AMXFP4: Taming Activation Outliers with Asymmetric Microscaling Floating-Point for 4-bit LLM Inference

Anonymous ACL submission

Abstract

As large language models (LLMs) grow in parameter size and context length, computation precision has been reduced from 16-bit to 4-bit to improve inference efficiency. However, this reduction causes accuracy degradation due to activation outliers. Recent rotation-based INT4 quantization attempts to address this through rotation matrix calibration, but they require hours of overhead per model deployment and leave significant computation unquantized in long-context scenarios. Microscaling (MX) floating-point (FP) formats offer fine-grained representation with a shared scale, enabling fully quantized matrix multiplications through direct casting without calibration. However, existing research shows unsatisfactory empirical results for MXFP4 inference, and the robustness of MX formats remains largely unexplored.

In this work, we uncover the fundamental tradeoffs of the MX format: while it effectively suppresses activation outliers, it does so at the cost of increased group-wise asymmetry. To address this, we propose an asymmetric MX format for a 4-bit floating point (AMXFP4), which employs asymmetric shared scales to handle both outliers and group-wise asymmetry without requiring calibration. Our custom compute-engine implementation shows that the AMXFP4-based Multiply-Accumulate (MAC) design adds marginal resource overhead while delivering substantial accuracy improvements. Extensive experiments across benchmarks demonstrate that AMXFP4 outperforms MXFP4 in visual question answering (VQA) by 3% and surpasses rotation-based techniques on CSQA by 1.6%. Additionally, AMXFP4 shows superior performance compared to the recently deployed commercial MXFP4 format.

1 Introduction

Multi-modal Large Language Models (LLMs) are widely used in advanced natural language processing tasks, including chatbots, long-document question-answering, and visual graph interpretation (Bai et al., 2023; Liu et al., 2023a). To enhance their capabilities, LLMs have been significantly scaled in both parameter size and context length (Chung et al., 2022; Chowdhery et al., 2022). For example, LLaMA3 (AI@Meta, 2024) now features 405 billion parameters and supports context lengths of up to 128K tokens. As shown in Fig. 1(a), this scaling results in peta-FLOP-level computational demands during the prefill phase, where the model processes user context before inference.

Leading computing platforms have focused on bit-precision scaling to meet the computational demands of LLMs (Andersch et al., 2022; Nvidia, 2024; AzureAI, 2024). Reducing operand bitwidths improves area and energy efficiency in arithmetic operations (Horowitz, 2014), enabling higher computation density in accelerators. As shown in Fig. 1(b), NVIDIA's Tensor Cores double computation speed by lowering multiply-accumulate (MAC) precision from FP16 to FP8 (Andersch et al., 2022) and from INT8 to INT4 (Nvidia, 2020).

Recent research explores activation and weight quantization to improve LLM inference efficiency by leveraging hardware precision scaling. However, quantizing both weights and activations to INT4 often degrades accuracy due to activation outliers (Dettmers et al., 2022; Xiao et al., 2022). Rotation-based transformations mitigate this by making activations more quantization-friendly (Ashkboos et al., 2024; Liu et al., 2024b), with approaches like QuaRot (Ashkboos et al., 2024) significantly reducing LLM perplexity in INT4 inference (Fig. 1(c)). Despite these benefits, rotation-based methods require extensive calibration, leading to overfitting risks (Lee et al., 2023; Lin et al., 2023) (cf. Table 2), and are impractical for user-specific model deployments that demand frequent recalibration (Bang et al., 2024). Additionally, they leave Softmax outputs unquantized, forcing FP16 multiplications with value vectors,

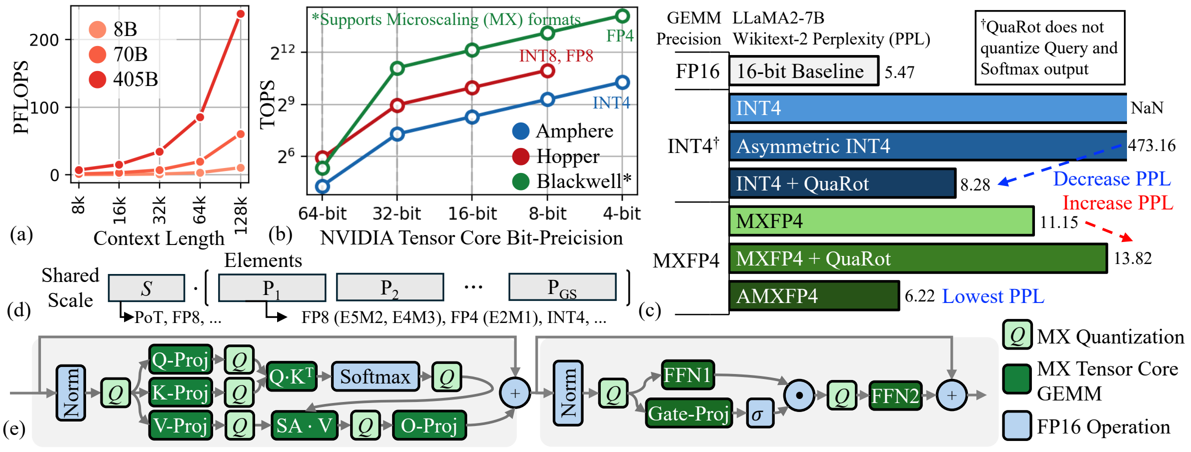


Figure 1: (a) FLOPS across context length and model sizes. (b) Precision scaling in NVIDIA Tensor Cores. (d) Impact of bit-precision and data rotation on perplexity. (d) MX format. (e) LLM inference with MX Tensor Core.

which account for 41% of total FLOPs in 8B LLMs with 128K-token inputs (cf. Fig. 8).

An alternative approach to quantization introduces reduced-precision formats that enable calibration-free data-type conversion (i.e., direct casting). For instance, the latest NVIDIA Tensor Core (Nvidia, 2024) supports the microscaling (MX) format, introduced by the Open Compute Project (OCP) (Rouhani et al., 2023a), which groups low-precision elements under a shared scale to mitigate dynamic range limitations (Fig. 1(b), (d)). As shown in Fig. 1(c), (e), MXFP4 achieves full matrix quantization with minimal perplexity degradation compared to INT4, without requiring data rotation. This is due to its fine-grained quantization, which enhances value representation precision. However, MXFP4 still lags behind the 16-bit baseline in perplexity and performs worse when combined with data rotation, and the root causes of this destructive interaction are mainly unexplored.

This work uncovers a key trade-off in the MX format: while it effectively suppresses activation outliers, it increases group-wise asymmetry. Grouping activation tensors into small micro-scaled units mitigates outliers, similar to rotation methods, but enables direct-cast inference. However, this grouping amplifies data asymmetry, necessitating an asymmetric numerical representation. To address this, we propose AMXFP4, a microscaling floating-point format designed for robust 4-bit LLM inference, which effectively handles activation outliers through micro-scaled asymmetric data representation. By employing an FP8 shared scale for both weights and activations, AMXFP4 achieves quantization error rates close to ideal Lloyd-Max quantization. To validate its broad applicability, we evaluate AMXFP4 across multi-turn conversation, long-context inference, and visual question-answering (VQA) tasks on decoder-only LLMs, vision-language models, and an encoder-decoder model. Results show that AMXFP4 enables calibration-free, direct-cast 4-bit inference, outperforming MXFP4 and leading rotation-based quantization methods. Additionally, AMXFP4 performs better than the recently deployed commercial MXFP4 format (NVFP4) (NVIDIA, 2024).

Our contributions can be summarized as follows:

- We examine the MXFP4 format, finding that microscaling effectively reduces activation outliers without calibration but introduces asymmetry, necessitating asymmetric numerical representation.
- We propose AMXFP4, a novel format that combines FP4 elements with shared asymmetric FP8 scales, significantly suppressing quantization error.
- We evaluate AMXFP4 across diverse applications, including multi-turn conversation, long-context inference, and VQA, across multiple model types, demonstrating consistently superior accuracy to MXFP4.

2 Background and Related Work

2.1 Bit-Precision Scaling for Accelerators

Reduced-precision formats are vital for enhancing scalability and computational efficiency in deep learning accelerators, conserving area and energy in direct proportion to bit-width reduction (Horowitz, 2014). This scaling enables higher floating-point operations per second (FLOPS) with lower power usage, thereby increasing accelerator throughput. For instance, NVIDIA's Tensor Cores have progressed from FP16 in Volta (Nvidia, 2017) to FP8 in Hopper (Andersch et al., 2022) and FP4 in Blackwell (Nvidia, 2024), boosting computational speeds from 112 tera to 20 peta FLOPS, as shown in Fig. 1(b). Similar advancements by other computing platform companies in scaling precision from 16-bit to 4-bit are crucial for managing the growing complexity of LLMs (AMD, 2024; AzureAI, 2024).

Recently, the microscaling (MX) format (Rouhani et al., 2023a; Darvish Rouhani et al., 2023; Rouhani et al., 2023b) has been developed from Block Floating Point (BFP) (Drumond et al., 2018; Darvish Rouhani et al., 2020) by incorporating a shared scale across a block of reduced-precision elements, thus mitigating quantization error due to limited dynamic range. While the original BFP format allows flexibility in design parameters-exponent (E) and mantissa (M) for the element (P_i) and the shared scale (S), and the group size (GS), MX prescribes specific *MX-compliant* configurations (cf. Table 6): MXFP8 (P_i:$E4M3$, S=$E8$, GS:32) and MXFP4 (P_i:$E2M1$, S=$E8$, GS:32), as shown in Fig. 1(d).

However, MXFP4's robustness for LLM inference remains uncertain, with significant performance degradation in 4-bit inference due to activation quantization (Rouhani et al., 2023b). Moreover, MXFP4 lacks validation on practical tasks such as multi-turn chatbot interactions, raising concerns about its real-world applicability. While MXFP4 models generate coherent answers, they often yield unhelpful responses, consistent with findings that quantization can impair conversational quality (Lee et al., 2024) (e.g., Fig. 11). These results underscore the need for new data formats to enable robust 4-bit inference.

