Quantize What Counts: More for Keys, Less for Values

Mohsen Hariri[1], **Alan Luo**[1], **Weicong Chen**[1], **Tianyi Zhang**[2], **Qifan Wang**[3], **Xiaotian Han**[1], **Vipin Chaudhary**[1]

[1]Case Western Reserve University; [2]Rice University; [3]Meta

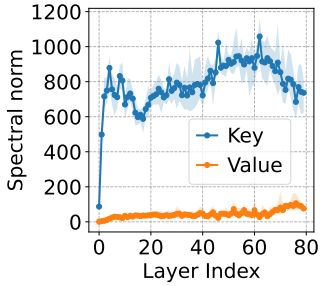
mohsen.hariri@case.edu

Abstract

Large Language Models (LLMs) suffer inference-time memory bottlenecks dominated by the attention Key-Value (KV) cache, which scales with model size and context length. While KV-cache quantization alleviates this cost, bit allocation between keys and values is often tuned heuristically, lacking theoretical grounding and generalizability. This paper proposes two theorems that anchor mixed-precision KV quantization in the intrinsic geometry of Transformer models. First, key projections systematically have larger spectral and Frobenius norms than value matrices, implying higher information density along the key path. Second, for any given memory budget, prioritizing precision for keys over values strictly reduces quantization error and better preserves accuracy. Empirical evaluations across various prominent LLMs and benchmarks show that key-favored allocations (e.g., 4-bit keys, 2-bit values) retain up to 98.3% accuracy compared to uniform allocations (e.g., 4-bit for both), while conserving memory. These results transform bit allocation from ad hoc tuning into a theoretically grounded, geometry-driven design principle for efficient LLM inference. Source code is available at https://github.com/mohsenhariri/spectral-kv.

1 Introduction

Large Language Models (LLMs) have rapidly scaled in recent years, driving major advances in generative capabilities and reasoning performance (Vaswani et al., 2017; Sutskever et al., 2014). Model size has increased by several orders of magnitude: GPT has grown from 117M parameters in GPT-1 (Radford et al., 2018) to 1.5B in GPT-2 (Radford et al., 2019), 175B in GPT-3 (Brown et al., 2020), and 1.8T in GPT-4 (OpenAI et al., 2024). Open-source models have followed a similar trajectory, with Llama reaching 2T parameters (Meta AI, 2025b), Mistral Large scaling to 123B (Mistral AI, 2024), and DeepSeek V3 to 671B (DeepSeek-AI et al., 2025b).



Size	K_2V_4	K_4V_2
1B	0.06	0.34
8B	0.55	0.75
14B	0.78	0.91
70B	0.76	0.87

Figure 1: **Key cache needs more bits.** (**Left**): Spectral norms of the key cache (blue) and value cache (orange) across layers in Llama3.3-70B show that key caches consistently exhibit higher norms. (**Right**): GSM8k accuracy for two schemes: K_2V_4, representing 2-bit allocation for the K cache and 4-bit allocation for the V cache and K_4V_2, representing 4-bit allocation for the K cache and 2-bit allocation for the V cache, demonstrates that allocating more bits to the key cache maintains strong performance, confirming the efficacy of norm-aware, mixed-precision quantization.

However, this rapid growth has introduced severe *inference-time memory bottlenecks*, primarily due to the Key-Value (KV) cache (Wei et al., 2022). As parameter counts increase, context lengths must also grow to support more complex reasoning, which further expands the KV cache and strains GPU memory (Sheng et al., 2023a). Modern systems already support extremely long contexts (Google DeepMind, 2025; OpenAI, 2024, 2025; DeepSeek-AI et al., 2025a), reaching up to 10 million tokens (Meta AI, 2025a), making memory-efficient methods essential.

KV cache quantization refers to reducing the precision of KV tensors (e.g., BF16 to INT4), which can provide substantial memory savings while maintaining *controlled accuracy degradation*, provided applied strategically (Han et al., 2016; Nagel et al., 2021). Although many KV quantization methods have been proposed, most determine key–value bit splits through ad hoc hyperparame-

ter tuning on cache statistics (e.g., inference-time activations) rather than grounding them in intrinsic model properties (e.g., model weights). This raises a fundamental question: ***How should bits be allocated in a principled and generalizable way?***

To answer, Figure 1 illustrates two key observations. First, key caches ($\mathbf{K}$) consistently have larger spectral norms than value caches ($\mathbf{V}$). Second, assigning more bits to keys (e.g., $\mathrm{K}_4\mathrm{V}_2$) improves accuracy across multiple models compared to key-underprovisioned allocations ($\mathrm{K}_2\mathrm{V}_4$). These motivate a deeper investigation into the distinct roles of key-value weights (i.e., $\mathbf{W^K}$ and $\mathbf{W^V}$) in attention mechanisms and their implications for quantization performance. Toward this, our contributions are:

- We propose the ***Key-Value Norm Disparity*** theorem, proving that the expected spectral and Frobenius norms of W^{K} predominantly exceed those of W^{V} across prominent LLM models (i.e., Llama3 and Mistral herds). We then derive the ***Key-Prioritized Quantization*** theorem, establishing the theoretical foundation on why assigning higher precision to K than V strictly reduces quantization error, enabling greater KV-cache compression while maintaining accuracy.
- We corroborate and operationalize these theorems across a diverse set of models (Llama-3.2-1B/3B/8B, Llama-3.3-70B, Phi-4-14B, Qwen3-0.6B/1.7B/4B/8B, DeepSeek-R1, Mistral-0.3-7B), datasets (C4, MMLU, GSM8K, EQ-Bench, CoQA, and LongBench[1]), and two quantization backends (Optimum Quanto and HQQ).

  Notably, $\mathrm{K}_4\mathrm{V}_2$ retains **98.3%** (1-shot) and **94.1%** (5-shot) accuracy of $\mathrm{K}_4\mathrm{V}_4$ accuracy while reducing KV-cache memory by **25%**, demonstrating both the theoretical soundness and practical effectiveness of the proposed strategy.
- Owing to its efficient one-off tunability, we show that our geometry-driven mixed-precision strategy is *orthogonal* to existing inference-time KV quantization methods and can be seamlessly integrated to yield synergistic gains. In a case study with rotation-based outlier redistribution, combining a key-prioritized quantization ($\mathrm{K}_4\mathrm{V}_2$) with *key-only rotation* outperforms $\mathrm{K}_4\mathrm{V}_4$ by **4.4-18%** in accuracy across tasks. In contrast, rotating value caches is unnecessary and sometimes detrimental

[1]We purposefully select generative tasks rather than commonsense reasoning tasks to better isolate quantization effects; see Section C.1 for details.

2 Background and Related Work

KV Quantization. Quantization methods for LLMs can be categorized by timing: *training-time* and *post-training* (PTQ) (Gholami et al., 2021). Training-time quantization integrates quantization into model training, typically achieving higher accuracy by quantizing weights or activations during the optimization process. However, it requires labeled data and incurs significant training overhead. PTQ applies quantization after training, avoiding retraining costs and labeled data requirements (Nagel et al., 2021), but sometimes yields lower accuracy.

PTQ can target different model components: weights, activations, or the KV cache. Weight-only quantization (Frantar et al., 2023a,b; Lin et al., 2024) achieves strong accuracy but does not reduce activation or KV memory. Weight-activation quantization (Dettmers et al., 2022; Xiao et al., 2023; Shao et al., 2024) reduces overall memory but often sacrifices accuracy. KV quantization offers the best of both: it targets the rapidly growing KV cache (Pope et al., 2023), providing activation-level memory savings while maintaining the accuracy of weight-only methods (Yue et al., 2024).

Existing KV Quantization Schemes. KV quantization methods can be categorized by how they treat keys and values (Li et al., 2025a):

- **Outlier redistribution.** These methods smooth or relocate outliers (i.e., unusually large activation values that dominate quantization ranges) in KV tensors, e.g., SmoothQuant (Xiao et al., 2023), AWQ (Lin et al., 2024), and OmniQuant (Shao et al., 2024).
- **Fixed-precision.** A single bit-width is used for both keys and values, ignoring their different roles and statistical properties (Yao et al., 2022; Sheng et al., 2023b; Liu et al., 2024).
- **Mixed-precision.** Different bit-widths are assigned to different parts of the cache (Hooper et al., 2024; Yue et al., 2024; Li et al., 2024; Dong et al., 2024; Duanmu et al., 2024; Zhang et al., 2024; Li et al., 2025b).

While mixed-precision schemes are the most flexible, existing methods have not systematically explored *asymmetric bit allocation between keys and values*. KVQuant (Hooper et al., 2024), for

example, focuses on vector-wise outlier handling rather than analyzing keys and values separately. Only a few methods have attempted to address this issue, and even then, only superficial insights have been gained.

Needs and Gaps. Despite rapid progress in KV quantization, several needs from the LLM research community remain unmet. **First**, there is a need for *principled strategies* to guide bit allocation. Current frameworks either treat the key-value split as a hyperparameter tuned through grid search (Li et al., 2025b) or rely on heuristics derived from cache statistics (e.g., activation ranges or distributions collected at inference time) (Zhang et al., 2023; Liu et al., 2025; Yang et al., 2024). These approaches are costly, model- or data-specific, and provide little theoretical insight into the inherent differences between keys and values. **Second**, there is a need to *understand and exploit key-value asymmetry*. Existing works such as KVTuner (Li et al., 2025b), SKVQ (Duanmu et al., 2024), and QAQ (Dong et al., 2024) briefly observe that allocating more bits to keys can preserve accuracy, but none explain *why* or propose a generalizable strategy. KVTuner reports differences in attention vs. perplexity errors across bit pairs without analysis; SKVQ finds asymmetric allocations (e.g., 2-bit keys, 1.5-bit values) through hyperparameter search rather than model structure; and QAQ focuses on different data types for keys and values rather than bit-width asymmetry. **Third**, there is a need for *lightweight, modular methods that integrate seamlessly with existing KV quantization frameworks*. The studies mentioned above often involve complex, runtime-dependent procedures that are difficult to generalize or combine, limiting their drop-in applicability and composability with other techniques.

Our Perspective. We address these needs by deriving bit allocation strategies *directly from model weights*, analyzing spectral and Frobenius norms of key and value matrices. Our approach is **lightweight**, incurring only a one-off analytical cost per model without any inference-time introspection; **generalizable**, since weight statistics are invariant across inputs and tasks; and **principled**, grounding allocation in linear algebraic properties rather than heuristic search. Moreover, our mixprecision quantization oracle is **orthogonal** to existing KV quantization techniques, paving a foundation that others can build upon. For example, pairing our mixed-precision allocation with rotation-based outlier redistribution techniques yields complementary improvements in both accuracy and memory savings. In this way, we elevate bit allocation from an ad hoc design choice to a geometry-informed building block for future KV quantization frameworks.

3 Norm Dynamics of KV Weights

We now establish a theoretical foundation for mixed-precision KV-cache quantization by analyzing the intrinsic geometry of the *key* and *value* projection weights. Our analysis proceeds in two steps. First, we prove that key weights systematically exhibit larger spectral and Frobenius norms than value weights (*Key-Value Norm Disparity*), a property that is preserved in the resulting key and value caches. Second, we demonstrate that this norm gap directly impacts quantization error, providing a formal guarantee that assigning higher bit precision to keys yields strictly lower distortion and higher inference accuracy than symmetric or inverted allocations (*Key-Prioritized Quantization*).

3.1 *Key-Value Norm Disparity*: Key Weights Dominate in Norm

The key weight matrix W^K maps hidden states into the key cache, whereas the value weight matrix W^V determines the representations retrieved during attention. Because quantization error scales approximately with the dynamic range of the signal, the relative magnitudes of $\|\mathrm{W}^K\|$ and $\|\mathrm{W}^V\|$, e.g., their Frobenius or spectral norms, indicate which cache is more sensitive to quantization.

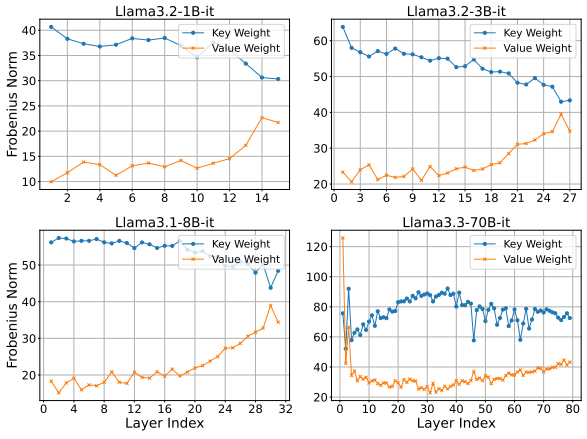


Figure 2: **Frobenius norms of key and value weight matrices across the Llama 3 family.** $\|W^K\|_F$ consistently exceeds $\|W^V\|_F$ across nearly all layers, with the exception occurring in early layers of the 70B variant.

Empirical measurements across four Llama-3

herds (Figure 2) show that the Frobenius norm of W^K consistently exceeds that of W^V in nearly every layer; the same ordering holds for spectral norms. This persistent gap motivates the following theorem.

Theorem 1 (*Key-Value Norm Disparity*)

Let W^K and W^V denote the key and value projection matrices in a Transformer. Then

$$\mathbb{E}\big[\|\mathrm{W}^K\|_F\big] > \mathbb{E}\big[\|\mathrm{W}^V\|_F\big].$$

The detailed proof is provided in Appendix A.2. The core idea here is to examine how the Frobenius norms of the key and value weight matrices evolve during training. The analysis begins with Xavier initialization (Glorot and Bengio, 2010), where all projection matrices have identical expected norms, and tracks their evolution under stochastic gradient descent (SGD). Intuitively, W^K play a dual role: they shape the attention map and determine what representations are stored in the cache. Each input is multiplied by W^K to produce keys that interact multiplicatively with the queries derived from W^Q (i.e., the query projection). As training proceeds, W^Q typically grows to sharpen attention, amplifying the gradient signals backpropagated into W^K. By contrast, W^V only influences post-attention representations, so its gradients lack this multiplicative amplification. This architectural asymmetry causes W^K to receive systematically larger updates, leading to persistently larger norms over time.

This phenomenon is ubiquitous. Appendix C.2 offers analogous results for Mistral family (Mistral AI, 2024), demonstrating the generality of the pattern. We next show that this norm disparity has direct consequences for quantization.

3.2 *Key-Prioritized Quantization*: Key-Favored Allocation Minimizes Quantization Error

Theorem 1 implies that, on average, W^K and their resulting activations, K, have larger magnitude than their value counterparts. Since quantization error under uniform scalar quantization scales with the signal energy, assigning equal bit precision to both is sub-optimal.

Consider an additive-residual Transformer block (layer normalization omitted for clarity (Elhage et al., 2021)):

$$h_{l+1} = h_l + W_l h_l.$$

Quantizing W_l to $\tilde{W}_l = W_l + \Delta_l$ introduces an error bounded by

$$\|h_{l+1} - \tilde{h}_{l+1}\|_2 = \|\Delta_l h_l\|_2 \le \|\Delta_l\|_2 \|h_l\|_2.$$

After L layers, the worst-case deviation accumulates multiplicatively:

$$\|h_L - \tilde{h}_L\|_2 \le \Big(\prod_{i=1}^{L} \|W_i\|_2\Big) \max_{1\le i\le L} \|\Delta_i\|_2.$$

Because the expected norms satisfy

$$\mathbb{E}\big[\|W^K\|_F^2\big] > \mathbb{E}\big[\|W^V\|_F^2\big],$$

any quantization noise injected along the key path is amplified more strongly through the network.

Let $X \in \mathbb{R}^{\text{seq}\times d_{\text{model}}}$ be the hidden-state matrix at a given layer. The same input generates both caches via

$$K = XW^K, \qquad V = XW^V.$$

Multiplying the previous inequality by X and applying sub-multiplicativity of the Frobenius norm yields

$$\mathbb{E}\big[\|K\|_F^2\big] > \mathbb{E}\big[\|V\|_F^2\big],$$

i.e., the norm gap persists in the caches.

