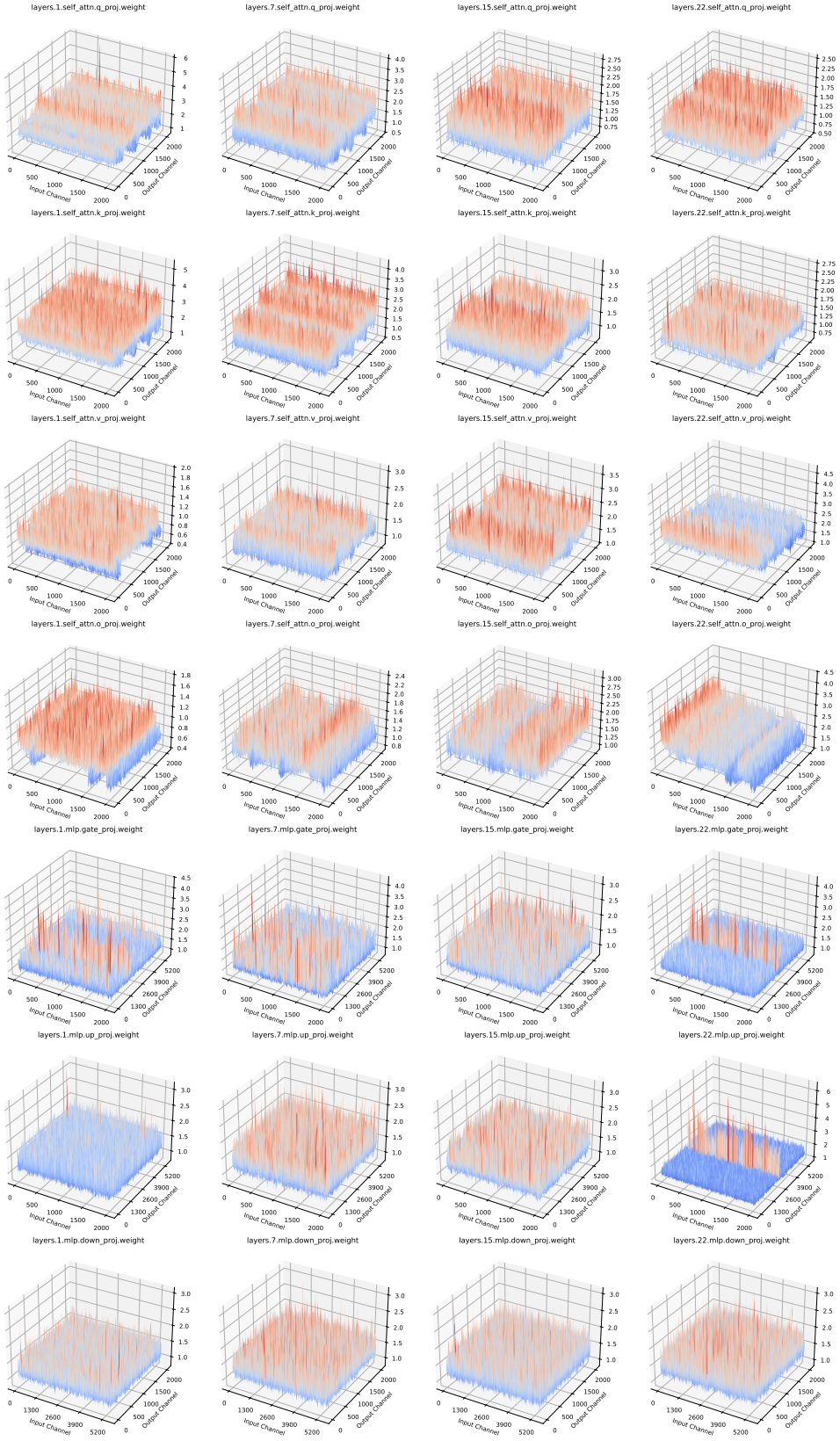


Figure 11: Weight distribution visualization of model trained with **OSP** across 1 trillion training tokens. The histograms display weight distributions within Multi-Head Self-Attention (MHSA) and Feed-Forward Network (FFN) layers at four transformer block depths: 1st, 7th, 15th, and 22nd layers. Results illustrate the evolution of weight distributions across network depth and demonstrate the characteristic patterns induced by the **OSP** framework in both attention and feed-forward layers.