2.2 Quantizing LLM's Activation and Weight

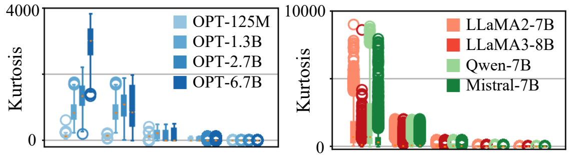
Recent research highlights the difficulty quantifying LLM activations due to outliers extending the activation dynamic range, leading to increased quantization error (Xiao et al., 2022; Ashkboos et al., 2024). Prior studies propose rescaling weights and activations to reshape their distributions for better quantization compatibility while preserving mathematical equivalence (Xiao et al., 2022; Shao et al., 2024; Lee et al., 2023). However, such methods often experience accuracy degradation in 4-bit inference (Lin et al., 2024). Data rotation strategies, including QuaRot (Ashkboos et al., 2024) and SpinQuant (Liu et al., 2024b), use orthogonal matrices to redistribute concentrated channel information (represented as R in Fig. 8(a)). QuaRot applies a randomized Hadamard matrix, while SpinQuant uses learned rotation matrices. DuQuant further enhances this approach by combining per-channel permutation and rotation, achieving state-of-the-art accuracy in 4-bit inference (Lin et al., 2024).

However, these rotation-based methods exclude quantization for the Softmax output, leaving matrix multiplications in the self-attention calculation to be computed in FP16. Since self-attention computation scales quadratically with context length during the prefill phase, the partial quantization of rotation methods significantly reduces overall computational efficiency in long-context inference. Additionally, these techniques require extensive calibration, such as GPTQ (Frantar et al., 2022) or training rotation matrices, to improve model accuracy. However, calibration introduces the risk of overfitting, as models may become overly tailored to the calibration dataset, limiting their adaptability across broader applications (Table 2). Further discussions on limitations of calibration-based methods are provided in the Appendix A.

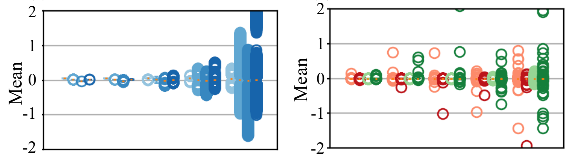
These challenges highlight the need for a generalizable quantization approach that minimizes calibration dependence and applies uniformly across computations. Although MXFP4, a previously explored reduced-precision format, applies to all matrix multiplication without calibration, it compromises model accuracy. This work analyzes MXFP4's strengths and limitations, and proposes AMXFP4, a superior 4-bit format that enables direct-casting with improved model accuracy.

3 Microscaling for Taming Outliers

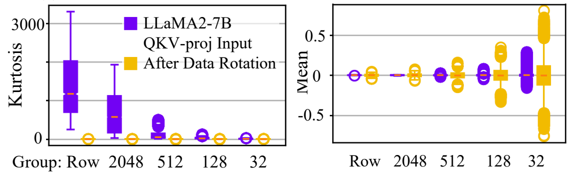
We systematically analyze activation outliers across various LLMs using representative statistical measures—kurtosis and mean—to understand the effects of microscaling (i.e., reducing a quantization group to 32 elements). Kurtosis, the fourth standardized moment, is commonly used to assess the prevalence of outliers (Liu et al., 2024b), while the mean reflects asymmetry within each group. We use box plots of kurtosis and mean to examine the



(a) Kurtosis of FFN1 Activations (b) Kurtosis of FFN2 Activations



(c) Mean of FFN1 Activations (d) Mean of FFN2 Activations



(e) Kurtosis after Data Rotation (f) Mean after Data Rotation

Figure 2: Data characteristics based on (a-d) types of LLM, layer and (e-f) data rotation across group sizes.

value distribution within groups, which are subject to quantization using a shared scale.

3.1 Analysis of LLM's Activation Outliers

Fig. 2(a) and (b) present the kurtosis box plots for the OPT (Zhang et al., 2022) and LLaMA-like models (LLaMA, Qwen, Mistral (Touvron et al., 2023; AI@Meta, 2024; Bai et al., 2023; Jiang et al., 2023)). In cases of row-wise grouping (typically $GS \gg 1024$), the OPT models exhibit high kurtosis in FFN1 activations, indicating many outliers that challenge quantization. Additionally, outlier prevalence increases with model size, aligning with previous findings that larger models are more affected by quantization (Dettmers et al., 2022). Conversely, LLaMA-like models use the Gated Linear Unit (GLU) activation function, involving extra matrix multiplication; thus, data passing through FFN1 undergoes element-wise multiplication before FFN2, further amplifying outliers—a phenomenon observed in recent studies (Yang et al., 2024; Fishman et al., 2024). Notably, *outlier dominance is reduced as group size decreases in both model types*. At GS=32, kurtosis nearly disappears, suggesting the activation dynamic range within groups becomes more suitable for quantization. This observation helps explain the preliminary success of MXFP8 in direct-casting for selected LLMs (Rouhani et al., 2023b), but it does not explain the disappointing performance of MXFP4.

To assess the trade-offs in the MX format's handling of outliers, we examine the box plots of group means, which reflect distribution asymmetry. Fig. 2(c) and (d) show the mean values for FFN1 and FFN2 input activations as group size decreases from an entire row to 32. Notably, with large group sizes, group means center around zero, but as group size decreases, the means scatter significantly. This scattering indicates that the symmetric data representation typically used in the MX format is suboptimal for microscaled activation quantization. In other words, *microscaling addresses activation outliers at the cost of data symmetry*. Thus, simply reducing group size (as in the MX format) may not adequately minimize quantization error; instead, an asymmetric data representation becomes essential.

3.2 Data Rotation vs. Microscaling

We then examine how data rotation reduces outliers alongside microscaling and assess its effectiveness as group size decreases. Fig. 2(e) shows the kurtosis before and after applying data rotation using a random Hadamard transform (Ashkboos et al., 2024) across decreasing group sizes. When the group size spans an entire row, activation rotation substantially lowers kurtosis, demonstrating its efficacy in 4-bit activation quantization. However, as group size decreases, the original activation's kurtosis also drops, reaching levels comparable to those achieved with rotation. Thus, the benefit of data rotation in outlier reduction diminishes with smaller group sizes.

On the other hand, Fig. 2(f) shows the group means of the activation before and after applying data rotation. As with the original activation, the group means scatter more as group sizes decrease, but this scattering is even more pronounced with rotated activations. This indicates that rotation introduces an additional asymmetry in group distributions, which complicates quantization with MXFP4's symmetric representation (cf. Table 1). In other words, data rotation and microscaling lack synergy, as both focus on outlier suppression without addressing asymmetry. Thus, a microscaling data format that effectively handles group distribution asymmetry presents a compelling alternative.

3.3 Multi-modal LLM's Activation Outlier

To further understand activation outliers under microscaling in multi-modal LLMs, we examine the popular vision-language model LLaVA (Liu et al., 2023a). LLaVA combines a visual encoder and a language model backbone: an image is processed by a vision transformer-based encoder (Dosovitskiy et al., 2021) to generate vision tokens, which

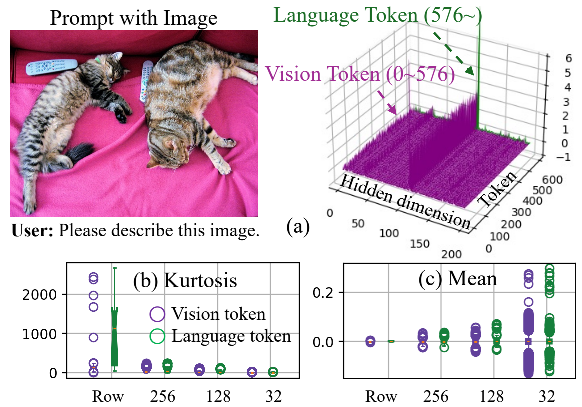


Figure 3: Characteristic of VLM activation outliers across group sizes (LLaVA-v1.6-Vicuna-7B Layer 1 QKV-Proj).

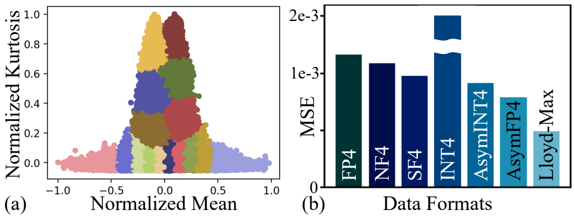


Figure 4: Cluster-wise Lloyd-Max quantization and quantization error across data formats (LLaMA2-7B layer 5 QKV-Proj input activation).

are then input to the language model along with language tokens from the user prompt.

As shown in Fig. 3(a), both vision and language tokens exhibit outliers within the same hidden dimension of the activation, though their distributions differ. Language tokens typically concentrate around larger magnitudes, while only some vision tokens reach high magnitudes, a trend observed consistently across layers. In Fig. 3(c), these differences result in varying kurtosis distributions for row-wise group quantization: language tokens have clustered outliers, while vision tokens show a sparser outlier distribution. However, this distinction fades as group size decreases, illustrating the effectiveness of microscaling in suppressing outliers. Similar to LLMs, LLaVA's group means scatter as group size decreases, indicating increased asymmetry in exchange for outlier suppression. This suggests microscaling could better handle diverse outlier patterns from vision and language tokens if designed to support asymmetric data representation.

4 Asymmetric Microscaling Format

The findings from Sec. 3 motivate the development of a new microscaling format that inherently supports asymmetric data representation. In this section, we explore the design space of the microscaling data format (P_i and S) alongside considerations for asymmetric quantization schemes.

4.1 Selecting Element-Wise Data Format

We first examine the design space of the element-wise data format P_i. To evaluate the benefits of asymmetric formats, we compare the mean-square error (MSE) on activation samples from LLaMA2-7B's QKV-Proj at layer 5 across four symmetric formats (INT4, FP4, NF4 (Dettmers et al., 2023), SF4 (Dotzel et al., 2024)) with two asymmetric formats:

- **Asymmetric INT (AsymINT):** INT quantization applies asymmetry through a zero-point, shifting the data range from zero-centered to span between the minimum and maximum values (Dettmers et al., 2022).
- **Asymmetric FP (AsymFP):** FP quantization introduces asymmetry by applying separate scales to positive and negative values due to FP's inherently zero-centered representation (Zhang et al., 2024b).

We compare the MSE of each format on activation samples from LLaMA2-7B's QKV-Proj at layer 5. Fig. 4(a) characterizes these activations by group mean (x-axis) and kurtosis (y-axis). As a reference, we cluster groups based on mean and kurtosis similarity, then apply the Lloyd-Max algorithm (Lloyd, 1982) for near-optimal quantization (100 iterations, with 16 clusters, as further clustering yields no additional MSE reduction).

Fig. 4(b) presents the MSE of various element-wise data formats. Compared to Lloyd-Max quantization (used as a reference), all symmetric data formats show a significant MSE increase, with INT4 experiencing the most notable degradation. In contrast, AsymINT4 and AsymFP4 achieve lower MSE, with AsymFP showing MSE closest to Lloyd-Max (a consistent trend across models and layers). This finding supports the selection of AsymFP4 as the element-wise format, further validated empirically in Table 1.