Quantization Error and Norm Magnitude. For a matrix $M \in \mathbb{R}^{m\times n}$, the squared Frobenius norm equals the total signal energy and the sum of squared singular values. When quantized with b-bit uniform scalar quantization, the expected mean-square error satisfies

$$\mathbb{E}\big[\|M - \tilde{M}\|_F^2\big] = \Theta\big(\|M\|_F^2\, 2^{-2b}\big),$$

with constants depending only on the quantizer. Since key and value caches have identical shape, minimizing quantization error reduces to allocating bits in proportion to their energy. When $\|K\|_F \gg \|V\|_F$ and equal bit-widths are used, key-cache error dominates:

$$\mathbb{E}\big[\|K - \tilde{K}\|_F^2\big] \gg \mathbb{E}\big[\|V - \tilde{V}\|_F^2\big].$$

This asymmetry in quantization error directly motivates an asymmetric bit allocation strategy, formalized as the following theorem:

Theorem 2 (*Key-Prioritized Quantization*)

Let (b_K, b_V) denote the bit allocations for key and value caches under a uniform scalar quantizer. For any pair with $b_K > b_V$, the expected inference accuracy is strictly higher than for the swapped allocation (b_V, b_K), provided that $\mathbb{E}[\|K\|_F^2] > \mathbb{E}[\|V\|_F^2]$.

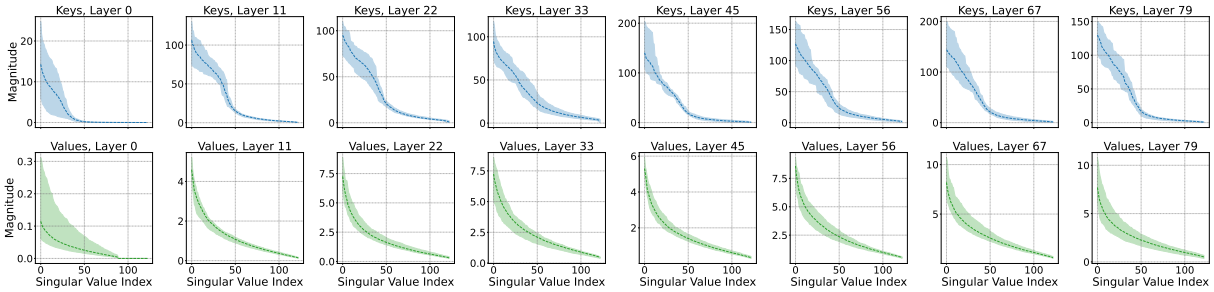


Figure 3: **Singular value spectra of key and value activations in Llama 3.3-70B on C4 benchmark dataset.** The x-axis shows singular value indices, ordered from the 5th largest onward for cleaner illustration, and the y-axis shows their magnitudes. Shaded regions mark the minimum-maximum range across attention heads within each layer, while dashed lines indicate the mean at each index. Beyond the top singular value (i.e., the spectral norm), key activations consistently exhibit larger singular values than value activations across the spectrum, highlighting their greater representational capacity. Full spectra are provided in Figure 6 of Appendix C.3.

Figure 3 provides empirical evidence for this effect. For Llama 3.3-70B on C4, the singular value spectra of the key caches consistently exceed those of the value caches beyond the top singular mode, indicating higher representational significance throughout the spectrum. Appendix C.3 extends this analysis to the full singular value range, where Figure 7 shows layer-wise Frobenius and spectral norms, all consistently revealing larger magnitudes for keys than for values.

The theoretical underpinnings of this phenomenon are established in Theorems 3 of Appendix B.2 and 4 of Appendix B.3, which derive a norm-dependent upper bound on the quantization error of an arbitrary matrix M. Appendix B.4 then applies this bound to the key and value caches, showing that the larger norms of the key caches translate into proportionally higher quantization errors under equal bit allocations.

4 Results

4.1 Experimental Setup

Three evaluations are conducted: *(i)* a quantization-error analysis, which measures reconstruction error across different bit-widths; *(ii)* a downstream task accuracy evaluation, which assesses how mixed-precision KV cache quantization affects model accuracy on practical benchmarks; and *(iii)* an integration case study with rotation-based outlier distribution methods, which investigates the effect of combining mixed precisions with rotation strategies. Full details on compute resources and software configurations are supplied in Appendix D.

Quantization-Error Evaluation. PyTorch's weight-packing quantization is applied without residual buffers or activation grouping to isolate quantization effects. Ten random sequences are sampled from C4 (Dodge et al., 2021), MMLU (Hendrycks et al., 2021), and GSM8K (Cobbe et al., 2021), padded to the longest length, and generated autoregressively for up to 1,000 tokens. Both per-layer and per-head reconstruction errors are computed, along with global averages across all heads, layers, and tokens, to offer a comprehensive view of bit-width sensitivity. This experiment spans seven models, including Llama-3.2-1B, Llama-3.1-8B (Grattafiori et al., 2024), Phi-4-14B (Abdin et al., 2024b), Mistral-0.3-7B (Mistral AI, 2024), Qwen-2.5-14B, Llama-3.3-70B, DeepSeek-R1-Qwen-14B, Phi-3-Medium-128K (Abdin et al., 2024a), and Llama-3.1-Nemotron-70B (Nvidia et al., 2024).

Downstream Task Accuracy. Two representative quantization backends are employed. *Optimum Quanto* applies token-wise (per-row) quantization with mixed 2/4-bit precision on Llama-3.2-1B, Llama-3.1-8B, Phi-4-14B, and DeepSeek-R1-Qwen-14B. *HQQ* (Badri and Shaji, 2023) applies channel-wise (per-column) quantization with bit-widths $\{1, 2, 4, 6, 8\}$ on Llama-3.1-8B, Llama-3.2-1B, Llama-3.2-3B, and Qwen3-0.6B/1.7B/4B/8B (Qwen Team, 2025). A 64-token residual buffer (which stores the most recent tokens in full precision before being periodically flushed) and 128-element activation grouping (which quantizes activations in fixed-size blocks to reduce overhead) are adopted to mirror practical decoding configurations used in KIVI (Liu et al., 2024), Flash-Decoding (Dao et al., 2023), and Marlin (Frantar et al., 2025), ensuring that the evaluation reflects realistic deployment rather

Table 1: **Quantization error (lower is better) of K and V caches under matched bit-widths.** Each cell reports the mean ± standard deviation of the reconstruction MSE between the *dequantized* cache and its BF16 reference, averaged over layers, heads, and tokens (10 sequences per dataset up to 1,000 tokens each). Both caches are quantized to the *same* precision, i.e., $\mathbf{K}_2\mathbf{V}_2$, $\mathbf{K}_3\mathbf{V}_3$, and $\mathbf{K}_4\mathbf{V}_4$, to isolate their intrinsic sensitivity at equal bit budgets. Across model families (Llama, Phi, Mistral, Qwen, DeepSeek) and datasets (C4, MMLU, GSM8K), $\mathbf{K}$ consistently exhibits larger reconstruction error than $\mathbf{V}$ at the same bit-width, and the error decreases monotonically with increasing precision, indicating that keys are the dominant source of quantization distortion.

Dataset	Model	2-bit		3-bit		4-bit	
		$\mathbf{K}_2$	$\mathbf{V}_2$	$\mathbf{K}_3$	$\mathbf{V}_3$	$\mathbf{K}_4$	$\mathbf{V}_4$
MMLU	Llama3.2-1B	4.851 ± 1.037	0.127 ± 0.101	1.037 ± 0.265	0.021 ± 0.015	0.227 ± 0.059	0.005 ± 0.003
	Llama3.1-8B-it	6.003 ± 1.782	0.187 ± 0.127	1.082 ± 0.244	0.028 ± 0.019	0.235 ± 0.055	0.006 ± 0.004
	Llama3.3-70B-it	4.883 ± 1.106	0.112 ± 0.093	0.942 ± 0.198	0.016 ± 0.012	0.206 ± 0.043	0.003 ± 0.003
	Phi4	5.929 ± 1.545	0.657 ± 0.472	1.306 ± 0.231	0.103 ± 0.070	0.286 ± 0.050	0.022 ± 0.015
	Mistral0.3-7B	4.718 ± 1.340	0.398 ± 0.405	0.941 ± 0.240	0.059 ± 0.059	0.206 ± 0.053	0.013 ± 0.013
	Qwen2.5-14B	5.184 ± 2.241	1.270 ± 1.547	1.005 ± 0.288	0.182 ± 0.221	0.223 ± 0.067	0.040 ± 0.052
	R1Q-14B	5.126 ± 2.375	1.406 ± 1.609	0.900 ± 0.269	0.198 ± 0.226	0.199 ± 0.062	0.044 ± 0.052
C4	Llama3.2-1B	4.885 ± 1.056	0.207 ± 0.166	1.074 ± 0.289	0.030 ± 0.024	0.233 ± 0.062	0.006 ± 0.005
	Llama3.1-8B-it	6.262 ± 1.789	0.254 ± 0.185	1.128 ± 0.249	0.036 ± 0.026	0.247 ± 0.056	0.008 ± 0.005
	Llama3.3-70B-it	4.391 ± 1.027	0.121 ± 0.097	0.847 ± 0.175	0.017 ± 0.013	0.186 ± 0.038	0.004 ± 0.003
	Phi4	5.715 ± 1.442	0.850 ± 0.684	1.316 ± 0.245	0.124 ± 0.093	0.291 ± 0.056	0.027 ± 0.020
	Mistral0.3-7B	5.027 ± 1.332	0.543 ± 0.493	1.014 ± 0.269	0.079 ± 0.068	0.223 ± 0.060	0.017 ± 0.015
	Qwen2.5-14B	4.382 ± 2.170	1.544 ± 1.872	0.846 ± 0.250	0.220 ± 0.265	0.187 ± 0.060	0.048 ± 0.060
	DeepSeekR1Q-14B	4.832 ± 2.354	1.651 ± 1.914	0.927 ± 0.283	0.232 ± 0.267	0.201 ± 0.061	0.051 ± 0.060
GSM8K	Llama3.2-1B	5.703 ± 1.557	0.179 ± 0.136	1.213 ± 0.352	0.026 ± 0.020	0.266 ± 0.078	0.005 ± 0.004
	Llama3.1-8B-it	6.445 ± 1.837	0.213 ± 0.161	1.184 ± 0.268	0.030 ± 0.022	0.257 ± 0.060	0.007 ± 0.005
	Llama3.3-70B-it	4.967 ± 1.127	0.113 ± 0.091	0.978 ± 0.203	0.016 ± 0.012	0.214 ± 0.044	0.004 ± 0.003
	Phi4	6.610 ± 1.624	0.785 ± 0.598	1.498 ± 0.293	0.116 ± 0.082	0.330 ± 0.064	0.025 ± 0.017
	Mistral0.3-7B	5.308 ± 1.367	0.461 ± 0.434	1.065 ± 0.288	0.067 ± 0.061	0.232 ± 0.061	0.015 ± 0.013
	Qwen2.5-14B	4.829 ± 2.179	1.736 ± 2.659	0.979 ± 0.264	0.241 ± 0.372	0.214 ± 0.061	0.051 ± 0.077
	DeepSeekR1Q-14B	4.477 ± 2.176	1.424 ± 1.752	0.830 ± 0.256	0.200 ± 0.242	0.181 ± 0.058	0.044 ± 0.056

than idealized settings. Accuracy is measured on three generative benchmarks: GSM8K (Cobbe et al., 2021), CoQA (Reddy et al., 2019), and EQ-BENCH (Paech, 2024), which collectively probe mathematical reasoning, conversational QA, and structured long-form generation.

Integration with Rotation-Based Methods. QuaRot (Ashkboos et al., 2024) is selected for this case study. It applies structured randomized Hadamard rotations to activations before quantization, effectively dispersing outliers and improving the uniformity of the quantization distribution. A three-dimensional design space is explored, spanning *bit-width allocation*, *group size configuration*, and *rotation strategies*. Specifically, mixed-precision settings $\mathrm{K}_2\mathrm{V}_2$, $\mathrm{K}_2\mathrm{V}_4$, $\mathrm{K}_4\mathrm{V}_2$, and $\mathrm{K}_4\mathrm{V}_4$; key and value group sizes in $\{32, 64, 128\}$; and four rotation strategies (no rotation, key-only, value-only, and both). The evaluation encompasses generative tasks including CoQA, GSM8K, EQ-BENCH, and LONGBENCH, utilizing the same seven models as in quantization-error evaluation.

4.2 Mixed-Precision Quantization Error

Table 1 reports reconstruction errors at 2-, 3-, and 4-bit precision for seven representative models spanning multiple model families (Llama, Phi, Mistral, Qwen, DeepSeek) and datasets; full results appear in Appendix C.4. Figure 8 shows the complete error curves for Llama-3.3-70B on C4, and Figure 9 provides a per-layer breakdown for Llama-3.1-8B. Across all models, datasets, and bit-widths, key caches consistently incur larger reconstruction errors than value caches, and this gap remains stable across precision levels. These findings empirically support the theoretical prediction that key representations have higher energy and are therefore more sensitive to quantization.

4.3 Mixed-Precision Downstream Accuracy

Table 2 summarizes the Optimum Quanto results on GSM8K under both 1-shot and 8-shot Chain-of-Thought (CoT) prompting. Although CoT prompting can sometimes reduce reasoning accuracy, it is included here to assess the impact of longer contexts on quantized decoding. Across four representative models, including Llama-3.2-1B, Llama-3.1-8B, Phi-4-14B, and DeepSeek-R1-Qwen-14B, a consistent pattern emerges: allocating higher precision to the *key* cache ($\mathrm{K}_4\mathrm{V}_2$) preserves accuracy substantially better than the inverse ($\mathrm{K}_2\mathrm{V}_4$). On average, $\mathrm{K}_4\mathrm{V}_2$ recovers approximately **94%** of the

full-precision baseline, whereas value-favored allocations incur losses of up to 30 percentage points (***pp***). The performance gap widens with model scale; for instance, under 1-shot GSM8K, the $\mathrm{K}_4\mathrm{V}_2$ configuration outperforms $\mathrm{K}_2\mathrm{V}_4$ by 30 pp on Llama-3.2-1B and by 16 pp on Phi-4-14B. Notably, $\mathrm{K}_4\mathrm{V}_2$ nearly matches the symmetric $\mathrm{K}_4\mathrm{V}_4$ baseline despite halving the value bit budget, indicating that downstream performance is primarily constrained by key precision.

Table 2: **GSM8K – Downstream accuracy with Optimum Quanto (token-wise) mixed-precision KV quantization.** Token-wise quantization is applied with supported precisions $i, j \in \{2, 4\}$, where $\mathrm{K}_i\mathrm{V}_j$ denotes i-bit keys and j-bit values. Results are shown for both 1-shot and 8-shot settings. Across models, the key-favored $\mathrm{K}_4\mathrm{V}_2$ consistently outperforms the value-favored $\mathrm{K}_2\mathrm{V}_4$ and approaches the $\mathrm{K}_4\mathrm{V}_4$ baseline, demonstrating the benefits of prioritizing key precision.

Model	Shots	$\mathbf{K}_2\mathbf{V}_2$	$\mathbf{K}_2\mathbf{V}_4$	$\mathbf{K}_4\mathbf{V}_2$	$\mathbf{K}_4\mathbf{V}_4$
Llama 3.2-1B-it	1-shot	0.033	0.035	0.338	0.357
	8-shot	0.031	0.031	0.289	0.369
Llama 3.1-8B-it	1-shot	0.511	0.547	0.752	0.754
	8-shot	0.408	0.441	0.770	0.782
Phi 4-14B	1-shot	0.759	0.783	0.913	0.923
	8-shot	0.771	0.815	0.927	0.931
DeepSeek R1Q-14B	1-shot	0.772	0.775	0.865	0.867
	8-shot	0.763	0.792	0.876	0.875

The HQQ results, shown in Table 3, extend these observations to a broader range of bit-widths, model scales, and tasks. The analysis systematically compares $\mathbf{K}_i\mathbf{V}_x$ against $\mathbf{K}_x\mathbf{V}_i$ for $i \in \{1, 2, 4, 6, 8\}$, where x denotes the mean accuracy over all bit-widths of the other cache. This directly addresses the question: "If i bits are available, should they be allocated to keys or values?" Across all models (0.6B-32B) and datasets (GSM8K, CoQA, EQ-Parseable), the answer is consistently "keys". At ultra-low precision (1-2 bits), the advantage is especially pronounced; for instance, on GSM8K, allocating a single extra bit to keys with 1-bit quantization yields gains of +48 pp for Qwen3-8B. Similar trends hold on EQ-Bench and CoQA: for Qwen3-0.6B on EQ-Parseable, prioritizing keys delivers up to +62 pp, and key-first allocations never underperform value-first ones in any configuration. Even at moderate precisions (4-6 bits), key-centric allocation continues to offer 7-12 pp improvements for 8-14B models, indicating that the advantage persists well beyond extreme compression. **On average, $\mathbf{K}_4\mathbf{V}_2$ retains 98.3% accuracy of $\mathbf{K}_4\mathbf{V}_4$** (CoQA: 99.2%, EQ-Bench: 99.35%, GSM8K: 97.7%; worst: 88.3%, best: 103.5%).

