More for Keys, Less for Values: Adaptive KV Cache Quantization

Mohsen Hariri[1], Lam Nguyen[1], Sixu Chen[1], Shaochen Zhong[2], Qifan Wang[3], Xia Hu[2], Xiaotian Han[1], Vipin Chaudhary[1]

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

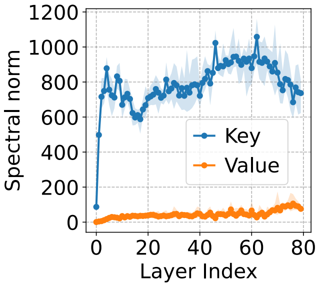
mohsen.hariri@case.edu

Abstract

This paper introduces an information-aware quantization framework that adaptively compresses the key-value (KV) cache in large language models (LLMs). Although prior work has underscored the distinct roles of key and value cache during inference, our systematic analysis—examining singular value distributions, spectral norms, and Frobenius norms—reveals, for the first time, that key matrices consistently exhibit higher norm values and are more sensitive to quantization than value matrices. Furthermore, our theoretical analysis shows that matrices with higher spectral norms amplify quantization errors more significantly. Motivated by these insights, we propose a mixed-precision quantization strategy, *KV-AdaQuant*, which allocates more bit-width for keys and fewer for values since key matrices have higher norm values. With the same total KV bit budget, this approach effectively mitigates error propagation across transformer layers while achieving significant memory savings. Our extensive experiments on multiple LLMs (1B - 70B) demonstrate that our mixed-precision quantization scheme maintains high model accuracy even under aggressive compression. For instance, using 4-bit for Key and 2-bit for Value achieves an accuracy of 75.2%, whereas reversing the assignment—2-bit for Key and 4-bit for Value—yields only 54.7% accuracy. The code is available at https://tinyurl.com/kv-adaquant.

1 Introduction

Large language models (LLMs) have advanced NLP by enabling parallel processing and capturing complex dependencies (Vaswani et al., 2017; Sutskever et al., 2014), which have scaled from GPT-1's 117M (Radford et al., 2018), GPT-2's 1.5B (Radford et al., 2019) to GPT-3's 175B (Brown et al., 2020) and GPT-4's 1.8T parameters (OpenAI et al., 2024). Open-source models like Llama have grown to 405B (Touvron et al., 2023), with Mistral Large (123B) (Mistral-AI, 2023) and DeepSeek V3 (671B) (DeepSeek-AI, 2023) also emerging. In parallel with increasing parameter counts, longer context windows are becoming essential for chain-of-thought reasoning, necessitating an expansion of the KV cache and contributing to memory bottlenecks (Wei et al., 2022). New systems support context windows of up to 1M tokens (DeepMind, 2025), 200k tokens (OpenAI, 2024, 2025), and 130k tokens (DeepSeek-AI et al., 2025). NVIDIA's Blackwell architecture introduces low-precision formats like FP8, FP4, and INT4 (NVIDIA, 2025), emphasizing the need for precise quantization methods. The KV cache is a critical component in these models and its effective operation is paramount and significant.



Size	K_2V_4	K_4V_2
1B	0.06	0.34
8B	0.55	0.75
14B	0.78	0.91
14B	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.

To accommodate larger context windows, the KV cache can become a significant memory bottleneck. One effective solution is to quantize the key and value caches from high-precision formats (e.g., FP32 or BF16) to lower-precision ones (e.g., INT4 or INT2). However, we first discovered that the Key cache generally exhibits higher norms than the Value cache, making it more sensitive to uniform quantization. Since matrices with larger norms are more prone to quantization errors, bit-width allocation should be tailored to each matrix's norm distribution. To confirm this, we conducted experiments with Llama3.3-70B (see Figures 1 (Left) and 2), as the Key cache (blue) consistently shows elevated norm values compared to the Value cache (orange). Furthermore, Figure 1 (Right) demonstrates that the $\mathrm{K}_{(4\mathrm{bit})}\mathrm{V}_{(2\mathrm{bit})}$ scheme significantly outperforms the $\mathrm{K}_{(2\mathrm{bit})}\mathrm{V}_{(4\mathrm{bit})}$ configuration on GSM8K dataset. These results underscore the need for a norm-aware, mixed-precision strategy that allocates higher precision to Keys and lower precision to Values.