4.2 Selecting Shared-Scale with Asymmetry

With AsymFP4 selected as the preferred element-wise data representation, its original design for

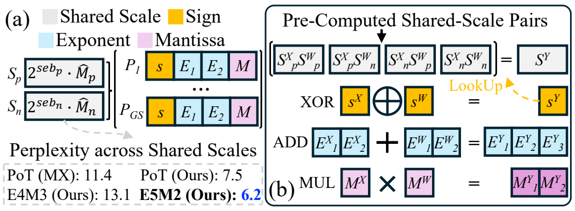


Figure 5: (a) Illustration of AMXFP4 and LLaMA2-7B Wikitext-2 perplexity across shared scale types. (b) Multiplication between two AMXFP4 datas.

weight-only quantization (Zhang et al., 2024b) requires high-precision dequantization before multiplication with activations. To integrate AsymFP into reduced-precision GEMM, we redefine AsymFP such that an exponent-bit-shifted mantissa represents a value, which is then scaled by a shared factor with sign-dependent polarity:

$$x_q = \begin{cases} (-1)^s \cdot 2^{E+eb} \cdot M \cdot (2^{seb_p} \cdot \hat{M}_p) \text{ if } s = 0, \\ (-1)^s \cdot 2^{E+eb} \cdot M \cdot (2^{seb_n} \cdot \hat{M}_n) \text{ if } s = 1, \end{cases} \tag{1}$$

where s, E, eb, and M represent an element's sign, exponent, exponent bias, and mantissa, respectively. As described in Fig. 5(a), the terms $2^{seb_p} \cdot \hat{M}_p$ and $2^{seb_n} \cdot \hat{M}_n$ represents the positive and negative scales shared within a quantization group.

PoT. When $\hat{M}_p = \hat{M}_n = 1$, the dynamic range for positive and negative values can be adjusted by modifying the exponent. However, we observe that MXFP4's PoT frequently triggers max clamping in small group sizes, causing significant performance degradation. To address this, we propose an advanced PoT that mitigates max clamping by modifying the PoT decision rule (see Appendix B.2 for details). As shown in Fig. 5(a), the proposed PoT shared scale reduces LLaMA2 perplexity by approximately 4.

FP8. Although proposed PoT scale prevents clamping errors, its limited resolution still causes accuracy loss. To mitigate this issue, we propose using FP8 scales to leverage additional mantissa bits for finer rounding. However, as shown in Fig. 5(a), a 4-bit exponent results in a narrower dynamic range, which in turn increases perplexity compared to PoT. Therefore, we select FP8 with a 5-bit exponent ($E5M2$) as the shared scale, as these scales largely mitigate accuracy degradation caused by the limited resolution and narrower dynamic range (see Table 15 for ablation studies).

4.3 Asymmetric Microscaling Floating-Point

Based on our exploration of the MX design space, we propose AMXFP4 (asymmetric microscaling 4-bit floating-point), which utilizes asymmetric FP8 shared scales. During multiplication, the shared scale is selected based on the signs of the two numbers. As shown in Fig. 5(b), this overhead remains minimal because the mantissa of the shared scale is only 2 bits, and the scale is computed once and shared within a group. To evaluate AMXFP4 on real hardware, we implement an AMXFP4 MAC unit via hardware synthesis by modifying the existing MX MAC unit (Darvish Rouhani et al., 2023). Our evaluation shows that AMXFP4 incurs only about a 10% overhead compared to MXFP4 (details are in Appendix B.3).

5 Experiments

In this section, we compare AMXFP4 with other formats and rotation-based methods. Unless otherwise specified, all experiments use the proposed FP8 shared scale across all formats (including INT4, MXFP4, and AMXFP4) for a fair comparison and quantize input operands for all decoder-layer matrix multiplications. Further details on quantization settings and benchmark descriptions are provided in Appendix C.

5.1 Impact of Microscaling and Data Rotation

Microscaling vs. Data Rotation. We empirically validate the findings discussed in Sec. 3.2, confirming that data rotation effectively mitigates activation outliers in configurations with large group sizes but has limited compatibility with microscaling. Table 1 presents the impact of data rotation (randomized Hadamard transform) on Wikitext-2 (Merity et al., 2016) perplexity, with group sizes ranging from an entire row to 32. When the group size spans an entire row, data rotation provides the best solution for MXFP4, outperforming asymmetric data representations. However, as the group size decreases, data rotation increases perplexity across all models with MXFP4, whereas AMXFP4 consistently reduces perplexity, achieving a 0.6-point reduction in LLaMA3-8B. This result further supports that outlier handling becomes less effective as group size decreases.

INT4 vs. FP4. We extend our analysis to microscaling INT (MXINT) to assess whether the adverse effects of data rotation stem from FP's non-uniform data representation. Similar to MXFP4,

Group Size	Data Rotation	Data Format	LLaMA 2-7B	2-13B	3-8B
FP16 Baseline			5.47	4.88	6.14
Row	-	MXINT4	NaN	2988.82	2603.42
		AMXINT4	2045.70	364.96	1800.44
		MXFP4	475.62	99.33	85.07
		AMXFP4	44.75	33.79	40.33
	✓	MXINT4	47.55	35.32	100.95
		AMXINT4	16.60	13.94	35.90
		MXFP4	**11.88**	**10.81**	13.27
		AMXFP4	12.05	11.54	**12.13**
MX (32)	-	MXINT4	7.01	6.11	9.01
		AMXINT4	6.33	5.55	9.62
		MXFP4	6.49	5.69	8.35
		AMXFP4	**6.22**	**5.47**	**7.72**
	✓	MXINT4	7.90	6.18	9.96
		AMXINT4	6.75	5.75	8.25
		MXFP4	10.09	6.89	9.48
		AMXFP4	8.36	6.35	9.20

Table 1: Wikitext-2 perplexity results by group size with and without data rotation applied (lower is better).

LLaMA	Eval Dataset	QuaRot	QuaRot + GPTQ		SpinQuant		AMXFP4
Calib Dataset		-	PM	EE	PM	EE	-
2-7B	PM ↓	7.7	5.4	5.5	5.7	5.9	**5.3**
	EE ↓	7.9	6.3	6.2	6.8	6.3	**6.1**
3-8B	PM ↓	9.4	7.4	7.6	7.5	7.7	**6.8**
	EE ↓	12.9	10.7	10.2	10.7	10.0	**9.4**
Calibration Dataset		-	PQ	WG	PQ	WG	-
2-7B	PQ ↑	72.0	77.4	76.2	76.4	73.1	**77.8**
	WG ↑	60.1	65.3	65.9	66.4	64.0	**67.5**

PM: PubMed, EE: Enron Emails, PQ: PIQA, WG: WinoGrande

Table 2: Impact of overfitting: Calibration on different data distribution on LLaMA models.

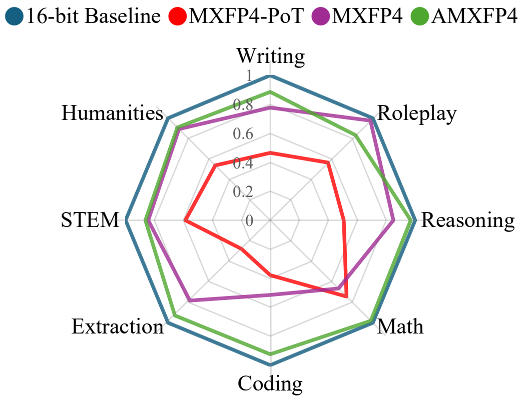


Figure 6: Normalized single score of MT-Bench (LLaMA2-Chat-7B). Absolute accuracies are in Table 13 in Appendix.

Data Format	VQA-T	DocVQA	OCRBench	ChartQA
16-bit Baseline	64.84	74.46	52.40	54.72
MXFP4-PoT	50.05	52.85	33.70	36.76
MXFP4	57.88	64.26	43.40	46.20
AMXFP4	**59.13**	**66.98**	**43.90**	**49.48**

Table 3: LLaVA1.6-7B inference results on multi-modal visual question-answering benchmarks.

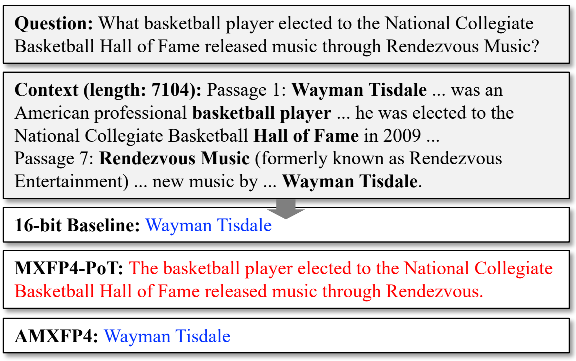
MXINT4 benefits from data rotation when the group size spans an entire row, significantly reducing perplexity compared to asymmetric representation (AMXINT4). However, at a group size of 32, data rotation tends to increase perplexity. Notably, at group size 32, AMXINT4 achieves lower perplexity than MXFP4, but AMXFP4 achieves the lowest perplexity overall. This result demonstrates that our element format selection in Sec. 4.1 effectively enhances LLM accuracy.

Robustness to Calibration Set Distributions. Table 2 examines the sensitivity of QuaRot and SpinQuant to varying calibration set distributions. Perplexity is measured on PubMed (of the U.S. National Library of Medicine, 2023) and Enron Emails (Klimt and Yang, 2004), while accuracy is measured on PIQA (Bisk et al., 2019) and WinoGrande (Sakaguchi et al., 2019), using both matched and mismatched calibration/evaluation sets. QuaRot with GPTQ and SpinQuant substantially outperform the random Hadamard rotation but tend to show better accuracy on data observed during calibration. One exception is SpinQuant, which attains strong accuracy on both PIQA and WinoGrande when calibrated on PIQA, although results vary by about 2–3% solely due to different calibration datasets. However, AMXFP4 remains unaffected by the calibration set and notably improves results and surpasses conventional calibration-based methods.

5.2 Enhancing MX Performance

In this section, we evaluate AMXFP4 against MXFP4 in practical applications, including chatbots, visual tasks, and long-document question answering. To assess our improvements over the *MX-compliant* format, we also include MXFP4 with PoT shared scale (MXFP4-PoT) from Sec 4.2 as a baseline for comparison.

Multi-Turn Chatbot Tasks. Quantization adversely affects the conversational capabilities of chatbots (Lee et al., 2024); therefore, we conduct an MT-Bench evaluation (Zheng et al., 2023) on LLaMA2-Chat-7B (Touvron et al., 2023). Fig. 6 presents the normalized scores with the 16-bit baseline score set to 1. While MXFP4 inference shows severe performance degradation across all categories, AMXFP4 demonstrates recovery of conversational abilities close to the baseline. Fig. 11 and 13 provide detailed examples, showing that while MXFP4 generates unhelpful sentences, AMXFP4 produces responses that are genuinely helpful.