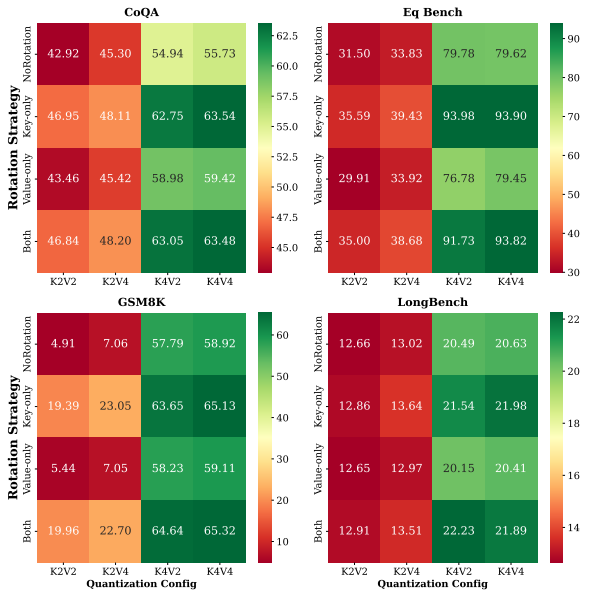


Figure 4: **Integration of rotation and mixed-precision quantization.** Downstream accuracy is shown for four quantization configurations ($\mathrm{K}_2\mathrm{V}_2$, $\mathrm{K}_2\mathrm{V}_4$, $\mathrm{K}_4\mathrm{V}_2$, $\mathrm{K}_4\mathrm{V}_4$) combined with four rotation strategies (none, key-only, value-only, both), using a fixed group size of 64 for both keys and values. Results are reported on CoQA, GSM8K, EQ-BENCH, and LONGBENCH, enabling a controlled comparison of precision-rotation interactions.

More detailed downstream accuracy results are provided in Appendix C.5. Overall, downstream accuracy is far more sensitive to key precision than to value precision. Across both token-wise and channel-wise quantization schemes, model scales, and task types, assigning the higher bit-width to K consistently yields near-baseline accuracy while substantially reducing KV memory. This establishes a simple, backend-agnostic design principle for mixed-precision KV cache quantization: *More for keys, less for values*.

4.4 Integrating with Rotation-Based Methods

Figure 4 visualizes how rotation interacts with key-value bit allocations. Across all tasks, applying rotation to *keys* consistently yields larger gains than applying it to values. Notably, applying key-only rotation to the $\mathrm{K}_4\mathrm{V}_2$ configuration achieves accuracy that closely matches the full $\mathrm{K}_4\mathrm{V}_4$ baseline, indicating that the primary benefits of rotation arise from mitigating key outliers. These trends are consistent across tasks and model scales, reinforcing

Table 3: **Downstream accuracy with HQQ (channel-wise) mixed-precision KV quantization across GSM8K, EQ-Parseable, and CoQA.** Each cell compares $\mathbf{K}_i\mathbf{V}_x$ and $\mathbf{K}_x\mathbf{V}_i$ for $i \in \{1, 2, 4, 6, 8\}$, where x represents the mean accuracy averaged over the other cache's bit-widths $B = \{1, 2, 4, 6, 8\}$. This corresponds to contrasting "allocate i bits to keys (values averaged over B)" versus "allocate i bits to values (keys averaged over B)." Across model scales from 0.6B to 32B and multiple tasks, key-favored allocations consistently yield higher accuracy, with more pronounced gains observed at lower precision and persistent benefits at higher precision, demonstrating that the key-first advantage generalizes beyond a single model, task, and quantization backend.

GSM8K	$\mathbf{K}_1\mathbf{V}_x$ vs. $\mathbf{K}_x\mathbf{V}_1$	$\mathbf{K}_2\mathbf{V}_x$ vs. $\mathbf{K}_x\mathbf{V}_2$	$\mathbf{K}_4\mathbf{V}_x$ vs. $\mathbf{K}_x\mathbf{V}_4$	$\mathbf{K}_6\mathbf{V}_x$ vs. $\mathbf{K}_x\mathbf{V}_6$	$\mathbf{K}_8\mathbf{V}_x$ vs. $\mathbf{K}_x\mathbf{V}_8$
Qwen3-0.6B	$0.00 \overset{+3\%}{<} \mathbf{0.03}$	$0.00 \overset{+16\%}{<} \mathbf{0.16}$	$0.04 \overset{+13\%}{<} \mathbf{0.17}$	$\mathbf{0.34} \overset{+16\%}{>} 0.18$	$\mathbf{0.34} \overset{+16\%}{>} 0.18$
Llama-3.2-1B	$0.00 \overset{+5\%}{<} \mathbf{0.05}$	$0.01 \overset{+18\%}{<} \mathbf{0.19}$	$\mathbf{0.27} \overset{+7\%}{>} 0.20$	$\mathbf{0.28} \overset{+8\%}{>} 0.20$	$\mathbf{0.28} \overset{+7\%}{>} 0.21$
Llama-3.2-3B	$0.00 \overset{+19\%}{<} \mathbf{0.19}$	$0.27 \overset{+17\%}{<} \mathbf{0.44}$	$\mathbf{0.57} \overset{+11\%}{>} 0.46$	$\mathbf{0.58} \overset{+12\%}{>} 0.46$	$\mathbf{0.59} \overset{+13\%}{>} 0.46$
Qwen3-4B	$0.00 \overset{+41\%}{<} \mathbf{0.41}$	$0.01 \overset{+49\%}{<} \mathbf{0.50}$	$\mathbf{0.80} \overset{+29\%}{>} 0.51$	$\mathbf{0.82} \overset{+31\%}{>} 0.51$	$\mathbf{0.82} \overset{+31\%}{>} 0.51$
Llama-3.1-8B	$0.00 \overset{+31\%}{<} \mathbf{0.31}$	$0.35 \overset{+17\%}{<} \mathbf{0.52}$	$\mathbf{0.70} \overset{+16\%}{>} 0.54$	$\mathbf{0.70} \overset{+16\%}{>} 0.54$	$\mathbf{0.70} \overset{+16\%}{>} 0.54$
Qwen3-8B	$0.00 \overset{+48\%}{<} \mathbf{0.48}$	$0.08 \overset{+46\%}{<} \mathbf{0.54}$	$\mathbf{0.86} \overset{+31\%}{>} 0.55$	$\mathbf{0.87} \overset{+32\%}{>} 0.55$	$\mathbf{0.86} \overset{+31\%}{>} 0.55$
Qwen3-32B	$0.00 \overset{+42\%}{<} \mathbf{0.42}$	$0.25 \overset{+23\%}{<} \mathbf{0.48}$	$\mathbf{0.73} \overset{+23\%}{>} 0.50$	$\mathbf{0.71} \overset{+21\%}{>} 0.50$	$\mathbf{0.72} \overset{+21\%}{>} 0.51$
EQ-Parseable					
Qwen3-0.6B	$0.00 \overset{+10\%}{<} \mathbf{0.10}$	$0.00 \overset{+53\%}{<} \mathbf{0.53}$	$0.51 \overset{+2\%}{<} \mathbf{0.53}$	$\mathbf{0.84} \overset{+32\%}{>} 0.53$	$\mathbf{0.85} \overset{+33\%}{>} 0.52$
Llama-3.2-1B	$0.00 \overset{+34\%}{<} \mathbf{0.34}$	$0.26 \overset{+37\%}{<} \mathbf{0.62}$	$\mathbf{0.87} \overset{+22\%}{>} 0.65$	$\mathbf{0.90} \overset{+24\%}{>} 0.66$	$\mathbf{0.90} \overset{+25\%}{>} 0.66$
Llama-3.2-3B	$0.00 \overset{+36\%}{<} \mathbf{0.35}$	$0.68 \overset{+5\%}{<} \mathbf{0.73}$	$\mathbf{0.90} \overset{+14\%}{>} 0.76$	$\mathbf{0.89} \overset{+13\%}{>} 0.76$	$\mathbf{0.90} \overset{+14\%}{>} 0.76$
Qwen3-4B	$0.00 \overset{+54\%}{<} \mathbf{0.48}$	$0.29 \overset{+33\%}{<} \mathbf{0.58}$	$\mathbf{0.84} \overset{+28\%}{>} 0.60$	$\mathbf{0.86} \overset{+31\%}{>} 0.59$	$\mathbf{0.85} \overset{+29\%}{>} 0.60$
Llama-3.1-8B	$0.00 \overset{+61\%}{<} \mathbf{0.61}$	$\mathbf{0.78} \overset{+2\%}{>} 0.76$	$\mathbf{0.96} \overset{+19\%}{>} 0.77$	$\mathbf{0.97} \overset{+20\%}{>} 0.77$	$\mathbf{0.97} \overset{+21\%}{>} 0.77$
Qwen3-8B	$0.00 \overset{+62\%}{<} \mathbf{0.58}$	$0.54 \overset{+17\%}{<} \mathbf{0.70}$	$\mathbf{0.95} \overset{+25\%}{>} 0.71$	$\mathbf{0.96} \overset{+27\%}{>} 0.71$	$\mathbf{0.96} \overset{+26\%}{>} 0.71$
Qwen3-32B	$0.00 \overset{+60\%}{<} \mathbf{0.47}$	$0.56 \overset{+7\%}{<} \mathbf{0.62}$	$\mathbf{0.80} \overset{+22\%}{>} 0.62$	$\mathbf{0.80} \overset{+23\%}{>} 0.62$	$\mathbf{0.80} \overset{+22\%}{>} 0.63$
CoQA					
Qwen3-0.6B	$0.20 \overset{+33\%}{<} \mathbf{0.44}$	$0.32 \overset{+29\%}{<} \mathbf{0.52}$	$\mathbf{0.65} \overset{+16\%}{>} 0.53$	$\mathbf{0.69} \overset{+23\%}{>} 0.53$	$\mathbf{0.69} \overset{+23\%}{>} 0.53$
Llama-3.2-1B	$0.22 \overset{+29\%}{<} \mathbf{0.42}$	$0.48 \overset{+13\%}{<} \mathbf{0.57}$	$\mathbf{0.67} \overset{+13\%}{>} 0.58$	$\mathbf{0.68} \overset{+14\%}{>} 0.58$	$\mathbf{0.68} \overset{+14\%}{>} 0.58$
Llama-3.2-3B	$0.21 \overset{+50\%}{<} \mathbf{0.60}$	$\mathbf{0.73} \overset{+6\%}{>} 0.68$	$\mathbf{0.79} \overset{+14\%}{>} 0.68$	$\mathbf{0.79} \overset{+14\%}{>} 0.68$	$\mathbf{0.79} \overset{+15\%}{>} 0.67$
Qwen3-4B	$0.34 \overset{+34\%}{<} \mathbf{0.62}$	$0.68 \overset{+3\%}{<} \mathbf{0.70}$	$\mathbf{0.81} \overset{+12\%}{>} 0.71$	$\mathbf{0.81} \overset{+12\%}{>} 0.71$	$\mathbf{0.81} \overset{+13\%}{>} 0.70$
Llama-3.1-8B	$0.21 \overset{+50\%}{<} \mathbf{0.60}$	$\mathbf{0.72} \overset{+7\%}{>} 0.66$	$\mathbf{0.78} \overset{+14\%}{>} 0.67$	$\mathbf{0.78} \overset{+15\%}{>} 0.67$	$\mathbf{0.78} \overset{+14\%}{>} 0.67$
Qwen3-8B	$0.38 \overset{+37\%}{<} \mathbf{0.68}$	$\mathbf{0.73} \overset{+1\%}{>} 0.72$	$\mathbf{0.82} \overset{+12\%}{>} 0.72$	$\mathbf{0.82} \overset{+12\%}{>} 0.72$	$\mathbf{0.82} \overset{+12\%}{>} 0.72$
Qwen3-32B	$0.32 \overset{+44\%}{<} \mathbf{0.68}$	$\mathbf{0.79} \overset{+9\%}{>} 0.72$	$\mathbf{0.82} \overset{+12\%}{>} 0.73$	$\mathbf{0.82} \overset{+12\%}{>} 0.73$	$\mathbf{0.82} \overset{+12\%}{>} 0.73$

that rotation is most effective when applied in conjunction with key-favored bit allocation.

Appendix C.6 presents detailed downstream results illustrating the synergistic effects of integrating rotation with mixed-precision quantization across tasks and models. It also examines the impact of key and value group sizes, showing that smaller sizes are beneficial for K due to higher information density, whereas larger group sizes suffice for V given lower sensitivity to quantization.

5 Conclusion

As large language models increasingly devote most of their inference cost to KV-cache storage and access, effective cache compression has become essential for practical deployment. This work provides a theoretically grounded justification and solution to the bit-allocation problem by bridging model geometry and quantization design. We theoretically establish that key projections consistently carry higher information density than value projections. Building on this, we show that allocating higher precision to keys and lower precision to values minimizes quantization error. Extensive experiments across nine model families, six benchmarks, and two hardware-aligned backends validate this principle: a $\mathrm{K}_4\mathrm{V}_2$ precision split reliably recovers up to 98.3% of $\mathrm{K}_4\mathrm{V}_4$ accuracy while significantly reducing memory consumption. Moreover, we demonstrate that our geometry-driven strategy is *orthogonal* to rotation-based outlier redistribution methods, enabling seamless integration and further accuracy gains. These findings elevate bit allocation from empirical tuning to a theoretically grounded geometry-driven design principle, providing clear guidance for efficient deployment and future hardware-algorithm co-design.

Limitations

While our geometry-driven mixed-precision quantization framework demonstrates both theoretical soundness and practical effectiveness through rigorous analysis and extensive experiments, several limitations remain. First, all evaluations are conducted with a maximum context length of 2,000 tokens, which reflects common inference-time configurations but does not fully capture the behavior of models operating at much larger context windows. Extending the approach to longer contexts may expose additional challenges, including increased quantization sensitivity and higher memory management overheads. Addressing these factors is an important direction for future work.

Ethical Considerations

This research aims to reduce the memory and computational costs of large language model inference by utilizing efficient KV-cache quantization. Such improvements have the potential to lower energy consumption and broaden access to language models in resource-constrained settings, promoting more sustainable and inclusive deployment. However, any compression technique entails accuracy trade-offs, which must be carefully monitored to avoid disproportionate impacts in high-stakes domains such as healthcare, law, or finance. Responsible deployment requires thorough evaluation of model behavior under mixed-precision settings, particularly for safety-critical applications.

Acknowledgment

This research was supported in part by NSF awards 2117439, 2112606, and 2320952.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024a. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024b. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

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

Hicham Badri and Appu Shaji. 2023. Half-quadratic quantization of large machine learning models.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*. Vancouver, Canada.

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. *Preprint*, arXiv:2110.14168.

Tri Dao. 2024. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*.

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)*.

Tri Dao, Daniel Haziza, Francisco Massa, and Grigory Sizov. 2023. Flash-decoding for long-context inference. Accessed: 2025-05-15.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025b. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

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

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. *Preprint*, arXiv:2104.08758.

Shichen Dong, Wen Cheng, Jiayu Qin, and Wei Wang. 2024. Qaq: Quality adaptive quantization for llm kv cache. *Preprint*, arXiv:2403.04643.

Haojie Duanmu, Zhihang Yuan, Xiuhong Li, Jiangfei Duan, Xingcheng Zhang, and Dahua Lin. 2024. Skvq: Sliding-window key and value cache quantization for large language models. In *Proceedings of COLM 2024*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, and 1 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023a. Gptq: Accurate post-training quantization for generative pre-trained transformers. In *Proceedings of the International Conference on Learning Representations (ICLR 2023)*.

Elias Frantar, Roberto L. Castro, Jiale Chen, Torsten Hoefler, and Dan Alistarh. 2025. Marlin: Mixed-precision auto-regressive parallel inference on large language models. In *Proceedings of the 30th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming*, PPoPP '25, page 239–251, New York, NY, USA. Association for Computing Machinery.

Elias Frantar, Sidak Pal Singh, and Dan Alistarh. 2023b. Optimal brain compression: A framework for accurate post-training quantization and pruning. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2021. A survey of quantization methods for efficient neural network inference. *Preprint*, arXiv:2103.13630.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Google DeepMind. 2025. Gemini 2.0 flash. Accessed: 2025-05-15.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. 2022. Accelerate: Training and inference at scale made simple, efficient and adaptable.

Song Han, Huizi Mao, and William J. Dally. 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *Proceedings of the International Conference on Learning Representations (ICLR 2016)*. Oral presentation.

Mohsen Hariri, Amirhossein Samandar, Michael Hinczewski, and Vipin Chaudhary. 2025. Don't pass@k: A bayesian framework for large language model evaluation. *arXiv preprint arXiv:2510.04265*.

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

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. 2024. Kvquant: Towards 10 million context length llm inference with kv cache quantization. In *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024)*.

Haoyang Li, Yiming Li, Anxin Tian, Tianhao Tang, Zhanchao Xu, Xuejia Chen, Nicole Hu, Wei Dong, Qing Li, and Lei Chen. 2025a. A survey on large language model acceleration based on kv cache management. *Preprint*, arXiv:2412.19442.

Xing Li, Zeyu Xing, Yiming Li, Linping Qu, Hui-Ling Zhen, Wulong Liu, Yiwu Yao, Sinno Jialin Pan, and Mingxuan Yuan. 2025b. Kvtuner: Sensitivity-aware layer-wise mixed precision kv cache quantization for efficient and nearly lossless llm inference. *Preprint*, arXiv:2502.04420.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024. Snapkv: Llm knows what you are looking for before generation. In *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024)*.

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 llm compression and acceleration. In *Proceedings of the 7th MLSys Conference 2024*, Santa Clara, CA, USA.

Tengxuan Liu, Shiyao Li, Jiayi Yang, Tianchen Zhao, Feng Zhou, Xiaohui Song, Guohao Dai, Shengen Yan, Huazhong Yang, and Yu Wang. 2025. Pm-kvq: Progressive mixed-precision kv cache quantization for long-cot llms. *Preprint*, arXiv:2505.18610.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, PMLR 235, Vienna, Austria.

Meta AI. 2025a. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation. `https://ai.meta.com/blog/llama-4-multimodal-intelligence/`. Official announcement of Llama 4. Accessed 2025-10-06.

Meta AI. 2025b. The llama 4 herd: The beginning of a new era of natively multimodal intelligence. Accessed: 2025-05-15.

Mistral AI. 2024. Large enough: Announcing mistral large 2. Accessed: 2025-05-15.

Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. 2021. A white paper on neural network quantization. *Preprint*, arXiv:2106.08295.

Nvidia, :, Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H. Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, Sirshak Das, Ayush Dattagupta, Olivier Delalleau, Leon Derczynski, Yi Dong, Daniel Egert, Ellie Evans, and 64 others. 2024. Nemotron-4 340b technical report. *Preprint*, arXiv:2406.11704.

OpenAI. 2024. Openai o1 system card. Accessed via OpenAI.

OpenAI. 2025. Openai o3-mini. Accessed via OpenAI.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Samuel J. Paech. 2024. Eq-bench: An emotional intelligence benchmark for large language models. *Preprint*, arXiv:2312.06281.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. 2023. Efficiently scaling transformer inference. In *Proceedings of the 6th MLSys Conference 2023*, Miami Beach, FL, USA. Copyright 2023 by the author(s).

Qwen Team. 2025. Qwen3: Think deeper, act faster. Accessed: 2025-05-15.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. Accessed via OpenAI.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Accessed via OpenAI.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge. *Preprint*, arXiv:1808.07042.

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 *Proceedings of the International Conference on Learning Representations (ICLR 2024)*.

Ying Sheng, Lianmin Zheng, Binhang Yuan, Zhuohan Li, Max Ryabinin, Beidi Chen, Percy Liang, Christopher Re, Ion Stoica, and Ce Zhang. 2023a. FlexGen: High-throughput generative inference of large language models with a single GPU. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 31094–31116. PMLR.

Ying Sheng, Lianmin Zheng, Binhang Yuan, Zhuohan Li, Max Ryabinin, Daniel Y. Fu, Zhiqiang Xie, Beidi Chen, Clark Barrett, Joseph E. Gonzalez, Percy Liang, Christopher Ré, Ion Stoica, and Ce Zhang. 2023b. Flexgen: High-throughput generative inference of large language models with a single gpu. In *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*, PMLR 202, Honolulu, Hawaii, USA. Copyright 2023 by the author(s).

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th Conference on Neural Information Processing Systems (NeurIPS 2014)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS 2017)*, Long Beach, CA, USA.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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 (ICML 2023)*, PMLR 202, Honolulu, Hawaii, USA.

June Yong Yang, Byeongwook Kim, Jeongin Bae, Beomseok Kwon, Gunho Park, Eunho Yang, Se Jung Kwon, and Dongsoo Lee. 2024. No token left behind: Reliable kv cache compression via importance-aware mixed precision quantization. *Preprint*, arXiv:2402.18096.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. 2022. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*.

Yuxuan Yue, Zhihang Yuan, Haojie Duanmu, Sifan Zhou, Jianlong Wu, and Liqiang Nie. 2024. Wkvquant: Quantizing weight and key/value cache for large language models gains more. *Preprint*, arXiv:2402.12065.

Hongxuan Zhang, Yao Zhao, Jiaqi Zheng, Chenyi Zhuang, Jinjie Gu, and Guihai Chen. 2024. Csr:achieving 1 bit key-value cache via sparse representation. *Preprint*, arXiv:2412.11741.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Re, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Appendix

A Dynamics of the Norms of Key and Value Weight Matrices

A.1 Preliminaries and Notation

c	Sequence length
d_m	Embedding dimension
$d_v = d_k = d_m/n_h$	Single head dimension
n_h	Number of attention heads

X	Input matrix $(c \times d_m)$
X^O	Output matrix $(c \times d_m)$
W^V	Value weights $(d_m \times d_m)$
W^K	Key weights $(d_m \times d_m)$
W^Q	Query weights $(d_m \times d_m)$
$K = XW^K$	$(c \times d_m)$
$Q = XW^Q$	$(c \times d_m)$
$V = XW^V$	$(c \times d_m)$
$S = QK^\top/\sqrt{d_m}$	$(c \times c)$
$A = \mathrm{softmax}(S)$	$(c \times c)$
$\mathcal{L}$	loss function.

A.2 Training Dynamics of Frobenius Norms

We compare the long-time behavior of the Frobenius norms of the key (W^K) and value (W^V) weight matrices of a single-head self-attention layer trained with stochastic gradient descent (SGD). We show that, under standard isotropic assumptions, $\|W^K\|_F$ grows faster than $\|W^V\|_F$.

Update rule. At step i, the weights follow

$$W_{i+1}^m \;=\; W_i^m - \eta\,\frac{\partial \mathcal{L}}{\partial W_i^m}, \qquad m \in \{K, V\},$$

with learning rate η. Squaring the Frobenius norm of both sides gives

$$\|W_{i+1}^m\|_F^2 = \|W_i^m\|_F^2 + \eta^2 \Big\|\frac{\partial \mathcal{L}}{\partial W_i^m}\Big\|_F^2 - 2\eta \left\langle W_i^m, \frac{\partial \mathcal{L}}{\partial W_i^m}\right\rangle_F.$$

where

$$\|A\|_F \equiv \sqrt{\mathrm{tr}(A^\top A)},$$
$$\langle A, B\rangle_F \equiv \mathrm{tr}(A^\top B).$$

Expectation over mini-batches. Taking an expectation over mini-batches yields

$$\Delta\mathbb{E}\big[\|W_i^m\|_F^2\big] \equiv \mathbb{E}\big[\|W_{i+1}^m\|_F^2\big] - \mathbb{E}\big[\|W_i^m\|_F^2\big] = \eta^2\,\mathbb{E}\Big[\Big\|\frac{\partial \mathcal{L}}{\partial W_i^m}\Big\|_F^2\Big] - 2\eta\,\mathbb{E}\Big[\Big\langle W_i^m, \frac{\partial \mathcal{L}}{\partial W_i^m}\Big\rangle_F\Big].$$

In high-dimensional weight space, the gradient is almost orthogonal to the current weights, making the second term negligible. Hence,

$$\Delta\mathbb{E}\big[\|W_i^m\|_F^2\big] \;\approx\; \eta^2\,\mathbb{E}\Big[\Big\|\frac{\partial \mathcal{L}}{\partial W_i^m}\Big\|_F^2\Big].$$

Throughout, we assume Xavier initialization: each entry of W^K and W^V is drawn i.i.d. from a zero-mean distribution whose variance preserves the input scale (Glorot and Bengio, 2010).

The attention outputs are

$$X^O \;=\; A\,V\,W^O,$$

where $A = \mathrm{softmax}\big(QK^\top/\sqrt{d_m}\big)$ and $V = XW^V$. Differentials give

$$d\mathcal{L}(X^O) = \mathrm{tr}\big[(\partial\mathcal{L}/\partial X^O)^\top\, AX\, dW^V W^O\big] = \mathrm{tr}\big[(AX)^\top\,\frac{\partial \mathcal{L}}{\partial X^O}\,(W^O)^\top\, dW^V\big].$$

so that

$$\frac{\partial \mathcal{L}}{\partial W^V} = (AX)^\top\,\frac{\partial \mathcal{L}}{\partial X^O}\,(W^O)^\top.$$

Similarly, writing $S = QK^\top/\sqrt{d_m}$,

$$d\mathcal{L}(S) = \mathrm{tr}\big[(\partial\mathcal{L}/\partial S)^\top\, dS\big] = \frac{1}{\sqrt{d_m}}\,\mathrm{tr}\big[X^\top\,(\partial\mathcal{L}/\partial S)^\top\, Q\, dW^K\big].$$

yielding

$$\frac{\partial \mathcal{L}}{\partial W^K} = \frac{1}{\sqrt{d_m}}\,X^\top\,(\partial\mathcal{L}/\partial S)^\top\, Q.$$

Squaring the Frobenius norm of (A.2) and taking an expectation under the isotropic-input assumption,

$$\mathbb{E}\big[\|\partial\mathcal{L}/\partial W^V\|_F^2\big] = c\,\sigma_x^2\,\sigma_o^2,$$

where $\mathbb{E}[X^\top X] = c\sigma_x^2 I$ and $\mathbb{E}\Big[\Big(A^\top \frac{\partial \mathcal{L}}{\partial X^O}(W^O)^\top\Big)\Big(A^\top \frac{\partial \mathcal{L}}{\partial X^O}(W^O)^\top\Big)^\top\Big] = \sigma_o^2 I$.

For the key weights,

$$\mathbb{E}\big[\|\partial\mathcal{L}/\partial W^K\|_F^2\big] = \frac{\sigma_x^2\,\sigma_s^2}{d_m}\,\|Q\|_F^2,$$

with σ_s^2 the entry-wise variance of $\partial\mathcal{L}/\partial S$.

Substituting these expectations into (A.2) shows that

$\mathbb{E}\big[\|W^K\|_F^2\big]$ grows as $\eta^2\,\frac{\sigma_x^2\sigma_s^2}{d_m}\,\|Q\|_F^2$, whereas $\mathbb{E}\big[\|W^V\|_F^2\big]$ grows as $\eta^2\,c\sigma_x^2\sigma_o^2$.
Because $\|Q\|_F^2$ itself increases during training, $\|W^K\|_F$ eventually dominates:

$$\mathbb{E}\big[\|W^K\|_F\big] \;>\; \mathbb{E}\big[\|W^V\|_F\big].$$

As shown in Equation A.2, the expectation of the Frobenius norm of W^k is greater than that of W^v.[2]

B Quantization Error Bounds

We establish upper bounds on the quantization error incurred when a matrix is represented using a finite bit-width in two's complement format. Two theorems are presented: one that characterizes the error in terms of the spectral norm and another in terms of the Frobenius norm.

The first theorem demonstrates that the spectral norm of the quantization error is bounded by

$$\|A - \widehat{A}\|_2 \;\lesssim\; \frac{\sqrt{m\,n}}{2^b}\,\|A\|_2.$$

This result implies that, for a given bit-width b, matrices with larger spectral norms incur proportionally larger quantization errors. Consequently, a matrix that exhibits a larger spectral norm is more susceptible to quantization errors. To control error propagation in such matrices, a higher bit width is necessary.

The second theorem provides an analogous bound for the Frobenius norm,

$$\|A - \widehat{A}\|_F \;\lesssim\; \frac{\sqrt{m\,n}}{2^b}\,\|A\|_F.$$

Similar to the spectral norm result, this bound indicates that the quantization error, measured in the Frobenius norm, is directly proportional to the norm of the original matrix. Hence, matrices with larger Frobenius norms are also more vulnerable to quantization errors and would benefit from a higher precision during quantization.

[2]The analysis considers a single-head decoder-only Transformer without loss of generality; the conclusions extend directly to multi-head, group-query, and multi-query attention.

B.1 Preliminaries

Let $A \in \mathbb{R}^{m\times n}$ denote a real matrix whose entries are to be stored using a fixed number of bits in two's complement representation. The following notation is adopted:

- *Bit Depth.* Given b bits in two's-complement format, each representable integer q lies in the interval

$$q \in \{-2^{b-1}, -2^{b-1}+1, \ldots, 2^{b-1}-1\}.$$

- *Maximum Entry Magnitude.* Define

$$M \;=\; \max_{1\le i\le m,\,1\le j\le n} |A_{ij}|.$$

- *Scale Factor.* Set

$$\alpha \;=\; \frac{M}{2^{b-1}-1}.$$

  This choice ensures that the scaled entries A_{ij}/α lie within the representable range.

- *Quantization.* Define the integer matrix $Q \in \mathbb{Z}^{m\times n}$ by

$$Q_{ij} = \mathrm{round}\Big(\frac{A_{ij}}{\alpha}\Big),$$
$$\text{with}\quad Q_{ij} \in \{-2^{b-1}, \ldots, 2^{b-1}-1\}.$$

- *Dequantization (Reconstruction).* The reconstructed matrix $\widehat{A}$ is given by

$$\widehat{A}_{ij} \;=\; \alpha\,Q_{ij}.$$

Objective. The aim is to bound the errors

$$\|A - \widehat{A}\|_2 \quad\text{and}\quad \|A - \widehat{A}\|_F,$$

in terms of b, m, n, and the norms of A.