Question: What basketball player elected to the National Collegiate Basketball Hall of Fame released music through Rendezvous Music?

Context (length: 7104): Passage 1: **Wayman Tisdale** ... was an American professional **basketball player** ... he was elected to the National Collegiate Basketball **Hall of Fame** in 2009 ...
Passage 7: **Rendezvous Music** (formerly known as Rendezvous Entertainment) ... new music by ... **Wayman Tisdale**.

16-bit Baseline: Wayman Tisdale

MXFP4-PoT: The basketball player elected to the National Collegiate Basketball Hall of Fame released music through Rendezvous.

AMXFP4: Wayman Tisdale

Figure 7: LongBench-E results on LLaMA2-Chat-7B.

Visual Tasks. Table 3 presents results on four multi-modal benchmarks (Zhang et al., 2024a) using LLaVA1.6-7B (Liu et al., 2023a). AMXFP4 improves MXFP4 scores by approximately 3.3 points on benchmarks such as ChartQA (Masry et al., 2022), highlighting the significant advantages of asymmetric data representation in VLMs (example is shown in Fig. 12).

Long-Context Tasks. We conduct the LongBench-E (Bai et al., 2024) evaluation to assess the effectiveness of AMXFP4 in long-context scenarios. As shown in Fig. 7, while MXFP4-PoT's generation quality significantly degrades on questions with lengthy contexts, AMXFP4 produces answers identical to the baseline. Detailed scores across 13 benchmarks, categorized by context length, are presented in Table 14. The results indicate that AMXFP4 outperforms MXFP4, achieving over a 2% accuracy improvement for context lengths exceeding 8K.

5.3 Comparison with Commercial MXFP4

Recently, NVFP4 (NVIDIA, 2024) adopts a smaller group size of 16 and employs a double-scaling strategy, which combines a tensor-wise FP32 shared scale with a group-wise FP8 (E4M3) shared scale. We evaluate whether our proposed asymmetric shared scale enhances the recently deplyed commercial MXFP4 by evaluating ANVFP4 (Asymmetric NVFP4) on Common-Sense Question Answering (CSQA) (Talmor et al., 2019) and MMLU (Hendrycks et al., 2020) benchmarks. As shown in Table 4, when GS=32, AMXFP4 and ANVFP4 surpass NVFP4 in accuracy, indicating that the asymmetric data representation offers a greater improvement than double scaling strategy. Notably, in the NVFP4 setting with GS=16, ANVFP4 increases MMLU accuracy by about 3%, which aligns with our observation that asymmetry becomes more beneficial at smaller group sizes.

GS	Data Format	MMLU Accuracy (%) ↑			CSQA Accuracy (%) ↑		
		2-7B	2-13B	3-8B	2-7B	2-13B	3-8B
16-bit Baseline		41.3	50.5	62.0	64.9	67.3	69.2
32	MXFP4-PoT	29.2	37.9	43.1	59.4	62.2	58.6
	MXFP4	33.6	42.8	49.5	61.6	65.1	62.0
	AMXFP4	**36.3**	45.0	52.8	62.0	64.9	62.2
	NVFP4	32.9	44.5	51.9	61.4	**65.0**	61.9
	ANVFP4	34.8	**45.8**	**54.0**	**62.2**	64.7	**62.9**
16	NVFP4	34.0	45.9	54.6	**62.6**	65.3	63.4
	ANVFP4	**37.3**	**47.7**	**57.1**	62.2	**66.2**	**64.9**

Table 4: MMLU and CSQA results on LLaMA models.

5.4 Ablation Studies

We extend our experiments to transformer model types (encoder-decoder models), quantization-aware training, and 3-bit quantization. Experimental details and additional results, including perplexity results and applying AMXFP4 on sparse models, are provided in Appendix D.1. Below is a summary of our ablation study findings.

Encoder-Decoder Language Model. We extend the comparison between AMXFP4 and MXFP4 to the summarization task using BART-Large (Lewis et al., 2019). As shown in Table 9, AMXFP4 exhibits a 0.9-point ROUGE-1 degradation compared to the baseline, whereas MXFP4 suffers a greater 1.3-point degradation.

Quantization-Aware Training (QAT). We investigate whether QAT can reduce the perplexity gap between MXFP4 and AMXFP4. As shown in Table 10, after applying QAT to LLaMA3-8B, AMXFP4 nearly matches the baseline perplexity, while MXFP4 still exhibits a remaining gap.

More Aggressive Quantization. We compare QuaRot and AMXFP under a 3-bit setting (W3A3) in Table 11. While QuaRot with GPTQ maintains LLaMA2-13B perplexity degradation within 1 in W4A4 (Ashkboos et al., 2024), it suffers a severe degradation exceeding 30 in W3A3. In contrast, AMXFP3 achieves a perplexity degradation of only 1.7 in direct-cast inference, highlighting AMXFP4's potential in lower-precision settings.

6 Conclusion

To meet the computational demands of large language models (LLMs) with extended contexts, we introduce Asymmetric Microscaling 4-bit Floating-Point (AMXFP4), which uses asymmetric shared scales to handle outliers and quantization asymmetry. AMXFP4 provides direct 4-bit inference with high accuracy, outperforming MXFP4 and other techniques for efficient, calibration-free inference.

Limitations

While AMXFP4 shows strong promise across various LLM tasks, our current hardware analysis remains focused on a MAC-level evaluation. This choice reflects a balanced starting point for proof-of-concept experiments and aligns with many common practices in precision-scaling research (Darvish Rouhani et al., 2023). However, as seen with recent system-level benchmarks (e.g., NVIDIA's Blackwell), there is significant potential to extend these findings to a full system-level evaluation. We plan to extend our evaluation accordingly, examining factors such as overall throughput, energy efficiency, and system-level trade-offs.

Additionally, our experiments have employed greedy decoding to ensure fair comparisons. However, recent deployment scenarios often rely on more advanced strategies—such as best-of-N sampling or self-refinement in reasoning LLMs—which require increased computational resources at inference time. Investigating AMXFP4's robustness and efficiency under these test-time scaling conditions is a natural next step and could further underscore the method's potential benefits in real-world applications.

References

AI@Meta. 2024. Llama 3 model card.

AMD. 2024. Amd instinct™ mi325x accelerators. `https://www.amd.com/content/dam/amd/en/documents/instinct-tech-docs/product-briefs/instinct-mi325x-datasheet.pdf`.

Michael Andersch, Greg Palmer, Ronny Krashinsky, Nick Stam, Vishal Mehta, Gonzalo Brito, and Sridhar Ramaswamy. 2022. Nvidia hopper architecture in-depth. `https://developer.nvidia.com/blog/nvidia-hopper-architecture-in-depth/`.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. 2024. Quarot: Outlier-free 4-bit inference in rotated llms. *arXiv preprint arXiv:2404.00456*.

AzureAI. 2024. Azure maia for the era of ai: From silicon to software to systems. `https://azure.microsoft.com/en-us/blog/azure-maia-for-the-era-of-ai-from-silicon-to-software-to-systems/`.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Jihwan Bang, Juntae Lee, Kyuhong Shim, Seunghan Yang, and Simyung Chang. 2024. Crayon: Customized on-device LLM via instant adapter blending and edge-server hybrid inference. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3720–3731, Bangkok, Thailand. Association for Computational Linguistics.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. *Preprint*, arXiv:1911.11641.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

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*.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Bita Darvish Rouhani, Daniel Lo, Ritchie Zhao, Ming Liu, Jeremy Fowers, Kalin Ovtcharov, Anna Vinogradsky, Sarah Massengill, Lita Yang, Ray Bittner, et al. 2020. Pushing the limits of narrow precision inferencing at cloud scale with microsoft floating point.

Advances in neural information processing systems, 33:10271–10281.

Bita Darvish Rouhani, Ritchie Zhao, Venmugil Elango, Rasoul Shafipour, Mathew Hall, Maral Mesmakhosroshahi, Ankit More, Levi Melnick, Maximilian Golub, Girish Varatkar, et al. 2023. With shared microexponents, a little shifting goes a long way. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, pages 1–13.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Jordan Dotzel, Yuzong Chen, Bahaa Kotb, Sushma Prasad, Gang Wu, Sheng Li, Mohamed S. Abdelfattah, and Zhiru Zhang. 2024. Learning from students: Applying t-distributions to explore accurate and efficient formats for llms. *International Conference on Machine Learning*.

Mario Drumond, Tao Lin, Martin Jaggi, and Babak Falsafi. 2018. Training dnns with hybrid block floating point. *Advances in Neural Information Processing Systems*, 31.

Maxim Fishman, Brian Chmiel, Ron Banner, and Daniel Soudry. 2024. Scaling fp8 training to trillion-token llms. *Preprint*, arXiv:2409.12517.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Daya Guo, Canwen Xu, Nan Duan, Jian Yin, and Julian McAuley. 2023. Longcoder: A long-range pre-trained language model for code completion. *Preprint*, arXiv:2306.14893.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *CoRR*, abs/2009.03300.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mark Horowitz. 2014. Energy table for 45nm process.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. *arXiv e-prints*, arXiv:1705.03551.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classi(cid:12)cation research.

Janghwan Lee, Minsoo Kim, Seungcheol Baek, Seok Hwang, Wonyong Sung, and Jungwook Choi. 2023. Enhancing computation efficiency in large language models through weight and activation quantization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14726–14739, Singapore. Association for Computational Linguistics.

Janghwan Lee, Seongmin Park, Sukjin Hong, Minsoo Kim, Du-Seong Chang, and Jungwook Choi. 2024. Improving conversational abilities of quantized large language models via direct preference alignment. In

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 11346–11364, Bangkok, Thailand. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. 2024. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. *Preprint*, arXiv:2406.01721.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. 2023. Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc.

Tianyang Liu, Canwen Xu, and Julian McAuley. 2023b. Repobench: Benchmarking repository-level code auto-completion systems. *Preprint*, arXiv:2306.03091.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xucheng Yin, Cheng lin Liu, Lianwen Jin, and Xiang Bai. 2024a. Ocrbench: On the hidden mystery of ocr in large multimodal models. *Preprint*, arXiv:2305.07895.

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*.

S. Lloyd. 1982. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems*.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland. Association for Computational Linguistics.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 2021. Docvqa: A dataset for vqa on document images. *Preprint*, arXiv:2007.00398.

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

Nvidia. 2017. Nvidia tesla v100 gpu architecture.

Nvidia. 2020. Nvidia a100 tensor core gpu architecture. `https://images.nvidia.com/aem-dam/en-zz/Solutions/data-center/nvidia-ampere-architecture-whitepaper.pdf`.

Nvidia. 2024. Nvidia blackwell architecture technical brief.

NVIDIA. 2024. Tensorrt-llm. `https://github.com/NVIDIA/TensorRT-LLM`.

Courtesy of the U.S. National Library of Medicine. 2023. Pubmed. `https://huggingface.co/datasets/ncbi/pubmed`.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bita Darvish Rouhani, Nitin Garegrat, Tom Savell, Ankit More, Kyung-Nam Han, Ritchie Zhao, Mathew Hall, Jasmine Klar, Eric Chung, Yuan Yu, Michael Schulte, Ralph Wittig, Ian Bratt, Nigel Stephens, Jelena Milanovic, John Brothers, Pradeep Dubey, Marius Cornea, Alexander Heinecke, Andres Rodriguez, Martin Langhammer, Summer Deng, Maxim Naumov, Paulius Micikevicius, Michael Siu, and Colin Verrilli. 2023a. Ocp microscaling formats (mx) specification.

Bita Darvish Rouhani, Ritchie Zhao, Ankit More, Mathew Hall, Alireza Khodamoradi, Summer Deng, Dhruv Choudhary, Marius Cornea, Eric Dellinger, Kristof Denolf, et al. 2023b. Microscaling data formats for deep learning. *arXiv preprint arXiv:2310.10537*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *Preprint*, arXiv:1907.10641.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

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*.

Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8317–8326.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *Preprint*, arXiv:1811.00937.

Hugo Touvron et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Julien Demouth, and Song Han. 2022. Smoothquant: Accurate and efficient post-training quantization for large language models. *arXiv preprint arXiv:2211.10438*.

Jaewoo Yang, Hayun Kim, and Younghoon Kim. 2024. Mitigating quantization errors due to activation spikes in glu-based llms. *Preprint*, arXiv:2405.14428.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *Preprint*, arXiv:1809.09600.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. 2024a. Lmms-eval: Reality check on the evaluation of large multimodal models. *Preprint*, arXiv:2407.12772.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint*.

Yijia Zhang, Sicheng Zhang, Shijie Cao, DaYou Du, Jianyu Wei, Ting Cao, and Ningyi Xu. 2024b. AFPQ: Asymmetric floating point quantization for LLMs. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 28–36, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

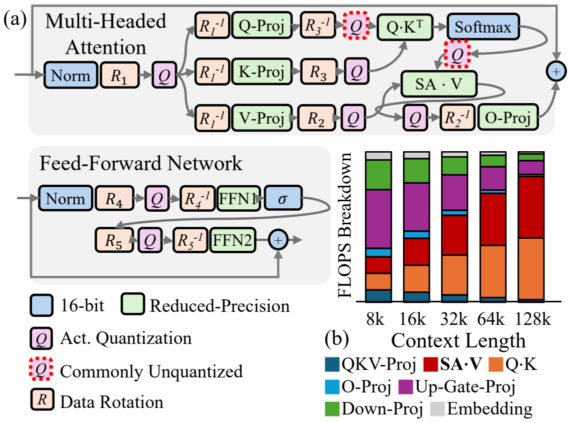


Figure 8: (a) Illustration of where reduced-precision matrix multiplication and data transformation are applied within a Transformer decoder layer. QuaRot and SpinQuant do not quantize the Query and Softmax outputs (red dotted box). (b) FLOPS breakdown of LLaMA3-8B in the prefill stage based on context length.

Rotation	Calibset-SeqLen-Samples	Calib. Time (A100)	PPL↓	Accuracy↑	
			Wiki	ARC-C	WG
16-bit Baseline			5.47	46.33	69.30
QuaRot	-	-	8.38	36.26	60.06
QuaRot+ GPTQ	Wiki-2048-128	∼20 min	6.08	41.64	66.22
	Wiki-1024-128		6.06	42.32	65.59
	Wiki-2048-64		6.11	41.64	65.51
	Wiki-2048-32		6.11	41.55	63.85
	PTB-2048-128		6.16	42.15	65.43
	PTB-1024-128		6.12	41.72	66.54
SpinQuant	Wiki-2048-100	∼2 hours	6.25	38.65	64.72
	Wiki-1024-100		6.32	40.87	63.77
	PTB-2048-100		7.11	38.74	60.30
	PTB-1024-100		7.14	37.71	63.54
AMXFP4 (direct-cast, no calibration)			**5.93**	**42.83**	**67.32**

Table 5: Calibration overhead on LLaMA2-7B.

Name	Element Data Type	Element Bits	Group Size	Shared Scale
MXFP8	FP8 (E5M2) FP8 (E4M3)	8	32	8-bit PoT
MXFP6	FP6 (E3M2) FP6 (E2M3)	6		
MXFP4	FP4 (E2M1)	4		
MXINT8	INT8	8		

Table 6: MX-compliant format. Configurations are adapted from (Rouhani et al., 2023a).

A Comparison with Rotation Techniques

Rotation-based methods, such as QuaRot and SpinQuant, typically avoid quantizing query and softmax output, and require on additional calibration, which introduces the following drawbacks:

High-Precision Query and Softmax Output. Fig. 8(a) illustrates how rotation-based methods apply rotation and quantization in reduced-precision LLM inference. While these techniques make activations more quantization-friendly, they do not quantize the softmax output. As shown in Fig. 8(b), as context length increases, the dominant FLOPS in the prefill stage come from query-key multiplication and attention operations, including softmax output (self-attention map; SA) and value multiplication. Processing these operations in high precision undermines the benefits of reduced-precision inference, limiting overall efficiency.

Calibration Overhead. Table 5 displays the effects of varying calibration settings (dataset, sequence length, and number of samples) on Wikitext-2 perplexity, ARC-Challenge (Clark et al., 2018) and WinoGrande accuracy for QuaRot and SpinQuant. When using QuaRot alone, CSQA accuracy drops by 10%. When combined QuaRot with GPTQ, results depend on calibration settings; using only 32 calibration samples leads to a 2.4% reduction in WinoGrande accuracy compared to using 128 samples. SpinQuant, which trains a rotation matrix, achieves higher accuracy than QuaRot alone but increases calibration time by approximately 6× and exhibits greater sensitivity to the calibration set. When calibrated with the PTB (Marcus et al., 1993) dataset instead of Wikitext-2, perplexity on Wikitext-2 rises by around 0.9. Our proposed AMXFP4 shows minimal performance degradation compared to the baseline and remains unaffected by calibration settings.

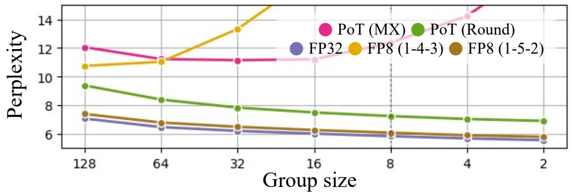
B MX Format Details and Emulation Framework

B.1 MX Configuration

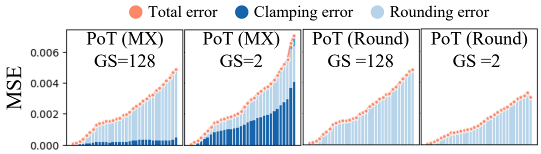
Algorithm 1 Quantization procedure in MX format. Algorithm is adapted from (Rouhani et al., 2023b).

```
Quantize vector elements ($\{V_i\}_{i=1}^{k}$) into MX format
$shared\_exp \leftarrow \lfloor \log_2(\max_i(|V_i|)) \rfloor - emax_{elem}$
$X \leftarrow 2^{shared\_exp}$
for $i = 1$ to $k$ do
    $P_i = \text{quantize}(V_i/X)$, clamping normal numbers
end for
return $X$, $\{P_i\}_{i=1}^{k}$
```

As the MX format is our primary focus for improvement, we aim to provide detailed information on it. We follow the MX format configuration and quantization procedure as (Rouhani et al., 2023a,b). The MX format offers a variety of bit-configurations for elements, ranging from 8 bits to 4 bits, while specifying only an 8-bit PoT for the shared scale. The process to determine this 8-bit PoT follows an Algorithm 1. As described in the



(a) Perplexity across data types of shared scale



(b) Output Error Breakdown

Figure 9: Impact of shared scale (LLaMA2-7B). More results on other models and data formats are in Table 15.

Data Format	Area-Memory	Power-Area	Power-Area -Memory
FP16	1.00×	1.00×	1.00×
MXFP4-PoT	10.44×	7.62×	28.67×
MXFP4	9.23×	5.65×	21.41×
AMXFP4	8.32×	4.58×	16.50×

Table 7: Hardware comparison between MXFP4 and AMXFP4.

entire quantization procedure, MX considers the maximum data value to determine the shared scale, performing a floor operation after extracting the exponent of the element's maximum value with log2.

B.2 Determining PoT Shared Scale: Floor vs. Round

As illustrated in Fig. 9(a), an undesirable performance degradation occurs in PoT scales as group size decreases. To analyze this degradation, we decompose the output error into maximum clamping error and rounding error. As shown in Fig. 9(b), with a group size of 2, the rounding error reduces significantly, while the maximum clamping error increases sharply, resulting in a net error rise. This issue is attributed to the floor operation on the exponent in MX, which introduces clamping error. To overcome maximum clamping errors while maintaining the hardware efficiency of PoT shared scales, we replace flooring with rounding. This exponent rounding approach significantly lowers total error, enhancing performance, as demonstrated in Fig. 9(a) and (b).

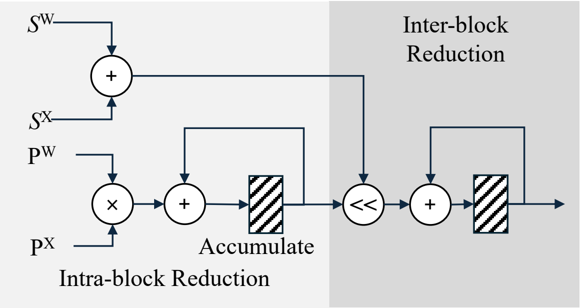


Figure 10: MX dot-product architecture.

B.3 Hardware Evaluation for MXFP4 and AMXFP4

Since a MAC unit is a major consumer of ASIC resources for deep learning accelerators, many representative prior works focus on MAC unit efficiency for hardware analysis (Darvish Rouhani et al., 2023). Thus, we also follow and expand (Darvish Rouhani et al., 2023)'s evaluation process (area, memory-efficiency) for AMXFP4. We design a fully custom MX-compatible MAC unit and its extension to AMX. Then, we synthesize it under a competitive operating environment with Synopsys Design Compiler (commercial 4nm technology node, supply voltage of 0.675V, and a clock frequency of 1.1GHz).