B.2 Spectral Norm Error Bound

Theorem 3 (Spectral Norm Error Bound for Uniform Quantization)

Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{N}$ be given. Consider two's complement quantization with a scale factor of

$$\alpha \;=\; \frac{M}{2^{b-1}-1}, \quad M \;=\; \max_{i,j}|A_{ij}|.$$

Define

$$Q_{ij} \;=\; \operatorname{round}\Big(\frac{A_{ij}}{\alpha}\Big) \quad \textit{and} \quad \widehat{A}_{ij} \;=\; \alpha\, Q_{ij}.$$

Then, the following bound holds:

$$M \le \|A-\widehat{A}\|_2 \le \sqrt{m\,n}\,\frac{M}{2(2^{b-1}-1)} \le \sqrt{m\,n}\,\frac{\|A\|_2}{2(2^{b-1}-1)}.$$

In approximate form for large b,

$$\|A-\widehat{A}\|_2 \;\lesssim\; \frac{\sqrt{m\,n}}{2^b}\,\|A\|_2.$$

Proof. **Entrywise Bound.** By construction,

$$\left|\frac{A_{ij}}{\alpha} - Q_{ij}\right| \;\le\; \frac{1}{2}.$$

Multiplying by α gives

$$|A_{ij}-\widehat{A}_{ij}| \;\le\; \frac{\alpha}{2} \;=\; \frac{M}{2\,(2^{b-1}-1)}.$$

Thus,

$$\max_{i,j}\,|A_{ij}-\widehat{A}_{ij}| \;\le\; \frac{M}{2\,(2^{b-1}-1)}.$$

Conversion to the Spectral Norm. Using the inequality

$$\|B\|_2 \;\le\; \sqrt{m\,n}\,\max_{i,j}|B_{ij}|,$$

with $B = A-\widehat{A}$, it follows that

$$\|A-\widehat{A}\|_2 \;\le\; \sqrt{m\,n}\,\frac{M}{2\,(2^{b-1}-1)}.$$

Relating M to $\|A\|_2$. Since

$$M \;=\; \max_{i,j}|A_{ij}| \;\le\; \|A\|_2,$$

the bound can be written as

$$\|A-\widehat{A}\|_2 \;\le\; \sqrt{m\,n}\,\frac{\|A\|_2}{2\,(2^{b-1}-1)}.$$

For large b, where $2^{b-1}-1 \approx 2^{b-1}$, the bound becomes

$$\|A-\widehat{A}\|_2 \;\lesssim\; \frac{\sqrt{m\,n}}{2^b}\,\|A\|_2.$$

□

B.3 Frobenius Norm Error Bound

Theorem 4 (Frobenius Norm Error Bound for Uniform Quantization)

Under the same setup as Theorem 3, the Frobenius norm of the quantization error satisfies

$$\|A-\widehat{A}\|_F \;\le\; \sqrt{m\,n}\,\tfrac{M}{2(2^{b-1}-1)} \;\le\; \sqrt{m\,n}\,\tfrac{\|A\|_F}{2(2^{b-1}-1)}.$$

In approximate form,

$$\|A-\widehat{A}\|_F \;\lesssim\; \frac{\sqrt{m\,n}}{2^b}\,\|A\|_F.$$

Proof. **Entrywise Bound.** As established,

$$|A_{ij}-\widehat{A}_{ij}| \;\le\; \frac{M}{2\,(2^{b-1}-1)} \quad \text{for all } i,j.$$

Conversion to the Frobenius Norm. By definition,

$$\|A-\widehat{A}\|_F^2 \;=\; \sum_{i=1}^{m}\sum_{j=1}^{n}(A_{ij}-\widehat{A}_{ij})^2,$$

which yields

$$\|A-\widehat{A}\|_F^2 \;\le\; m\,n\left(\frac{M}{2\,(2^{b-1}-1)}\right)^2.$$

Taking square roots leads to

$$\|A-\widehat{A}\|_F \;\le\; \sqrt{m\,n}\,\frac{M}{2\,(2^{b-1}-1)}.$$

Relating M to $\|A\|_F$. Since

$$M^2 \;\le\; \sum_{i=1}^{m}\sum_{j=1}^{n}A_{ij}^2 \;=\; \|A\|_F^2,$$

it follows that $M \le \|A\|_F$ and hence

$$\|A-\widehat{A}\|_F \;\le\; \sqrt{m\,n}\,\frac{\|A\|_F}{2\,(2^{b-1}-1)}.$$

For large b, this simplifies to

$$\|A - \widehat{A}\|_F \lesssim \frac{\sqrt{m\,n}}{2^b} \|A\|_F.$$

□

Remark. The results indicate that both the spectral norm and Frobenius norm errors satisfy similar approximate bounds:

$$\|A - \widehat{A}\|_2 \lesssim \frac{\sqrt{m\,n}}{2^b} \|A\|_2,$$
$$\|A - \widehat{A}\|_F \lesssim \frac{\sqrt{m\,n}}{2^b} \|A\|_F.$$

B.4 Implications for KV Cache Quantization

Consider the key and value cache

$$\mathbf{V} \in \mathbb{R}^{L \times d_{\text{head}}}, \quad \mathbf{K} \in \mathbb{R}^{L \times d_{\text{head}}},$$

with quantization bit-widths denoted by b_V and b_K, respectively. Let $\widehat{\mathbf{V}}$ and $\widehat{\mathbf{K}}$ denote the dequantized matrices, and define the quantization errors as

$$\mathbf{E}_V = \mathbf{V} - \widehat{\mathbf{V}}, \quad \mathbf{E}_K = \mathbf{K} - \widehat{\mathbf{K}}.$$

An empirical observation is that

$$\|\mathbf{V}\|_* < \|\mathbf{K}\|_*,$$

where $*$ denotes either the spectral norm ($\|\cdot\|_2$) or the Frobenius norm ($\|\cdot\|_F$). In practice, $\mathbf{K}$ typically exhibits a larger norm than $\mathbf{V}$.

Spectral-Norm Perspective. Standard quantization error bounds yield

$$\|\mathbf{E}_V\|_2 \lesssim \frac{\sqrt{L\,d_{\text{head}}}}{2^{b_V}} \|\mathbf{V}\|_2,$$
$$\|\mathbf{E}_K\|_2 \lesssim \frac{\sqrt{L\,d_{\text{head}}}}{2^{b_K}} \|\mathbf{K}\|_2.$$

To achieve comparable spectral-norm errors (i.e., $\|\mathbf{E}_V\|_2 \approx \|\mathbf{E}_K\|_2$), it is necessary that

$$\frac{\sqrt{L\,d_{\text{head}}}}{2^{b_V}} \|\mathbf{V}\|_2 \approx \frac{\sqrt{L\,d_{\text{head}}}}{2^{b_K}} \|\mathbf{K}\|_2.$$

Cancelling the common factor $\sqrt{L\,d_{\text{head}}}$ yields

$$2^{b_V} \|\mathbf{V}\|_2 \approx 2^{b_K} \|\mathbf{K}\|_2,$$

or equivalently,

$$2^{\,b_K - b_V} \approx \frac{\|\mathbf{V}\|_2}{\|\mathbf{K}\|_2}.$$

Since $\|\mathbf{V}\|_2 < \|\mathbf{K}\|_2$, it follows that $b_K > b_V$.

Frobenius Norm (MSE) Perspective. The Frobenius norm of the quantization error corresponds directly to the mean-squared error (MSE) when normalized by the number of elements. Specifically, for a matrix $\mathbf{A}$ with quantized approximation $\widehat{\mathbf{A}}$, the MSE is

$$\text{MSE}(\mathbf{A}, \widehat{\mathbf{A}}) = \frac{1}{\text{nnz}(\mathbf{A})} \|\mathbf{A} - \widehat{\mathbf{A}}\|_F^2,$$

where $\text{nnz}(\mathbf{A})$ denotes the number of entries. Thus, controlling the Frobenius norm is equivalent to controlling the MSE up to a scaling factor.

The quantization error bounds under the Frobenius norm are given by

$$\|\mathbf{E}_V\|_F \lesssim \frac{\sqrt{L\,d_{\text{head}}}}{2^{b_V}} \|\mathbf{V}\|_F,$$
$$\|\mathbf{E}_K\|_F \lesssim \frac{\sqrt{L\,d_{\text{head}}}}{2^{b_K}} \|\mathbf{K}\|_F,$$

where L is the sequence length and d_{head} is the head dimension.

To ensure comparable Frobenius (or MSE) errors between keys and values, we require

$$\frac{\sqrt{L\,d_{\text{head}}}}{2^{b_V}} \|\mathbf{V}\|_F \approx \frac{\sqrt{L\,d_{\text{head}}}}{2^{b_K}} \|\mathbf{K}\|_F,$$

which simplifies to

$$2^{b_V} \|\mathbf{V}\|_F \approx 2^{b_K} \|\mathbf{K}\|_F,$$

and therefore

$$2^{\,b_K - b_V} \approx \frac{\|\mathbf{V}\|_F}{\|\mathbf{K}\|_F}.$$

Since typically $\|\mathbf{V}\|_F < \|\mathbf{K}\|_F$, it follows that $b_K > b_V$. This reinforces the earlier spectral-norm result: keys should be allocated more bits than values to achieve balanced quantization error under an MSE criterion.

C Supplemental Results

C.1 Focus on Generative Tasks

We validate and operationalize our theorems across a diverse set of datasets, including C4, MMLU, GSM8K, EQ-Bench, CoQA, and LongBench. Our evaluation purposefully focuses on *generative* tasks, i.e., open-ended generation and free-form responses, because KV-cache quantization primarily affects the *decoding phase* rather than the *prefill phase*. To assess quantization error in isolation,

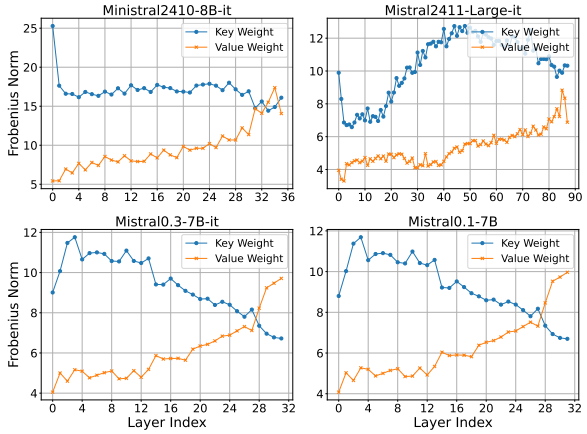


Figure 5: **Frobenius norm plot for Mistral Family.** The x-axis represents the layer index in the model, while the y-axis represents the Frobenius norm magnitude. The spectral norms are higher for the key weights than for the value weights across layers.

we use C4 under a free-form decoding setup that mirrors pretraining usage, enabling us to analyze how compression directly distorts activations. To quantify downstream impact, we select GSM8K, CoQA, EQ-Bench, and LongBench, all of which rely on `generate_until`-style evaluation, where the model must produce extended, structured responses. In contrast, commonsense reasoning benchmarks (e.g., BoolQ, PIQA, HellaSwag, or DoRA) use log-likelihood scoring over candidate options and do not engage KV-cache quantization unless the input prompt itself is compressed. Consequently, such discriminative evaluations are orthogonal to our focus and were intentionally excluded.

Within LongBench, we surveyed all subtasks (gov_report, lcc, lsht, multi_news, narrativeqa, qasper, qmsum, repobench-p, samsum) and found that only `gov_report` and `qmsum` require substantial generation, with `gov_report` involving the longest outputs. Throughout this work, "LongBench" refers specifically to `gov_report`.

C.2 Key-Value Weight Norm

Figure 5 presents the Frobenius norms of key and value weights for the Mistral family, exhibiting the same pattern (keys consistently having higher norms than values) as shown in Figure 2 for the Llama family, further corroborating the Key-Value Norm Disparity theorem established in Section 3.1.

C.3 Singular Value Distributions

Figure 6 illustrates the full-spectrum singular value distribution of key and value caches across layers of the Llama 3.3 70B model on the C4 dataset. The horizontal axis indexes singular values in descending order, starting from the largest (i.e., the spectral norm), while the vertical axis shows their magnitudes. The shaded region represents the minimum-maximum range of singular values across attention heads within each layer, and the solid curves indicate the mean singular value at each rank.

C.4 Quantization Error

By using quantization error, we refer to evaluating how closely the quantized-then-dequantized key/value (KV) caches reconstruct their full-precision counterparts, measured as the reconstruction mean-squared error (MSE), which is exactly the Frobenius norm error normalized by the number of elements. This serves as a practical proxy for downstream quality because uniform b-bit quantization admits norm-based bounds in which reconstruction error scales with the cache norm and decays roughly like 2^{-b}; thus, at a fixed bit budget, higher-norm caches incur larger distortion, and bit allocations that minimize reconstruction error are directly targeting the quantity most affected by quantization.

C.5 Downstream Accuracy Across Quantization Precisions

Comprehensive downstream accuracy results for KV cache quantization using the HQQ backend are provided for CoQA (F1 scores), EQ-Parseable (parseable-pass rates and exact-match scores), and GSM8K (flexible and strict accuracy) in Tables 7, 8, 9, and 10, respectively.

C.6 Rotation and Grouping Results

Tables 11-14 present the full set of downstream evaluation results for integrating rotation-based outlier redistribution with mixed-precision KV quantization across multiple models and tasks, including GSM8K (flexible and strict exact match), CoQA (F1), and CoQA (exact match). Each table reports accuracy for four key-value bit allocations ($\mathbf{K}_2\mathbf{V}_2$, $\mathbf{K}_2\mathbf{V}_4$, $\mathbf{K}_4\mathbf{V}_2$, $\mathbf{K}_4\mathbf{V}_4$) under four Rotations: none, key-only, value-only, and both key and value. The group size is fixed at 64 for both keys and values in these experiments to isolate the effect of rotation.

The results reveal several consistent trends across model scales and tasks. First, applying rota-

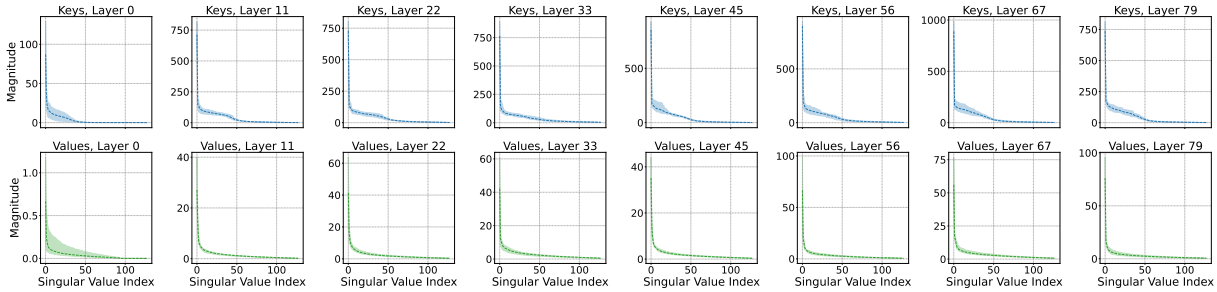


Figure 6: **Complete singular value distribution of key and value cache** for Llama 3.3-70B on the C4 dataset. The x-axis denotes the singular value indices, ordered from the largest (spectral norm) to the smallest, while the y-axis represents the corresponding magnitudes. The shaded region illustrates the range between the minimum and maximum singular values across attention heads within each layer, and the solid lines indicate the mean singular value magnitude at each index. This full-spectrum view highlights that key matrices consistently maintain significantly higher singular values throughout the entire distribution, further reinforcing their dominant representational capacity compared to value matrices.

Table 4: Quantization error (mean ± std) for the key and value caches ($\mathbf{K}_i$ and $\mathbf{V}_i$) at 2-bit, 3-bit, and 4-bit quantization, evaluated on the **MMLU** dataset.