MX format's group-wise data representation decouples intra-group dot products from group-wise scaling, enabling efficient MAC implementations with minimal overhead from inter-group scale adjustments (Fig. 10). As shown in Table 7, our MX-compatible MAC unit implementation shows that 4-bit MX formats reduce area-memory costs by over 8×. This aligns with MXFP4's early adoption in recent deep learning accelerators, offering a 2× speedup over 8-bit computation (Nvidia, 2024; AzureAI, 2024).

Asymmetric FP8 scales require additional multiplication of the mantissas for positive and negative scales depending on the operand's sign; however, this incurs minimal resource overhead due to the small mantissa size, and once calculated, the overhead is shared within a group. Our evaluation shows that AMXFP4 incurs only a 10% overhead compared to MXFP4.

B.4 Code Snippet of Our Framework

As shown in the below example, our proposed AMXFP4 applies different shared scales to positive and negative numbers, enabling more refined value representation compared to MXFP4. Additionally, the PoT shared scale significantly clamps the largest value in the input, 31, to 24, while the FP8 shared scale, using the same number of bits, more precisely quantizes 31 to 30.

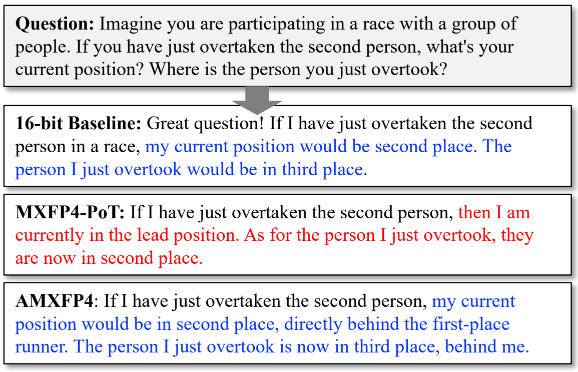
```
class MXQuantizer(object):
    def __init__(self, elem_format,
        group_size, scale_mode):
        self.elem_format = elem_format #
            Element Format
        self.group_size = group_size # group
            Size
        self.scale_mode = scale_mode #
            Shared Scale Type
        self.mx_specs = MxSpecs(
            a_elem_format=self.elem_format,
            group_size=self.group_size,
            custom_cuda=True,
            scale_mode=scale_mode,
        )
    def quantize(self, x):
        qx = quantize_mx_op(
            x,
            self.mx_specs,
            elem_format=self.elem_format,
            axes=[-1],
        )
        return qx

# Example: Asymmetrically distributed tensor
    with a single row
x = torch.linspace(-4.9, 31, 1024)

# MXFP4
mx_fp4 = MXQuantizer(elem_format='fp4_e2m1',
    group_size=-1, scale_mode=0)
qx_mx_fp4 = mx_fp4.quantize(x)
# AMXFP4 (Shared Scale: PoT)
mx_fp4_asym =
    MXQuantizer(elem_format='fp4_e2m1_asym',
    group_size=-1, scale_mode=0)
qx_mx_fp4_asym = mx_fp4_asym.quantize(x)
# AMXFP4 (Shared Scale: FP8)
mx_fp4_asym_fp8scale =
    MXQuantizer(elem_format='fp4_e2m1_asym',
    group_size=-1, scale_mode=152)
qx_mx_fp4_asym_fp8scale =
    mx_fp4_asym_fp8scale.quantize(x)

# Quantized tensor
print(qx_mx_fp4.unique()) # MXFP4
>> tensor([-4., -2.,  0.,  2.,  4.,  6.,
    8., 12., 16., 24.], device='cuda:0')
print(qx_mx_fp4_asym.unique()) # AMXFP4
    (Shared Scale: PoT)
>> tensor([-4.0000, -3.0000, -2.0000,
    -1.5000, -1.0000, -0.5000,  0.0000,
    2.0000,
         4.0000,  6.0000,  8.0000, 12.0000,
             16.0000, 24.0000],
             device='cuda:0')
print(qx_mx_fp4_asym_fp8scale.unique()) #
    AMXFP4 (Shared Scale: FP8)
>> tensor([-5.2500, -3.5000, -2.6250,
    -1.7500, -1.3125, -0.8750, -0.4375,
    0.0000,
         2.5000,  5.0000,  7.5000, 10.0000,
             15.0000, 20.0000, 30.0000],
       device='cuda:0')
```

C Experimental Details

Quantization Settings. Our experiments is conducted by modifying the PyTorch and CUDA code within the `MX Emulation library` (Rouhani et al., 2023b). We quantize all weights and activations in Transformer decoder layers, including ***Query***, ***Key***, ***Self-attention map***, **and** ***Value*** as a default.

Models. The models used in the experiments include OPT (Zhang et al., 2022), LLaMA (Touvron et al., 2023), (AI@Meta, 2024), Qwen (Bai et al., 2023), and Mistral (Jiang et al., 2023), LLaMA2-Chat (Touvron et al., 2023), BART (Lewis et al., 2019), and LLaVA (Liu et al., 2023a) (which backbone is Vicuna-7B (Chiang et al., 2023)).

Robustness Measurment Settings in Table 2. Following the calibration robustness measurement method introduced in AWQ (Lin et al., 2023), we select two subsets from the Pile dataset (Gao et al., 2020): PubMed Abstracts (of the U.S. National Library of Medicine, 2023) and Enron Emails (Klimt and Yang, 2004). The calibration and evaluation sets are distinct, with no overlap; 128 samples with a sequence length of 2048 are used for calibration, and 200 samples are reserved for perplexity evaluation. Additionally, we configure the calibration set with questions and answers from the PIQA (Bisk et al., 2019) and WinoGrande (Sakaguchi et al., 2019) datasets to analyze calibration effects in question-answering tasks. To determine whether our improved MX format can effectively replace existing techniques for W4A4 inference, we align the experimental settings, applying reduced-precision activations consistent with prior studies (excluding



Question: Imagine you are participating in a race with a group of people. If you have just overtaken the second person, what's your current position? Where is the person you just overtook?

16-bit Baseline: Great question! If I have just overtaken the second person in a race, my current position would be second place. The person I just overtook would be in third place.

MXFP4-PoT: If I have just overtaken the second person, then I am currently in the lead position. As for the person I just overtook, they are now in second place.

AMXFP4: If I have just overtaken the second person, my current position would be in second place, directly behind the first-place runner. The person I just overtook is now in third place, behind me.

Figure 11: Example of chatbot interactions from MT-Bench (LLaMA2-Chat-7B)

Data Format	OPT		LLaMA			Mistral
	6.7B	13B	2-7B	2-13B	3-8B	7B
16-bit Baseline	10.86	10.13	5.47	4.88	6.14	5.25
MXFP4-PoT	25.51	12.88	7.83	6.98	11.17	6.34
MXFP4	13.71	12.09	6.49	5.69	8.31	5.88
AMXFP4	**13.06**	**11.90**	**6.22**	**5.47**	**7.72**	**5.71**

Table 8: Wikitext-2 inference for MXFP4 and AMXFP4.

Data Format	ROUGE-1 ↑	ROUGE-2 ↑	ROUGE-L ↑
16-bit Baseline	45.09	21.60	31.43
MXFP4-PoT	42.47	19.10	29.18
MXFP4	43.73	20.50	30.43
AMXFP4	**44.13**	**20.79**	**30.72**

Table 9: CNN/DailyMail summarization task on BART-Large.

quantization for Query and Softmax output). We reproduce the performance of QuaRot and SpinQuant following their official repositories, with modifications to calibration and evaluation datasets.

MT-Bench. MT-Bench assigns scores ranging from 1 to 10, given by GPT-4 (OpenAI, 2023), to responses generated from an initial question and a subsequent follow-up question across 80 multi-turn conversations.

Visual Tasks. For evaluating VLMs, we utilize `lmms-eval` (Zhang et al., 2024a), including TextVQA (VQA-T) (Singh et al., 2019), DocVQA (Mathew et al., 2021), OCRBench (Liu et al., 2024a), and ChartQA (Masry et al., 2022).

Long-Context Benchmarks. To measure the effectiveness of AMXFP4 while long-context is given, we utilize LongBench-E (Bai et al., 2024) on LLaMA2-Chat-7B. LongBench-E includes 13 tasks: Qasper (Dasigi et al., 2021), MultiFieldQA (Bai et al., 2024), HotPotQA (Yang et al., 2018), MultihopQA (Ho et al., 2020), GovReport (Huang et al., 2021), MultiNews (Bai et al., 2024), TREC (Li and Roth, 2002), TriviaQA (Joshi et al., 2017), SAMSum (Gliwa et al., 2019), PassageCount (Bai et al., 2024), PassageRetrieval (Bai et al., 2024), LCC (Guo et al., 2023), and RepoBench-P (Liu et al., 2023b).

Knowledge Evaluation Benchmarks (MMLU and CSQA). We evaluate our method into commonsense QA (CSQA) (PIQA (Bisk et al., 2020), WinoGrande (Sakaguchi et al., 2019), ARC challenge (Clark et al., 2018)) and MMLU (Hendrycks et al., 2020). and CSQA and MMLU accuracies are assessed using the `lm-evaluation-harness` (Gao et al., 2021).

D More Experimental Results

D.1 Ablation Studies

Language Modeling Tasks. We evaluate on language modeling with WikiText (Merity et al., 2016). The perplexity measurement on the Wikitext test dataset involves grouping 2048 tokens collectively. Table 8 presents Wikitext-2 perplexity results for six LLMs across MXFP4 and AMXFP4 with PoT and FP8 shared scale. While MXFP4-PoT introduces significant perplexity degradation across all models, employing MXFP4 with an enhanced shared scale substantially reduces perplexity in each case. Notably, AMXFP4, through asymmetric data representation, achieves a 0.59 perplexity reduction in LLaMA3-8B compared to MXFP4 and limits perplexity degradation to only about 0.46 in models like Mistral-7B.

Encoder-Decoder Language Model. Table 9 displays the ROUGE (Lin, 2004) scores for BART-Large's (Lewis et al., 2019) summarization task on the CNN/DailyMail dataset (See et al., 2017) across different MX format options. AMXFP4 exhibits only a 0.7-point drop in ROUGE-L score compared to the baseline, demonstrating that the proposed data format also enables effective 4-bit inference in encoder-decoder models.