	$\mathbf{K}_2$	$\mathbf{V}_2$	$\mathbf{K}_3$	$\mathbf{V}_3$	$\mathbf{K}_4$	$\mathbf{V}_4$
Llama3.2-1B	4.851 ± 1.037	0.127 ± 0.101	1.037 ± 0.265	0.021 ± 0.015	0.227 ± 0.059	0.005 ± 0.003
Llama3.2-1B-it	4.373 ± 1.034	0.124 ± 0.090	0.879 ± 0.218	0.019 ± 0.013	0.192 ± 0.047	0.004 ± 0.003
Llama3.2-3B	3.943 ± 0.924	0.193 ± 0.096	0.849 ± 0.150	0.030 ± 0.015	0.183 ± 0.031	0.007 ± 0.003
Llama3.2-3B-it	4.487 ± 1.180	0.202 ± 0.100	0.894 ± 0.180	0.030 ± 0.015	0.193 ± 0.037	0.007 ± 0.003
Llama2-7B	3.190 ± 0.783	0.259 ± 0.184	0.769 ± 0.194	0.042 ± 0.030	0.168 ± 0.042	0.009 ± 0.006
Llama3.1-8B-it	6.003 ± 1.782	0.187 ± 0.127	1.082 ± 0.244	0.028 ± 0.019	0.235 ± 0.055	0.006 ± 0.004
Llama3.3-70B-it	4.883 ± 1.106	0.112 ± 0.093	0.942 ± 0.198	0.016 ± 0.012	0.206 ± 0.043	0.003 ± 0.003
Nemotron3.1-it	5.125 ± 1.284	0.114 ± 0.094	0.985 ± 0.207	0.016 ± 0.012	0.216 ± 0.046	0.003 ± 0.003
Phi3-Medium-128K-it	5.063 ± 1.914	0.584 ± 0.559	1.000 ± 0.319	0.087 ± 0.083	0.217 ± 0.068	0.019 ± 0.018
Phi4	5.929 ± 1.545	0.657 ± 0.472	1.306 ± 0.231	0.103 ± 0.070	0.286 ± 0.050	0.022 ± 0.015
Mistral0.3-7B	4.718 ± 1.340	0.398 ± 0.405	0.941 ± 0.240	0.059 ± 0.059	0.206 ± 0.053	0.013 ± 0.013
Qwen2.5-14B	5.184 ± 2.241	1.270 ± 1.547	1.005 ± 0.288	0.182 ± 0.221	0.223 ± 0.067	0.040 ± 0.052
DeepSeekR1L-8B	5.502 ± 1.549	0.189 ± 0.118	0.955 ± 0.204	0.028 ± 0.017	0.209 ± 0.046	0.006 ± 0.004
DeepSeekR1Q-14B	5.126 ± 2.375	1.406 ± 1.609	0.900 ± 0.269	0.198 ± 0.226	0.199 ± 0.062	0.044 ± 0.052

Table 5: Quantization error (mean ± std) for the key and value caches ($\mathbf{K}_i$ and $\mathbf{V}_i$) at 2-bit, 3-bit, and 4-bit quantization, evaluated on the **C4** dataset.

	$\mathbf{K}_2$	$\mathbf{V}_2$	$\mathbf{K}_3$	$\mathbf{V}_3$	$\mathbf{K}_4$	$\mathbf{V}_4$
Llama3.2-1B	4.885 ± 1.056	0.207 ± 0.166	1.074 ± 0.289	0.030 ± 0.024	0.233 ± 0.062	0.006 ± 0.005
Llama3.2-1B-it	4.524 ± 1.108	0.193 ± 0.137	0.925 ± 0.235	0.028 ± 0.020	0.201 ± 0.050	0.006 ± 0.005
Llama3.2-3B	3.885 ± 0.777	0.282 ± 0.150	0.909 ± 0.168	0.042 ± 0.023	0.194 ± 0.032	0.009 ± 0.005
Llama3.2-3B-it	4.135 ± 1.088	0.274 ± 0.137	0.912 ± 0.176	0.039 ± 0.020	0.195 ± 0.036	0.009 ± 0.004
Llama2-7B	6.337 ± 1.710	0.456 ± 0.247	1.054 ± 0.263	0.071 ± 0.038	0.213 ± 0.052	0.015 ± 0.008
Llama3.1-8B-it	6.262 ± 1.789	0.254 ± 0.185	1.128 ± 0.249	0.036 ± 0.026	0.247 ± 0.056	0.008 ± 0.005
Llama3.3-70B-it	4.391 ± 1.027	0.121 ± 0.097	0.847 ± 0.175	0.017 ± 0.013	0.186 ± 0.038	0.004 ± 0.003
Nemotron3.1-it	5.367 ± 1.332	0.127 ± 0.105	1.049 ± 0.222	0.018 ± 0.013	0.231 ± 0.049	0.004 ± 0.003
Phi3-Medium-128K-it	4.831 ± 1.759	0.788 ± 0.726	1.022 ± 0.306	0.109 ± 0.097	0.220 ± 0.064	0.023 ± 0.021
Phi4	5.715 ± 1.442	0.850 ± 0.684	1.316 ± 0.245	0.124 ± 0.093	0.291 ± 0.056	0.027 ± 0.020
Mistral0.3-7B	5.027 ± 1.332	0.543 ± 0.493	1.014 ± 0.269	0.079 ± 0.068	0.223 ± 0.060	0.017 ± 0.015
Qwen2.5-14B	4.382 ± 2.170	1.544 ± 1.872	0.846 ± 0.250	0.220 ± 0.265	0.187 ± 0.060	0.048 ± 0.060
DeepSeekR1L-8B	4.575 ± 1.122	0.204 ± 0.134	0.817 ± 0.141	0.030 ± 0.019	0.179 ± 0.033	0.006 ± 0.004
DeepSeekR1Q-14B	4.832 ± 2.354	1.651 ± 1.914	0.927 ± 0.283	0.232 ± 0.267	0.201 ± 0.061	0.051 ± 0.060

Table 6: Quantization error (mean ± std) for the key and value caches (K_i and V_i) at 2-bit, 3-bit, and 4-bit quantization, evaluated on the GSM8K dataset.

	$\mathbf{K}_2$	$\mathbf{V}_2$	$\mathbf{K}_3$	$\mathbf{V}_3$	$\mathbf{K}_4$	$\mathbf{V}_4$
Llama3.2-1B	5.703 ± 1.557	0.179 ± 0.136	1.213 ± 0.352	0.026 ± 0.020	0.266 ± 0.078	0.005 ± 0.004
Llama3.2-1B-it	5.002 ± 1.383	0.171 ± 0.130	1.024 ± 0.287	0.025 ± 0.020	0.223 ± 0.061	0.006 ± 0.004
Llama3.2-3B	4.840 ± 1.396	0.261 ± 0.136	1.045 ± 0.211	0.038 ± 0.021	0.226 ± 0.044	0.008 ± 0.005
Llama3.2-3B-it	3.604 ± 0.850	0.226 ± 0.129	0.790 ± 0.135	0.034 ± 0.019	0.171 ± 0.028	0.007 ± 0.004
Llama2-7B	5.081 ± 1.396	0.405 ± 0.231	0.969 ± 0.238	0.065 ± 0.037	0.205 ± 0.050	0.014 ± 0.008
Llama3.1-8B-it	6.445 ± 1.837	0.213 ± 0.161	1.184 ± 0.268	0.030 ± 0.022	0.257 ± 0.060	0.007 ± 0.005
Llama3.3-70B-it	4.967 ± 1.127	0.113 ± 0.091	0.978 ± 0.203	0.016 ± 0.012	0.214 ± 0.044	0.004 ± 0.003
Nemotron3.1-it	4.752 ± 1.124	0.113 ± 0.089	0.940 ± 0.194	0.016 ± 0.012	0.206 ± 0.042	0.004 ± 0.003
Phi3-Medium-128K-it	4.940 ± 1.834	0.605 ± 0.579	1.042 ± 0.320	0.088 ± 0.082	0.227 ± 0.069	0.019 ± 0.018
Phi4	6.610 ± 1.624	0.785 ± 0.598	1.498 ± 0.293	0.116 ± 0.082	0.330 ± 0.064	0.025 ± 0.017
Mistral0.3-7B	5.308 ± 1.367	0.461 ± 0.434	1.065 ± 0.288	0.067 ± 0.061	0.232 ± 0.061	0.015 ± 0.013
Qwen2.5-14B	4.829 ± 2.179	1.736 ± 2.659	0.979 ± 0.264	0.241 ± 0.372	0.214 ± 0.061	0.051 ± 0.077
DeepSeekR1L-8B	5.547 ± 1.517	0.193 ± 0.129	1.000 ± 0.212	0.028 ± 0.018	0.218 ± 0.049	0.006 ± 0.004
DeepSeekR1Q-14B	4.477 ± 2.176	1.424 ± 1.752	0.830 ± 0.256	0.200 ± 0.242	0.181 ± 0.058	0.044 ± 0.056

Table 7: **Downstream Accuracy on CoQA (Word-Overlap F1) Across Quantization Precisions**. (K_xV_y) denotes x-bit keys and y-bit values; higher is better. Results show the keys are the bottleneck while values can be compressed more aggressively: moving from 1-bit to 2-bit **keys** with **values fixed at 1-bit** yields large gains (e.g., Qwen3-32B: 0.231 $\rightarrow$ 0.732; Llama-3.2-3B: 0.128 $\rightarrow$ 0.577), whereas at **fixed keys $\geq$4-bit**, sweeping values from 1$\rightarrow$8 bits changes F1 only marginally (e.g., Qwen3-8B (K_6V_1) 0.813 vs. (K_6V_8) 0.819; Qwen3-32B (K_8V_1) 0.818 vs. (K_8V_8) 0.827). With sufficiently precise keys (6-8 bits), even **2-bit values** nearly match BF16: Llama-3.2-1B (K_6V_2) 0.700 vs. BF16 0.701; Qwen3-32B (K_6V_2) 0.826 vs. BF16 0.826. In contrast, raising **values** at 1-bit keys barely helps (e.g., Qwen3-4B (K_1V_2) 0.361 vs. (K_1V_8) 0.364).

	Qwen3 0.6B	**Llama-3.2** 1B	**Llama-3.2** 3B	**Qwen3** 4B	**Llama-3.1** 8B	**Qwen3** 8B	**Qwen3** 32B
$\mathbf{K_1V_1}$	0.130	0.148	0.128	0.222	0.127	0.359	0.231
$\mathbf{K_1V_2}$	0.227	0.236	0.223	0.361	0.207	0.384	0.328
$\mathbf{K_1V_4}$	0.226	0.237	0.229	0.376	0.242	0.386	0.349
$\mathbf{K_1V_6}$	0.215	0.243	0.230	0.379	0.221	0.379	0.352
$\mathbf{K_1V_8}$	0.220	0.240	0.222	0.364	0.242	0.382	0.355
$\mathbf{K_2V_1}$	0.253	0.190	0.577	0.464	0.565	0.599	0.732
$\mathbf{K_2V_2}$	0.334	0.526	0.760	0.721	0.757	0.767	0.809
$\mathbf{K_2V_4}$	0.333	0.559	0.766	0.732	0.763	0.761	0.809
$\mathbf{K_2V_6}$	0.340	0.565	0.764	0.730	0.764	0.766	0.807
$\mathbf{K_2V_8}$	0.332	0.568	0.764	0.733	0.766	0.770	0.804
$\mathbf{K_4V_1}$	0.497	0.575	0.769	0.803	0.773	0.815	0.818
$\mathbf{K_4V_2}$	0.666	0.693	0.797	0.806	0.786	0.822	0.820
$\mathbf{K_4V_4}$	0.692	0.701	0.798	0.807	0.788	0.819	0.825
$\mathbf{K_4V_6}$	0.688	0.702	0.797	0.809	0.784	0.817	0.824
$\mathbf{K_4V_8}$	0.687	0.700	0.797	0.807	0.782	0.818	0.823
$\mathbf{K_6V_1}$	0.650	0.597	0.771	0.801	0.765	0.813	0.817
$\mathbf{K_6V_2}$	0.693	0.700	0.798	0.803	0.785	0.821	0.826
$\mathbf{K_6V_4}$	0.702	0.705	0.796	0.807	0.786	0.821	0.825
$\mathbf{K_6V_6}$	0.701	0.706	0.795	0.808	0.790	0.819	0.825
$\mathbf{K_6V_8}$	0.694	0.703	0.796	0.807	0.789	0.819	0.825
$\mathbf{K_8V_1}$	0.651	0.599	0.769	0.802	0.767	0.815	0.818
$\mathbf{K_8V_2}$	0.691	0.696	0.798	0.803	0.788	0.823	0.825
$\mathbf{K_8V_4}$	0.701	0.704	0.795	0.807	0.789	0.821	0.826
$\mathbf{K_8V_6}$	0.699	0.705	0.795	0.809	0.787	0.819	0.826
$\mathbf{K_8V_8}$	0.697	0.704	0.795	0.808	0.786	0.821	0.827
BF16	0.708	0.701	0.795	0.808	0.785	0.820	0.826

Table 8: **Downstream Accuracy on EQ-BENCH PARSEABLE Across Quantization Precisions**. (K_xV_y) denotes x-bit keys and y-bit values; the higher the value, the better. *Parseable accuracy* is the share of prompts where the model's output follows the required structured format so the scorer can automatically extract the four 0-10 emotion ratings. Results consistently show that keys are the bottleneck, while values can be compressed more aggressively. With 1-bit keys, outputs are never parseable across all models regardless of value precision ((K_1V_y=0) everywhere). Upgrading to **2-bit keys** yields large jumps even with very low-precision values. For instance, Llama-3.2-3B rises from 0 to **80.702** at (K_2V_2), Llama-3.1-8B from 0 to **85.965**, and Qwen3-32B from 0 to **67.836**, while further increasing **values** from 2→8 bits at fixed (K=2) brings only modest gains (e.g., Llama-3.2-3B **80.702**→**84.211**, Llama-3.1-8B **85.965**→**90.059**, Qwen3-8B **65.497**→**68.421**). Once **keys are ≥4-6 bits**, even **1-2-bit values** nearly saturate parseability and often match or exceed BF16 (e.g., Qwen3-8B (K_4V_1) **98.830** vs. BF16 **94.737**; Qwen3-32B (K_6V_1) **79.532** ≈ BF16 **79.532**; Llama-3.2-1B (K_6V_2) **97.076** vs. BF16 **97.661**).

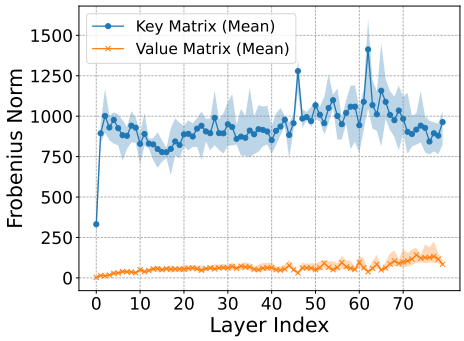
	Qwen3 0.6B	**Llama-3.2** 1B	**Llama-3.2** 3B	**Qwen3** 4B	**Llama-3.1** 8B	**Qwen3** 8B	**Qwen3** 32B
$\mathbf{K_1V_1}$	0	0	0	0	0	0	0
$\mathbf{K_1V_2}$	0	0	0	0	0	0	0
$\mathbf{K_1V_4}$	0	0	0	0	0	0	0
$\mathbf{K_1V_6}$	0	0	0	0	0	0	0
$\mathbf{K_1V_8}$	0	0	0	0	0	0	0
$\mathbf{K_2V_1}$	0	0	8.187	0	35.673	0	2.339
$\mathbf{K_2V_2}$	0	16.374	80.702	27.485	85.965	65.497	67.836
$\mathbf{K_2V_4}$	0	34.503	84.795	39.766	88.889	70.175	70.175
$\mathbf{K_2V_6}$	0	39.181	83.041	36.842	90.059	67.836	69.591
$\mathbf{K_2V_8}$	0	38.012	84.211	40.936	88.889	68.421	70.760
$\mathbf{K_4V_1}$	0	47.953	58.480	77.778	83.626	98.830	77.778
$\mathbf{K_4V_2}$	63.743	97.076	95.322	87.719	97.661	95.322	80.702
$\mathbf{K_4V_4}$	65.497	97.076	98.830	84.795	99.415	95.322	80.702
$\mathbf{K_4V_6}$	64.328	97.076	98.830	85.380	98.830	93.567	80.702
$\mathbf{K_4V_8}$	62.573	97.076	98.830	84.795	98.830	93.567	80.702
$\mathbf{K_6V_1}$	23.392	60.234	53.801	81.287	90.643	96.491	79.532
$\mathbf{K_6V_2}$	99.415	97.076	94.737	87.719	97.661	94.737	80.702
$\mathbf{K_6V_4}$	99.415	97.661	99.415	87.135	98.830	95.906	80.702
$\mathbf{K_6V_6}$	99.415	97.661	98.830	86.550	98.830	96.491	80.702
$\mathbf{K_6V_8}$	99.415	97.661	98.830	87.135	98.830	97.661	80.702
$\mathbf{K_8V_1}$	26.901	60.234	56.725	78.947	92.983	97.076	77.193
$\mathbf{K_8V_2}$	99.415	97.076	94.737	87.135	97.661	96.491	79.532
$\mathbf{K_8V_4}$	99.415	97.661	99.415	87.719	98.830	95.322	80.702
$\mathbf{K_8V_6}$	99.415	97.661	99.415	86.550	98.830	95.906	80.702
$\mathbf{K_8V_8}$	99.415	97.661	99.415	86.550	98.830	95.906	80.702
BF16	100	97.661	99.415	87.719	98.830	94.737	79.532

Table 9: **Downstream Accuracy on GSM8K (FLEXIBLE) Across Quantization Precisions**. (K_xV_y) denotes x-bit keys and y-bit values; the higher the value, the better. *Flexible accuracy* counts a prediction as correct if the gold final numeric answer appears anywhere in the model's output (ignoring extra formatting), rather than requiring an isolated exact-match box. Results consistently show that keys are the bottleneck, while values can be compressed more aggressively. With 1-bit keys, performance is essentially zero across all models (max $= 0.028$). Upgrading to **2-bit keys** yields large jumps even with low-precision values—for example, Llama-3.2-3B rises from (0.015) at (K_1V_2) to $\mathbf{0.278}$ at (K_2V_2), Llama-3.1-8B from (0.020) to $\mathbf{0.385}$, and Qwen3-32B from (0.018) to $\mathbf{0.385}$. At fixed $\mathbf{K(\geq 6)}$, even **2-bit values** already match or beat BF16 (e.g., Qwen3-8B (K_6V_2) $\mathbf{0.880}$ vs. BF16 $\mathbf{0.877}$; Llama-3.1-8B $\mathbf{0.762}$ vs. $\mathbf{0.760}$; Qwen3-4B $\mathbf{0.830}$ vs. $\mathbf{0.845}$), and increasing values from (2→8) bits yields only modest gains (Llama-3.2-3B $\mathbf{0.644}(\rightarrow)\mathbf{0.665}$, Qwen3-8B $\mathbf{0.880}(\rightarrow)\mathbf{0.886}$). In several cases, quantized caches even *exceed* BF16 (e.g., Qwen3-8B (K_6V_8) $\mathbf{0.886} > 0.877$; Llama-3.2-3B (K_4V_6) $\mathbf{0.668} > 0.663$; Qwen3-32B (K_4V_1) $\mathbf{0.721} > 0.603$).

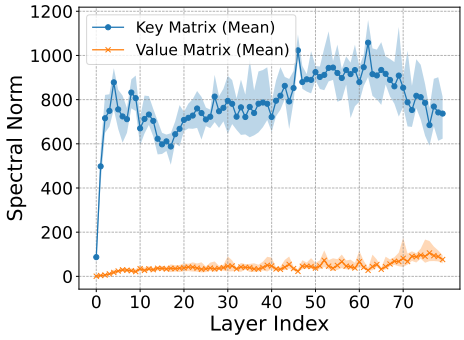
	Qwen3 0.6B	**Llama-3.2** 1B	**Llama-3.2** 3B	**Qwen3** 4B	**Llama-3.1** 8B	**Qwen3** 8B	**Qwen3** 32B
$\mathbf{K_1V_1}$	0.011	0.012	0.016	0.010	0.016	0.011	0.020
$\mathbf{K_1V_2}$	0.008	0.014	0.015	0.010	0.020	0.013	0.018
$\mathbf{K_1V_4}$	0.009	0.020	0.025	0.014	0.020	0.008	0.011
$\mathbf{K_1V_6}$	0.010	0.011	0.027	0.017	0.020	0.008	0.016
$\mathbf{K_1V_8}$	0.008	0.015	0.028	0.009	0.015	0.014	0.011
$\mathbf{K_2V_1}$	0.003	0.018	0.015	0.008	0.050	0.013	0.014
$\mathbf{K_2V_2}$	0.003	0.029	0.278	0.027	0.385	0.129	0.385
$\mathbf{K_2V_4}$	0.003	0.020	0.351	0.043	0.455	0.171	0.434
$\mathbf{K_2V_6}$	0.004	0.026	0.344	0.040	0.446	0.169	0.419
$\mathbf{K_2V_8}$	0.004	0.028	0.355	0.035	0.460	0.167	0.449
$\mathbf{K_4V_1}$	0.023	0.073	0.281	0.577	0.517	0.779	0.721
$\mathbf{K_4V_2}$	0.040	0.294	0.639	0.808	0.757	0.877	0.649
$\mathbf{K_4V_4}$	0.040	0.333	0.653	0.830	0.778	0.877	0.629
$\mathbf{K_4V_6}$	0.055	0.332	0.668	0.832	0.763	0.881	0.619
$\mathbf{K_4V_8}$	0.042	0.341	0.658	0.826	0.758	0.878	0.632
$\mathbf{K_6V_1}$	0.093	0.096	0.308	0.632	0.522	0.795	0.701
$\mathbf{K_6V_2}$	0.368	0.312	0.644	0.830	0.762	0.880	0.598
$\mathbf{K_6V_4}$	0.405	0.334	0.654	0.837	0.770	0.878	0.610
$\mathbf{K_6V_6}$	0.420	0.337	0.663	0.844	0.772	0.884	0.596
$\mathbf{K_6V_8}$	0.414	0.342	0.665	0.841	0.771	0.886	0.608
$\mathbf{K_8V_1}$	0.088	0.096	0.304	0.637	0.530	0.798	0.708
$\mathbf{K_8V_2}$	0.390	0.313	0.647	0.826	0.750	0.882	0.606
$\mathbf{K_8V_4}$	0.416	0.340	0.660	0.832	0.770	0.875	0.621
$\mathbf{K_8V_6}$	0.419	0.342	0.667	0.843	0.767	0.880	0.608
$\mathbf{K_8V_8}$	0.413	0.334	0.666	0.843	0.759	0.880	0.603
BF16	0.412	0.328	0.663	0.845	0.760	0.877	0.603

Table 10: **Downstream Accuracy on GSM8K (STRICT) Across Quantization Precisions**. (K_xV_y) denotes x-bit keys and y-bit values; higher is better. Same dataset as Table 9, but *Strict accuracy* only counts a prediction as correct when the scorer can extract a single final numeric answer that exactly matches the gold. As with the Flexible metric, keys are the bottleneck, while values can be compressed more aggressively. With 1-bit keys, accuracy is essentially zero across all models (max $= 0.002$). Upgrading to **2-bit keys** yields large jumps even with low-precision values—for example, Llama-3.2-3B rises from (0) to $\mathbf{0.276}$ at (K_2V_2), Llama-3.1-8B from (0) to $\mathbf{0.380}$, and Qwen3-32B from (0) to $\mathbf{0.234}$. Once **keys are ≥4-6 bits**, even **2-bit values** are near or above BF16 (e.g., Qwen3-8B (K_6V_2) $\mathbf{0.879}$ vs. BF16 $\mathbf{0.874}$; Llama-3.2-3B (K_8V_6) $\mathbf{0.661} > \mathbf{0.657}$; Qwen3-32B ($K_8V_6$) $\mathbf{0.723} > \mathbf{0.718}$). At fixed high key precision, increasing values from (2→8) bits brings only modest gains (e.g., Qwen3-8B at (K= 6): $\mathbf{0.879}(\rightarrow)\mathbf{0.885}$; Llama-3.2-3B at (K= 6): $\mathbf{0.639}(\rightarrow)\mathbf{0.657}$).

	Qwen3 0.6B	**Llama-3.2** 1B	**Llama-3.2** 3B	**Qwen3** 4B	**Llama-3.1** 8B	**Qwen3** 8B	**Qwen3** 32B
$\mathbf{K_1V_1}$	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$\mathbf{K_1V_2}$	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$\mathbf{K_1V_4}$	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$\mathbf{K_1V_6}$	0.000	0.000	0.002	0.000	0.000	0.001	0.000
$\mathbf{K_1V_8}$	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$\mathbf{K_2V_1}$	0.000	0.001	0.005	0.000	0.028	0.000	0.002
$\mathbf{K_2V_2}$	0.000	0.011	0.276	0.004	0.380	0.067	0.234
$\mathbf{K_2V_4}$	0.000	0.010	0.350	0.009	0.444	0.104	0.334
$\mathbf{K_2V_6}$	0.000	0.012	0.344	0.010	0.439	0.115	0.330
$\mathbf{K_2V_8}$	0.000	0.014	0.353	0.005	0.449	0.109	0.347
$\mathbf{K_4V_1}$	0.006	0.061	0.293	0.647	0.500	0.785	0.701
$\mathbf{K_4V_2}$	0.042	0.297	0.628	0.825	0.751	0.876	0.726
$\mathbf{K_4V_4}$	0.044	0.331	0.643	0.844	0.759	0.876	0.734
$\mathbf{K_4V_6}$	0.055	0.331	0.657	0.839	0.748	0.880	0.733
$\mathbf{K_4V_8}$	0.049	0.342	0.649	0.835	0.745	0.877	0.737
$\mathbf{K_6V_1}$	0.083	0.083	0.326	0.698	0.506	0.803	0.702
$\mathbf{K_6V_2}$	0.369	0.308	0.639	0.839	0.750	0.879	0.709
$\mathbf{K_6V_4}$	0.405	0.334	0.644	0.845	0.756	0.876	0.719
$\mathbf{K_6V_6}$	0.422	0.335	0.656	0.853	0.751	0.882	0.717
$\mathbf{K_6V_8}$	0.414	0.341	0.657	0.848	0.753	0.885	0.721
$\mathbf{K_8V_1}$	0.085	0.084	0.324	0.704	0.513	0.805	0.695
$\mathbf{K_8V_2}$	0.388	0.314	0.640	0.829	0.739	0.882	0.717
$\mathbf{K_8V_4}$	0.417	0.340	0.652	0.845	0.756	0.875	0.721
$\mathbf{K_8V_6}$	0.419	0.340	0.661	0.848	0.749	0.879	0.723
$\mathbf{K_8V_8}$	0.415	0.332	0.660	0.851	0.744	0.876	0.723
BF16	0.413	0.328	0.657	0.852	0.741	0.874	0.718



(a) Frobenius Norm



(b) Spectral Norm

Figure 7: **Frobenius and spectral norms of key and value caches across layers for the Llama 3.3 70B model on the C4 dataset.** The x-axis represents the layer index. The shaded regions indicate the min-max range across attention heads within each layer, while the solid curves represent the mean norm per layer. In both cases, key matrices consistently exhibit substantially higher norms than value matrices, reflecting their stronger representational capacity.

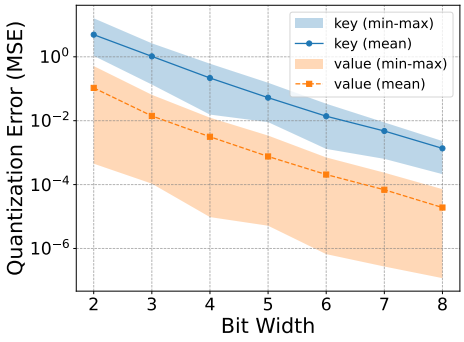


Figure 8: **MSE of KV quantization for Llama 3.3-70B on the C4 dataset.** We use quantization bit-widths ranging from 2 to 8. The x-axis represents the quantization bit-width, while the y-axis shows the MSE on a logarithmic scale. A logarithmic scale is used to highlight differences at higher bit-widths, where MSE values decrease significantly and approach zero, particularly at 8-bit. Solid lines indicate the mean MSE across layers, while the shaded regions represent the min-max range of errors.

tion to *keys* consistently improves performance at lower bit-widths ($\mathrm{K}_2\mathrm{V}_2$ and $\mathrm{K}_2\mathrm{V}_4$), significantly narrowing the gap to higher-precision baselines. In contrast, rotating *values* alone yields marginal or even negative effects, especially at low precisions. Notably, applying rotation to *keys only* under the $\mathrm{K}_4\mathrm{V}_2$ configuration achieves accuracy that closely matches the full $\mathrm{K}_4\mathrm{V}_4$ baseline, indicating that the dominant benefits of rotation stem from mitigating key outliers rather than value outliers. Applying rotation to both keys and values provides little additional gain beyond key-only rotation, reinforcing that keys are the primary target for rotation-based improvements.

Figure 11 examines the effect of *group size configuration* under a fixed $\mathrm{K}_4\mathrm{V}_2$ mixed-precision setting without rotation. Group size (gs) determines the block granularity of quantization: smaller groups offer finer scaling at the cost of increased metadata and computation. The results reveal a clear asymmetry between keys and values. Reducing group size for *keys* consistently improves downstream accuracy across CoQA, GSM8K, EQ-Bench, and LongBench, with $\mathrm{gs}_K = 32$ achieving the best overall performance. This reflects the higher sensitivity of key caches to quantization distortion. In contrast, increasing value group size to 64 or 128 has a negligible impact on accuracy while reducing overhead, consistent with their lower sensitivity.

Together, these findings provide practical guidance for combining geometry-driven bit allocation with rotation and grouping strategies: rotation should primarily target *keys*, while keys benefit from finer group granularity and higher precision; values, on the other hand, can use coarser grouping and lower precision with minimal accuracy loss.

D Reproducibility and Resources

Inference was performed using the Hugging Face Transformers (Wolf et al., 2020) and Accelerate (Gugger et al., 2022) with FlashAttention (Dao et al., 2022; Dao, 2024). We integrated both Quanto and HQQ into the Language Model Evaluation Harness (Gao et al., 2024) to enable systematic and reproducible evaluation of model performance under varying quantization schemes. Quantization error is measured by MSE, and confidence intervals use Bayesian variance. (Hariri et al., 2025). Evaluations were executed on two High-performance Computing (HPC) clusters, detailed in Table 16.

Table 11: **Rotation-based Outlier Redistribution Integration on GSM8K - Exact Match (Flexible).** Downstream accuracy under different key-value bit allocations and rotation scopes. Group size is fixed to 64 for both keys and values.

Model	Rotation	$\mathbf{K_2V_2}$	$\mathbf{K_2V_4}$	$\mathbf{K_4V_2}$	$\mathbf{K_4V_4}$
Llama 3.1-8B	K+V	52.99	55.80	76.50	76.95
	K only	52.92	57.32	75.44	76.50
	V only	20.70	25.70	75.28	76.95
	none	18.65	24.41	75.21	76.95
Llama 3.2-1B	K+V	1.97	2.43	30.93	33.13
	K only	2.05	3.03	28.43	33.13
	V only	2.05	1.29	28.58	29.57
	none	1.82	2.20	27.22	29.57
Llama 3.2-3B	K+V	40.11	44.12	62.85	64.37
	K only	38.06	45.26	61.18	63.91
	V only	16.60	19.26	63.46	63.91
	none	14.86	21.08	62.93	63.53
Qwen 0.6B	K+V	0.76	0.61	37.83	36.47
	K only	1.52	1.06	36.92	37.68
	V only	1.52	1.67	1.36	2.12
	none	0.83	1.90	1.36	2.05
Qwen 32B	K+V	31.69	33.81	66.49	64.22
	K only	30.10	33.51	63.99	65.28
	V only	6.90	11.07	65.81	63.68
	none	6.67	10.92	64.82	63.76
Qwen 4B	K+V	0.91	0.76	83.78	85.37
	K only	1.14	0.68	82.26	84.76
	V only	0.83	1.14	82.56	83.70
	none	1.14	0.83	81.80	83.40
Qwen 8B	K+V	29.34	36.01	88.17	88.17
	K only	29.57	35.56	87.64	87.95
	V only	1.97	1.21	86.73	87.34
	none	2.27	1.67	86.58	88.02

Table 12: **Rotation-based Outlier Redistribution Integration on GSM8K - Exact Match (Strict).** Downstream accuracy under different key-value bit allocations and rotation scopes. Group size is fixed to 64 for both keys and values.