Quantization-Aware Training. We conduct quantization-aware training (QAT) experiments on LLaMA3-8B, specifically because it exhibits relatively high perplexity degradation under direct-cast quantization. We quantize all linear layer weights and activations to 4 bits and employ flash-attention (Dao et al., 2022) for attention operations. We construct QAT dataset by randomly sampling 3200 sequences, each with a length of

Method	Data Format	Wiki-2 PPL ↓	Required Memory for Fine-Tuning	Required Time for Fine-Tuning
Direct-Cast	16-bit-Baseline	6.14	-	-
Direct-Cast	MXFP4-PoT	7.70	-	-
Direct-Cast	AMXFP4	**6.97**	-	-
QAT	MXFP4-PoT	6.68	148GB	4h 30m
QAT	AMXFP4	**6.33**	148GB	4h 30m

Table 10: Quantization-aware training results on LLaMA3-8B with Wikitext-2 dataset. Training time is measured in two A100-80GB GPUs.

Method	Format	Direct-cast?	LLaMA2-7B	LLaMA2-13B
16-bit Baseline			5.47	4.88
QuaRot-RTN	INT	no	1032.30	1105.95
QuaRot-GPTQ			38.47	37.42
AMXFP3	MX	yes	**8.40**	**6.53**

Table 11: Wikitext-2 perplexity results on 3-bit inference.

2048 tokens (a total of 6.5M tokens), from the Wikitext-2 training set. Training is performed for 100 steps with an effective batch size of 32, and we search learning rates between 2e-6 and 1e-5 to determine the best hyperparameters for both MXFP4 and AMXFP4. As shown in Table 10, under direct-cast inference, MXFP4 exhibits a perplexity degradation of 1.6 compared to the 16-bit baseline, whereas our proposed AMXFP4 experiences only a 0.8 increase. QAT significantly reduces perplexity for both MXFP4 and AMXFP4, with AMXFP4 still achieving lower perplexity than MXFP4, approaching baseline levels. However, QAT requires 150GB of GPU memory and 5 hours of fine-tuning time, in addition to the overhead for hyperparameter tuning.

More Aggressive Quantization. To explore the potential future of the proposed microscaling format, we compare AMXFP with QuaRot and MXFP at settings lower than 4-bit. Table 11 shows the inference results for the LLaMA2 model at 3-bit. In 3-bit inference, QuaRot with GPTQ shows significant performance degradation. MXFP3-PoT also experiences a significant deterioration in perplexity. Conversely, AMXFP3 demonstrates a significant improvement in perplexity, indicating that our findings are effective at lower bit settings. This underscores the robustness of AMXFP in maintaining performance with reduced bit precision, potentially paving the way for more efficient computational models in resource-constrained environments.

Conjunction with Sparsity We conduct an ablation study by applying MXFP4 to a pruned model to see if improvements in the micro-scaled reduced-precision option can work in conjunction with other methods like sparsity. We use 20% pruning with LLM-Pruner (Ma et al., 2023) as the baseline for the sparse model. Table 12 shows the accuracy when applying various MXFP4 options to the pruned model for four CSQA tasks. The model with 20% pruning reduces the requried memory while tolerating a slight drop in accuracy. Applying MXFP4-PoT to the pruned model results in an additional 5% performance drop. On the other hand, advancements in shared scale and the representation of asymmetric data have progressively enhanced accuracy even in pruned models, showing that the improvements of the proposed MX format have a cumulative effect.

Ablation Study on Shared-Scale Bit-Encoding. Table 15 illustrates the perplexity according to the type of shared scale across various models and group sizes. In the case of FP4, using the default 8-bit PoT (Floor) shared scale option of MX, there is a notable increase in perplexity as the group size decreases. This trend is also observed in AsymFP4, primarily due to the increased error from frequent clamping caused by the Floor operation. To address this, our proposed 8-bit PoT consistently improves performance even with smaller group sizes. On the other hand, FP8, another 8-bit alternative, with a 4-bit exponent, significantly degrades performance in models like Mistral, a consequence of its inherent limitations in dynamic range. Conversely, our findings demonstrate that using a 5-bit exponent FP8 shared scale can achieve performance close to FP16.

D.2 Detailed Results for Practical Applications

Chatbot Results. Fig. 13 presents an example from MT-Bench. While the 16-bit baseline provides responses aligned with the user's intent, MXFP4 tends to generate repetitive and unhelpful sentences. In contrast, AMXFP4 produces responses that, similar to the baseline, are useful to the user. Table 13 displays the single scores from MT-Bench across different categories. The proposed AMXFP4 demonstrates the ability to recover baseline performance in most sub-categories.

Visual Question Answering Results. Fig. 12 presents an example response to a given chart image using MXFP4 and AMXFP4. While MXFP4-PoT generates irrelevant responses, AMXFP4 produces the correct ground-truth answer, identical to the baseline.

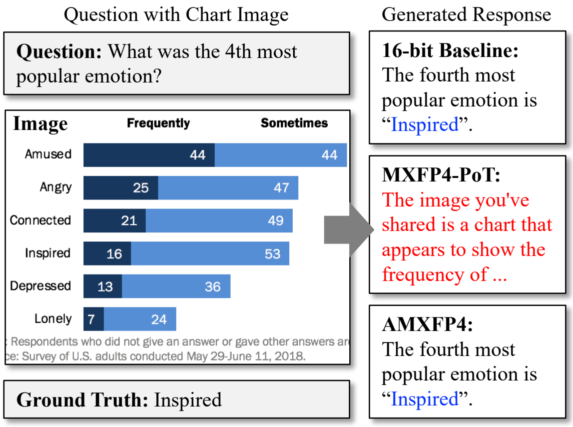
LongBench-E Results. Table 14 provides de-

Pruning Ratio	**Bit-Configurations**	**Memory (GB)**	**BoolQ**	**OBQA**	**PIQA**	**ARC-C**	**Average ↑**
0%	FP16	13.48	75.11	44.40	79.16	44.71	60.85
20%	FP16	10.85	66.45	41.40	78.13	39.42	56.35
20%	MXFP4-PoT	3.27	61.74	36.80	73.39	35.15	51.77
20%	MXFP4	3.27	62.91	37.60	75.19	36.77	53.12
20%	AMXFP4	3.27	62.72	38.60	75.73	36.43	53.37

Table 12: Performance comparison across different pruning ratios and bit configurations (LLaMA-7B).

Data Format	Writing	Roleplay	Reasoning	Math	Coding	Extraction	STEM	Humanities	Single Score
16-bit Baseline	9.25	7.20	4.65	2.55	3.30	5.55	8.93	9.58	6.38
MXFP4-PoT	4.30	4.05	2.35	1.90	1.25	1.55	5.23	5.15	3.22
MXFP4	7.20	**7.03**	3.95	1.70	1.70	4.35	7.53	8.53	5.25
AMXFP4	**8.20**	5.98	**4.50**	**2.50**	**3.05**	**5.16**	**7.70**	**8.70**	**5.73**

Table 13: MT-Bench Single Score (LLaMA2-Chat-7B).


Question with Chart Image
Generated Response
Image
Frequently
Sometimes
Amused
44
44
Angry
25
47
Connected
21
49
Inspired
16
53
Depressed
13
36
Lonely
7
24
Respondents who did not give an answer or gave other answers ar
ce: Survey of U.S. adults conducted May 29-June 11, 2018.

Question: What was the 4th most popular emotion?

Ground Truth: Inspired

16-bit Baseline: The fourth most popular emotion is “Inspired”.

MXFP4-PoT: The image you've shared is a chart that appears to show the frequency of ...

AMXFP4: The fourth most popular emotion is “Inspired”.

Figure 12: Comparison between responses from MXFP4-PoT and AMXFP4 in ChartQA example.

tailed scores for 13 benchmarks in LongBench-E. MXFP4-PoT shows significant performance degradation overall, with a score drop of 16 for contexts exceeding 8K. While MXFP4 substantially improves over MXFP4-PoT, it still experiences a score reduction of over 6 when handling contexts above 8K. AMXFP4 increases the average score by more than 2 compared to MXFP4 for long-context scenarios exceeding 8K.

Question from User: The city of Vega intends to build a bridge that will span the Vegona River, covering a distance of 1.8 kilometers. The proposed location falls within a seismically active area that has experienced several high-magnitude earthquakes. Given these circumstances, what would be the best approach to constructing the bridge?

Response from 16-bit Baseline: Thank you for your question. Building a bridge in a seismically active area requires careful consideration and a multi-faceted approach to ensure the safety of the structure and its users. Here are some key factors to consider:\n1. Seismic Design: The bridge must be designed to withstand earthquakes. This involves using seismic-resistant materials, such as reinforced concrete or steel, and incorporating seismic isolation systems, such as base isolators or seismic isolation bearings. These systems can help reduce the bridge's response to seismic forces, minimizing damage and ensuring continued functionality during earthquakes.\n2. Site Selection: Choose a location for the bridge that is away from potential earthquake fault lines and has a low risk of landslides or other geological hazards. This may involve conducting geological surveys and assessments to identify potential hazards and selecting a site that is safe and stable.\n3. Structural Integrity: The bridge's structural elements, such as the piers, abutments, and superstructure, must be designed to withstand the forces generated by earthquakes. This may involve using reinforced concrete or steel, and incorporating redundancy in the structure to ensure that it can continue to function even if some elements fail during an earthquake.\n4. Foundation Design: The bridge's foundation must be designed to transfer the loads from the bridge to the ground without settling or failing during earthquakes. This may involve using deep foundations, such as piles or caissons, that are capable of resisting lateral loads and settlement. ...

Response from MXFP4-PoT: I'm glad you're interested in building a bridge! Let's explore some considerations first.\n1. Seismic activity: As you've noticed, Vega is an active seismic zone. Given the geological activity, what would be the best approach to constructing the bridge?\n2. Geological constraints: Vega is a volcanic in origin, and the magma is not a problem. The volcanic activity is not a concern for the bridge construction.\n3. Tectonic activity: Vega is a tectonic plate, and the magma is not a problem. The tectonic activity is not a concern for the bridge construction.\n4. Volcanic activity: Vega is a volcanic in origin, and the magma is not a problem. The volcanic activity is not a concern for the bridge construction.\n5. Geological constraints: Vega is a geological hotspot, and the geological activity is not a concern for the bridge construction.\n6. Tectonic activity: Vega is a tectonic spot, and the tectonic activity is not a problem for the bridge construction.\n7. Earthquake activity: Vega is an earthquake-prone area, and the earthquake activity is not a concern for the bridge construction.\n8. Geological constraints: Vega is a geological hotspot, and the geological activity is not a concern for the bridge construction.\n9. Tectonic activity: Vega is a tectonic hotspot, and the tectonic activity is not a concern for the bridge construction.\n10. Geological constraints: Vega is a geological hotspot, and the geological activity is not a concern for the bridge construction.\n1. Volcanic activity: Vega is a volcanic in origin, and the volcanic activity is not a concern for the bridge construction.\n2. Seismic activity: Vega is a seismic in origin, and the seismic activity is not a concern for the bridge construction.\n3. Tectonic activity: Vega is a tectonic hotspot, and the tectonic activity is not a concern for the bridge construction.\n4. Volcanic activity: Vega is a volcanic in origin, and the volcanic activity is not a concern for the bridge construction.\n5. Geological constraints: Vega is a geological hotspot, and the geological activity is not a concern for the bridge construction. ...

Response from AMXFP4: Thank you for sharing this information. Given the seismic activity in the area, it is crucial to prioritize safety when designing and constructing the bridge. Here are some recommendations for the best approach:\n1. Seismic Design: Incorporate seismic design principles into the bridge's structure and materials. This may include using seismic-resistant materials, such as reinforced concrete or steel, and designing the bridge with seismic loads in mind.\n2. Damping Systems: Implement damping systems to reduce the bridge's response to seismic activity. These can include devices such as shock absorbers or dampers, which can help reduce the bridge's motion and prevent damage from earthquakes.\n3. Reinforced Foundations: Reinforce the bridge's foundations with additional support systems, such as piles or caissons, to ensure stability during seismic activity. This will help prevent the bridge from shifting or collapsing during earthquakes.\n4. Monitoring Systems: Install monitoring systems to track the bridge's condition during seismic activity. This can include sensors to measure movement, strain, or other parameters, which can help identify potential issues before they become major problems.\n5. Emergency Response Plan: Develop an emergency response plan that includes procedures for assessing and repairing damage to the bridge during or after an earthquake. This can help ensure the safety of the bridge and those who use it.\nBy prioritizing safety and incorporating these recommendations into the bridge's design and construction, you can help ensure the safety of the bridge and those who use it during seismic activity. ...

Figure 13: MT-Bench example (LLaMA2-Chat-7B).

Data Format	Context Length	Single Doc-QA		Multi Doc-QA		Summarization		Few-shot Learning			Synthetic Tasks		Code Completion		Average
		Qasper	MultiField QA	Hotpot QA	Multihop QA	Gov Report	Multi News	TREC	Trivia QA	SAM Sum	Passage Count	Passage Retrieval	LCC	Repo Bench-P	
16-bit Baseline	0-4k	22.99	43.37	37.14	35.79	31.13	26.84	54.00	83.13	39.33	6.35	18.00	62.45	49.02	39.20
	4-8k	18.37	32.29	30.47	24.36	27.89	23.14	60.00	84.02	37.73	2.01	4.00	59.98	48.05	34.79
	8k+	21.42	25.59	24.08	23.37	25.14	23.11	60.00	91.51	40.22	2.72	7.00	56.88	48.51	34.58
MXFP4-PoT	0-4k	12.02	31.91	14.27	15.82	20.23	20.16	32.00	44.39	28.37	4.48	9.42	31.54	34.96	23.04
	4-8k	11.02	17.56	13.83	13.32	15.71	13.96	37.00	36.66	25.93	6.07	2.12	32.13	32.50	19.83
	8k+	9.27	10.26	10.78	10.10	13.94	13.13	36.00	41.83	24.92	5.72	5.09	27.31	35.29	18.74
MXFP4	0-4k	13.16	40.81	25.27	24.27	22.68	23.66	46.00	77.49	38.97	5.71	9.98	49.54	41.24	32.21
	4-8k	14.26	27.40	21.96	19.36	19.91	18.59	58.00	75.53	35.98	1.50	0.79	48.15	38.45	29.22
	8k+	10.04	23.07	19.15	17.19	18.09	18.66	49.00	79.39	37.82	3.68	5.00	45.10	41.77	28.30
AMXFP4	0-4k	16.93	34.62	32.16	25.52	23.21	23.49	50.00	76.52	37.88	9.81	10.50	50.76	43.95	**33.49**
	4-8k	19.56	26.96	26.03	19.74	19.80	19.71	54.00	70.53	36.29	2.04	5.27	48.05	40.77	**29.90**
	8k+	34.32	17.40	20.52	21.72	18.04	18.70	50.00	79.92	38.73	3.39	9.00	45.12	40.50	**30.57**

Table 14: Detailed scores of LongBench-E (Bai et al., 2024).

Data Format	Shared Scale	Group Size	OPT		LLaMA2		LLaMA3	Mistral 7B	Qwen
			6.7B	13B	7B	13B	8B	7B	7B
16-bit Baseline			10.860	10.128	5.472	4.884	6.137	5.252	7.605
MXFP4	FP16	128	12.566	12.415	7.065	6.208	9.826	6.137	8.669
		64	11.843	11.958	6.470	5.667	8.368	5.854	8.364
		32	11.475	11.084	6.206	5.444	7.851	5.722	8.214
		16	11.233	10.841	6.015	5.284	7.334	5.607	8.084
	PoT (Floor)	128	24.126	16.151	12.056	11.243	17.848	8.454	10.407
		64	22.605	14.820	11.228	10.453	16.636	8.846	10.023
		32	22.525	14.473	11.150	10.270	16.636	9.454	9.762
		16	23.463	14.638	11.212	10.065	18.582	10.392	9.651
	PoT (Round)	128	40.288	14.460	9.383	8.472	15.741	7.000	9.635
		64	27.696	13.238	8.393	7.669	12.450	6.585	9.185
		32	25.512	12.879	7.834	6.982	11.171	6.337	8.940
		16	25.155	12.683	7.495	6.649	10.381	6.206	8.764
	FP8 (1-4-3)	128	21.914	14.075	10.749	9.883	9.842	55.719	8.783
		64	18.637	15.840	11.036	9.340	8.761	670.647	8.458
		32	24.109	21.447	13.334	9.705	8.733	6050.050	8.358
		16	28.186	33.131	17.082	11.330	8.340	25756.484	8.229
	FP8 (1-5-2)	128	15.857	14.530	7.390	6.450	10.408	6.234	8.806
		64	14.075	12.777	6.788	5.923	8.952	5.957	8.542
		32	13.712	12.091	6.490	5.691	8.307	5.883	8.366
		16	13.534	11.808	6.265	5.520	7.824	5.725	8.247
AMXFP4	FP16	128	12.107	11.718	6.564	5.712	8.364	5.898	8.408
		64	11.489	11.187	6.173	5.400	7.660	5.702	8.272
		32	11.242	10.900	5.999	5.261	7.296	5.588	8.066
		16	11.118	10.581	5.840	5.149	6.978	5.507	7.953
	PoT (Floor)	128	23.161	15.074	11.555	10.839	18.404	8.594	10.123
		64	24.002	14.635	10.956	10.380	18.910	9.217	9.840
		32	25.233	14.569	11.362	10.433	18.748	10.710	9.584
		16	27.992	14.910	12.255	11.118	22.084	14.090	9.595
	PoT (Round)	128	28.781	13.485	8.454	7.466	12.307	6.517	9.235
		64	26.021	12.939	7.803	7.002	10.683	6.311	8.987
		32	24.995	12.651	7.456	6.596	10.048	6.189	8.780
		16	24.240	12.585	7.172	6.362	9.688	6.120	8.673
	FP8 (1-4-3)	128	17.243	13.764	9.725	8.966	8.640	1053.763	8.468
		64	18.093	16.331	10.582	8.622	8.609	3718.406	8.303
		32	20.803	22.674	13.080	9.435	8.193	13421.343	8.231
		16	31.017	40.884	17.459	11.331	8.260	30513.367	8.175
	FP8 (1-5-2)	128	14.580	12.652	6.847	5.901	8.777	6.003	8.568
		64	13.480	12.132	6.451	5.618	8.092	5.817	8.400
		32	13.058	11.902	6.223	5.469	7.725	5.707	8.215
		16	12.941	11.625	6.064	5.374	7.421	5.632	8.114

Table 15: Ablation study on shared scale bit-encoding.

Cluster ID	Centroids		Data Formats						
	Normalized Mean	Normalized Kurtosis	NF4	SF4	INT4	Asym INT4	FP4	Asym FP4	Lloyd-Max
0	0.041	0.003	4.14E-04	5.24E-04	5.77E-04	3.90E-04	5.45E-04	4.65E-04	3.85E-04
1	-0.084	0.472	2.63E-03	1.86E-03	7.06E-03	2.41E-03	2.42E-03	1.43E-03	8.07E-04
2	-0.357	-0.010	4.18E-04	5.70E-04	4.80E-04	3.17E-04	5.40E-04	4.77E-04	3.30E-04
3	0.533	-0.016	3.72E-04	5.27E-04	4.16E-04	2.68E-04	5.44E-04	4.91E-04	2.89E-04
4	0.100	0.577	4.01E-03	2.80E-03	1.06E-02	3.44E-03	3.80E-03	2.19E-03	9.62E-04
5	0.231	-0.002	4.04E-04	5.17E-04	5.55E-04	3.61E-04	5.47E-04	4.71E-04	3.70E-04
6	-0.137	-0.001	4.16E-04	5.39E-04	5.51E-04	3.72E-04	5.41E-04	4.64E-04	3.72E-04
7	-0.236	-0.003	4.20E-04	5.48E-04	5.32E-04	3.52E-04	5.40E-04	4.68E-04	3.59E-04
8	-0.084	0.206	1.13E-03	8.89E-04	2.76E-03	1.18E-03	1.10E-03	7.67E-04	7.36E-04
9	0.353	-0.009	3.83E-04	5.14E-04	4.87E-04	3.18E-04	5.39E-04	4.76E-04	3.33E-04
10	-0.093	0.772	8.39E-03	5.83E-03	2.02E-02	6.59E-03	7.95E-03	4.18E-03	1.60E-03
11	-0.046	0.000	4.10E-04	5.29E-04	5.50E-04	3.78E-04	5.40E-04	4.61E-04	3.73E-04
12	0.096	0.830	1.14E-02	7.93E-03	2.58E-02	8.76E-03	1.09E-02	5.78E-03	1.86E-03
13	0.113	0.279	1.53E-03	1.15E-03	3.93E-03	1.52E-03	1.47E-03	9.78E-04	8.58E-04
14	0.132	0.002	4.12E-04	5.22E-04	5.79E-04	3.84E-04	5.48E-04	4.68E-04	3.86E-04
15	-0.533	-0.016	4.19E-04	5.95E-04	4.12E-04	2.69E-04	5.38E-04	4.85E-04	2.86E-04
Overall Error			1.09E-03	9.74E-04	2.25E-03	9.15E-04	1.17E-03	<u>7.89E-04</u>	**4.83E-04**

Table 16: Detailed MSE across clusters (LLaMA2-7B Layer 5 QKV-Proj Activations in Wikitext-2 inference).