Model	Rotation	$\mathbf{K_2V_2}$	$\mathbf{K_2V_4}$	$\mathbf{K_4V_2}$	$\mathbf{K_4V_4}$
Llama 3.1-8B	K+V	51.71	55.27	74.37	74.45
	K only	51.93	56.48	73.24	73.92
	V only	19.33	24.49	73.01	74.60
	none	17.97	22.74	72.93	74.60
Llama 3.2-1B	K+V	0.83	1.21	31.01	32.98
	K only	0.68	1.29	29.04	33.06
	V only	0.38	0.61	28.51	29.57
	none	0.45	0.61	27.22	29.34
Llama 3.2-3B	K+V	39.65	43.67	61.94	63.84
	K only	36.47	44.96	60.27	63.46
	V only	16.00	19.26	63.00	63.38
	none	14.40	21.00	62.02	62.85
Qwen 0.6B	K+V	0.00	0.00	38.14	36.92
	K only	0.00	0.00	37.38	37.83
	V only	0.00	0.00	0.15	0.53
	none	0.00	0.00	0.30	0.08
Qwen 32B	K+V	21.99	28.35	74.75	75.59
	K only	21.46	26.38	74.75	75.28
	V only	2.20	5.00	74.00	75.13
	none	1.52	4.93	73.77	74.37
Qwen 4B	K+V	0.23	0.08	84.53	85.60
	K only	0.00	0.08	83.40	84.99
	V only	0.00	0.00	82.49	83.70
	none	0.00	0.08	81.80	83.78
Qwen 8B	K+V	25.32	30.33	87.72	87.87
	K only	25.17	32.15	87.49	87.34
	V only	0.15	0.00	86.43	86.88
	none	0.00	0.08	86.50	87.41

Table 13: **Rotation-based Outlier Redistribution Integration on CoQA - F1 Score.** Downstream accuracy under different key-value bit allocations and rotation scopes. Group size is fixed to 64 for both keys and values.

Model	Rotation	$\mathbf{K}_2\mathbf{V}_2$	$\mathbf{K}_2\mathbf{V}_4$	$\mathbf{K}_4\mathbf{V}_2$	$\mathbf{K}_4\mathbf{V}_4$
Llama 3.1-8B	K+V	77.70	77.83	78.88	78.45
	K only	77.39	78.10	77.93	78.75
	V only	75.46	76.71	79.34	78.76
	none	74.72	76.57	78.10	79.18
Llama 3.2-1B	K+V	50.66	53.55	70.44	70.47
	K only	50.18	53.46	69.40	70.27
	V only	39.50	44.93	69.58	70.52
	none	38.82	44.33	69.53	70.31
Llama 3.2-3B	K+V	78.45	78.78	79.24	79.13
	K only	78.24	78.88	78.71	79.20
	V only	73.25	74.48	78.86	79.27
	none	74.10	74.55	79.14	79.37
Qwen 0.6B	K+V	33.99	34.06	68.92	70.09
	K only	33.88	33.70	69.62	70.08
	V only	29.00	27.39	48.19	48.06
	none	28.89	28.04	46.49	48.41
Qwen 32B	K+V	64.85	67.39	82.46	82.59
	K only	67.16	67.18	82.62	82.55
	V only	77.82	78.55	82.35	82.50
	none	76.80	78.35	82.60	82.77
Qwen 4B	K+V	62.43	63.82	80.16	80.56
	K only	62.09	64.35	80.46	80.63
	V only	56.73	59.26	80.07	80.16
	none	54.93	59.68	80.41	80.19
Qwen 8B	K+V	79.33	79.65	82.32	81.98
	K only	78.27	79.89	81.15	82.09
	V only	67.51	70.41	82.23	82.01
	none	66.86	69.83	81.49	81.99

Table 14: **Rotation-based Outlier Redistribution Integration on CoQA - Exact Match.** Downstream accuracy under different key-value bit allocations and rotation scopes. Group size is fixed to 64 for both keys and values.

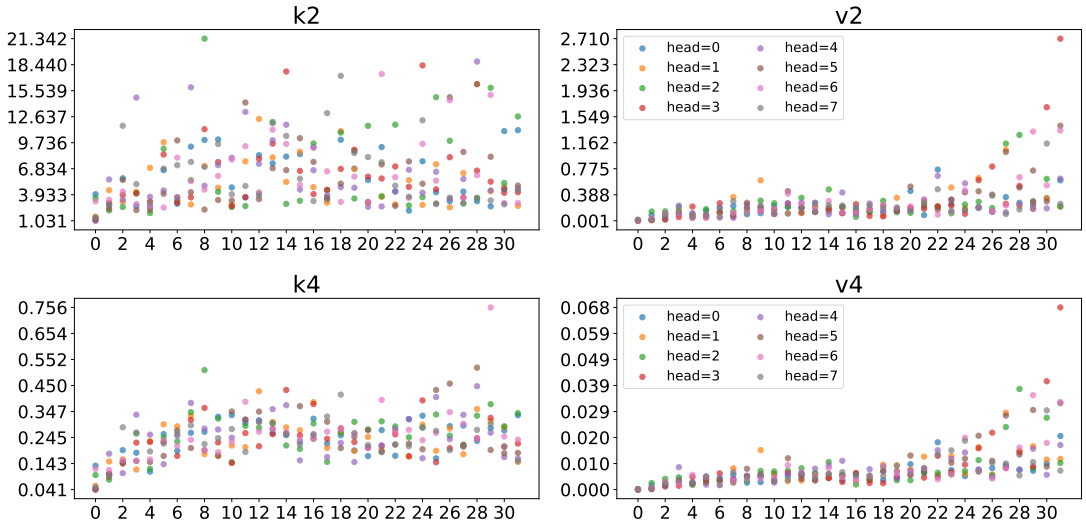
Model	Rotation	$\mathbf{K}_2\mathbf{V}_2$	$\mathbf{K}_2\mathbf{V}_4$	$\mathbf{K}_4\mathbf{V}_2$	$\mathbf{K}_4\mathbf{V}_4$
Llama 3.1-8B	K+V	62.43	63.37	63.25	63.40
	K only	62.12	63.32	63.03	63.65
	V only	58.13	60.12	64.48	63.75
	none	58.12	59.65	62.92	64.45
Llama 3.2-1B	K+V	33.38	37.87	54.78	56.02
	K only	33.72	37.30	54.22	55.32
	V only	25.47	30.37	53.97	55.65
	none	24.42	30.23	54.30	55.45
Llama 3.2-3B	K+V	62.47	62.28	62.83	63.40
	K only	62.12	62.72	62.93	63.53
	V only	56.07	57.48	62.70	63.73
	none	56.45	57.62	63.42	63.77
Qwen 0.6B	K+V	19.67	19.57	56.07	56.73
	K only	18.72	18.18	55.53	56.73
	V only	13.97	14.27	28.18	28.17
	none	14.25	13.80	28.22	28.82
Qwen 32B	K+V	46.97	49.95	70.18	70.77
	K only	50.28	49.42	70.48	70.87
	V only	61.77	64.08	70.02	70.37
	none	61.12	63.92	69.90	70.50
Qwen 4B	K+V	41.28	42.45	65.68	66.28
	K only	41.65	43.45	66.40	66.47
	V only	39.22	40.03	65.43	66.20
	none	37.70	40.22	65.52	66.07
Qwen 8B	K+V	61.70	61.88	68.55	67.73
	K only	60.07	62.40	66.62	68.23
	V only	49.62	51.58	68.10	68.07
	none	48.40	51.65	67.07	68.00

Table 15: **Performance by rotation strategy across tasks.** Values report the mean performance across seven models (0.6B–32B parameters) and four quantization configurations (K_2V_2, K_2V_4, K_4V_2, K_4V_4). The $\pm$ values indicate the variability range across model scales and quantization settings, with larger ranges reflecting higher sensitivity to these factors.

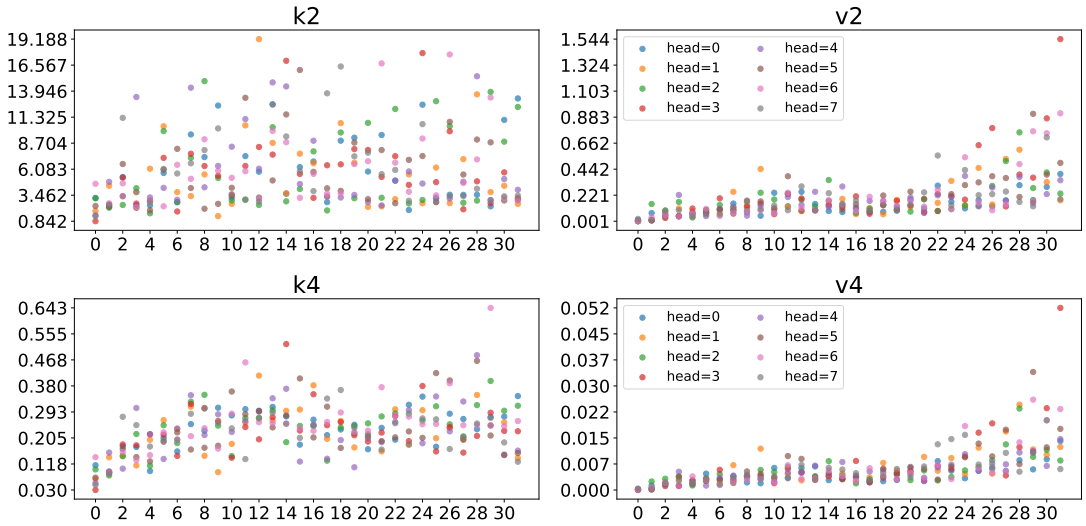
Task	Metric	No Rotation	Value-Only	Key-Only	Both
CoQa	Exact Match	50.10 ± 17.65	51.82 ± 17.22	55.34 ± 14.21	55.39 ± 14.10
Eq Bench	Parseable	56.18 ± 42.57	55.01 ± 41.93	65.73 ± 39.70	64.81 ± 39.19
GSM8K	Exact Match	32.73 ± 33.42	32.96 ± 33.48	43.46 ± 28.98	43.80 ± 29.28
Longbench	ROUGE Score	16.70 ± 11.75	16.54 ± 11.82	17.50 ± 12.83	17.63 ± 12.81

Table 16: Specifications of Two High-Performance Computing (HPC) Clusters Used in This Study

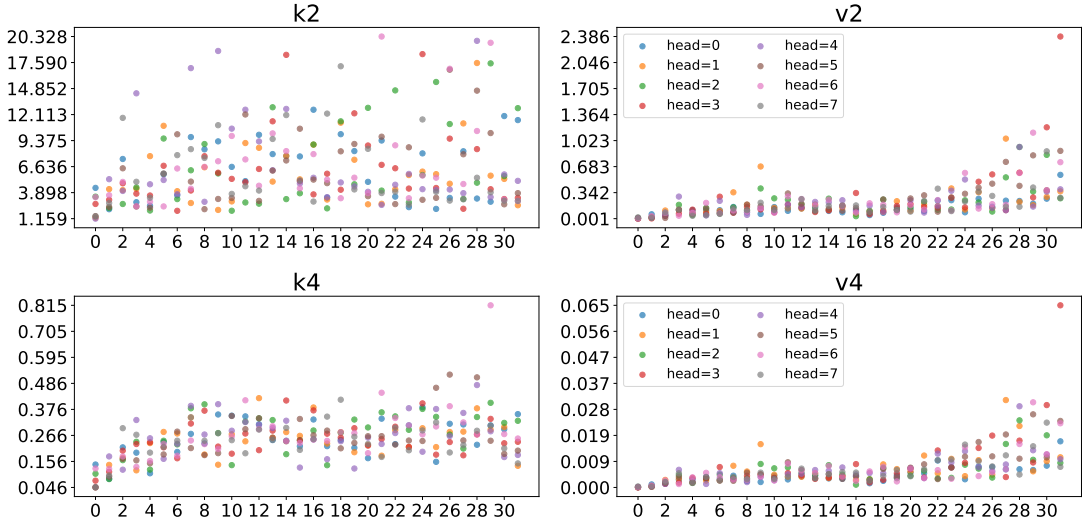
Cluster	Cluster A	Cluster B
Processor	AMD EPYC 7742 $\times$ 2	Intel Xeon Platinum 8468 $\times$ 2
RAM	2048 GB	2048 GB
GPU	NVIDIA A100 $\times$ 8	NVIDIA H200 $\times$ 8
VRAM	80 GB HBM2e	141 GB HBM3e
Scale	5 nodes (40 A100)	5 nodes (40 H200)



(a) C4



(b) MMLU



(c) GSM8k

Figure 9: **Quantization error (MSE) of key and value caches in the Llama 3.1 8B model across 32 layers for (a) C4, (b) MMLU, and (c) GSM8K.** The top row shows 2-bit quantization (K_2, V_2), and the bottom row shows 4-bit quantization (K_4, V_4). Each point corresponds to an attention head within the respective layer. The x-axis denotes the layer index, and the y-axis indicates the quantization error (MSE).

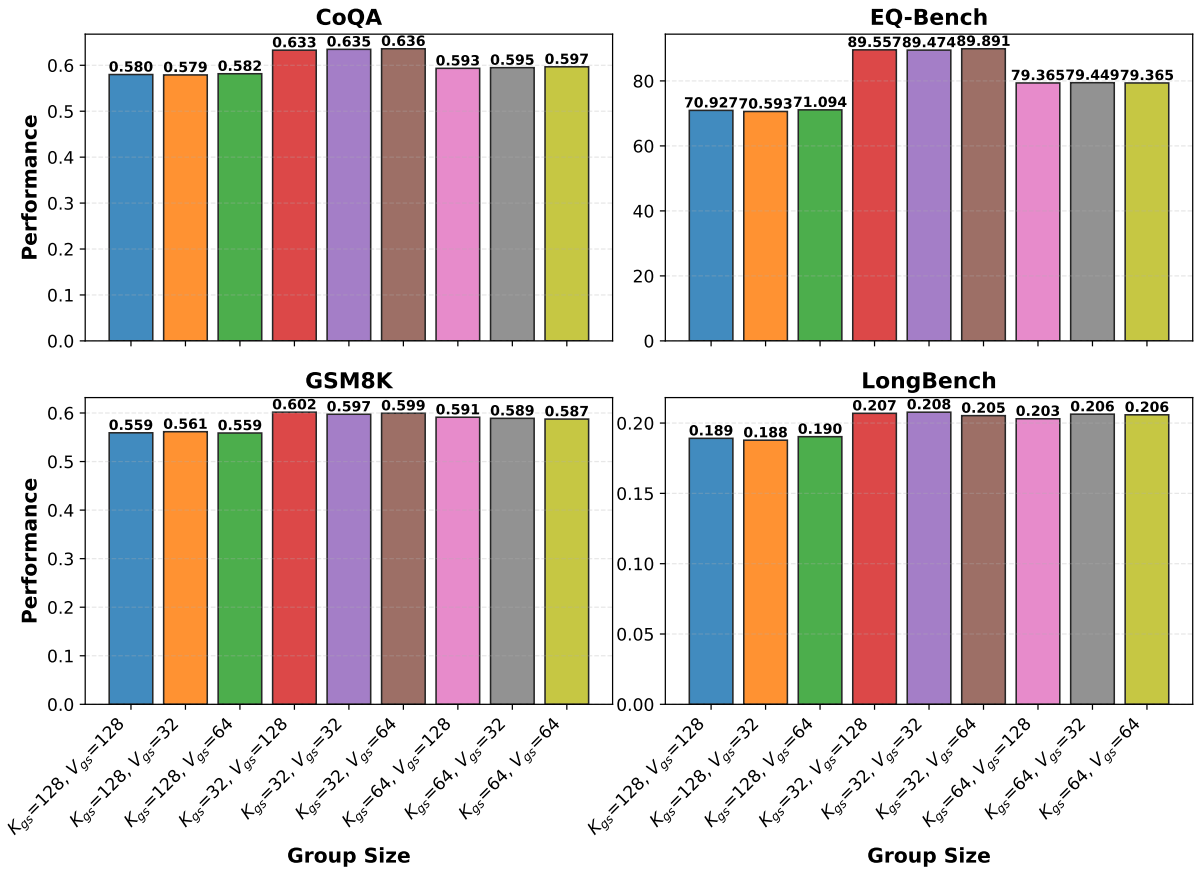


Figure 11: **Effect of group size configurations on downstream accuracy.** Each bar shows performance on CoQA, GSM8K, EQ-BENCH, and LONGBENCH for different key-value grouping combinations ($\text{gs}_K, \text{gs}_V \in \{32, 64, 128\}$) under the $\mathbf{K}_4\mathbf{V}_2$ mixed-precision setting. Smaller group sizes for **keys** consistently improve accuracy across all tasks, while value group size has a smaller but non-negligible effect. Configurations with $\text{gs}_K = 32$ achieve the best overall performance, highlighting the benefit of finer quantization granularity for key caches, which are more sensitive to quantization distortion. In contrast, larger group sizes for values ($\text{gs}_V = 64$ or 128) preserve performance while reducing overhead, aligning with their lower sensitivity.