Guided by these inspiring observations, we propose a mixed-precision quantization strategy, *KV-AdaQuant*, which allocates different bit-width to more bits for keys and fewer bits for values. since key matrices consistently have higher norm values. With the same total KV bit budget, this approach effectively mitigates error propagation across transformer layers while achieving significant memory savings. *KV-AdaQuant* uses singular values, spectral norms, and Frobenius norms to assign more bits to Keys and fewer to Values, balancing memory savings with preserving critical information. This work makes the following contributions: Guided by these insights, we propose a mixed-precision quantization strategy, *KV-AdaQuant*, which assigns a higher bit-width to key caches and a lower bit-width to value caches—reflecting the consistently higher norm values observed in key matrices. *KV-AdaQuant* leverages metrics such as singular values, spectral norms, and Frobenius norms to identify the critical information in KV cache. By maintaining the same overall KV bit budget, our approach effectively mitigates error propagation across transformer layers while offering significant memory savings. Our contributions include:

- We systematically investigate the KV cache in transformers, uncovering structural patterns in how information is stored and retrieved.
- We introduce *KV-AdaQuant*, an information-aware quantization method that leverages singular value distributions to allocate mixed precision—assigning higher bit-widths to keys and lower bit-widths to values.
- We demonstrate the efficacy of our approach on the GSM8K benchmark, achieving substantial memory savings without compromising performance.

2 Key and Value are Different

This section investigates the differences between key and value caches to enhance quantization strategies.

2.1 Singular Value Distribution of the KV Cache

An extensive collection of models, ranging in size up to 70B parameters (Llama 3.3 70B (Grattafiori et al., 2024)), is examined to explore the singular values of the KV cache. Random batches are sampled from the C4 (Dodge et al., 2021), GSM8K (Cobbe et al., 2021), and MMLU (Hendrycks et al., 2021) datasets, and KV caches are extracted across all attention heads and layers. Figure 3 illustrates the distribution of singular values for the KV cache, starting from the 5th largest singular value onward, highlighting how keys and values diverge. Additionally, Figure 7 in Appendix C presents the full-range distribution, further revealing the discrepancy in larger singular values. Beyond individual singular values, the spectral norm ($\|M\|_2$) and the Frobenius norm ($\|M\|_F$) are computed, providing empirical evidence of the substantial gap in key and value matrix singular values and illustrating how this characteristic of the KV cache can inform the theoretical analysis of quantization error propagation in multi-layer attention networks. This discrepancy underscores the importance of careful precision allocation to mitigate performance degradation while saving more memory in the inference time.

2.2 Rank of KV Cache

The rank of a matrix corresponds to the number of nonzero singular values and determines the effective dimension of its column or row space. In multi-head self-attention, the key (K) and value (V) matrices possess one fixed dimension, d_{head}, and one variable dimension corresponding to the sequence length, seq_len. Consequently, K and V can be analyzed in two distinct regimes: (a) when $\text{seq\_len} < d_{\text{head}}$, and (b) when $\text{seq\_len} \geq d_{\text{head}}$. In the first regime, the matrices are relatively short and thus offer limited opportunities for compression. Accordingly, many existing KV-cache compression methods either avoid compressing in this stage or employ specialized strategies that differ from those used later.

When seq_len $\geq d_{\text{head}}$, the K and V matrices assume rectangular shapes with dimensions seq_len $\times d_{\text{head}}$. Although the exact rank is data-dependent (since $K = XW_K$, where X is drawn from a high-dimensional embedding space), empirical observations suggest that both key and value cache typically approach a rank close to d_{head}. Therefore, in practical settings it is reasonable to assume that there are approximately d_{head} nonzero singular values for both K and V. In the remainder of this section, we focus on this regime to analyze how spectral and Frobenius norm considerations guide our quantization strategy.

2.3 Stability of the Dequantized KV Cache: Error Propagation

The spectral norm of a matrix M, denoted $\|M\|_2$, quantifies the maximum stretching factor that M can impart on any vector. In a multi-layer transformer (e.g., Llama 3.3 (Grattafiori et al., 2024) with 80 layers or Phi-4 (Abdin et al., 2024) with 16 layers), the sequential application of transformations implies that quantization errors introduced at one layer can propagate and amplify in subsequent layers.

To illustrate this effect, consider a simplified layer without normalization or additional scaling (Elhage et al., 2021), where the hidden state at layer $(l+1)$ is given by

$$h_{l+1} = h_l + W_l h_l, \tag{1}$$

Let W_l denote the weight matrix at layer l. Define $\widetilde{W_l}$ as the result of quantizing the KV cache, and W_l as the unquantized version. Then, the local error incurred at layer l can be bounded as

$$\begin{aligned} \|h_{l+1} - \widetilde{h}_{l+1}\|_2 &= \|(W_l - \widetilde{W_l})h_l\|_2 \\ &\leq \|W_l - \widetilde{W_l}\|_2 \|h_l\|_2. \end{aligned} \tag{2}$$

Propagating such errors through multiple layers yields a cumulative effect that is proportional to the product of the spectral norms at each layer. In other words, even small perturbations introduced by quantization can be significantly amplified if the weight matrices exhibit large singular values. Empirical observations indicate that *key* matrices consistently have higher spectral norms than *value* matrices. This suggests that errors in key matrices are more prone to propagate and magnify throughout the transformer stack. As a result, it is advisable to employ less aggressive quantization (e.g., 4-bit rather than 2-bit) for key matrices, while value matrices—characterized by lower spectral norms—can tolerate more aggressive quantization without incurring substantial distortion.

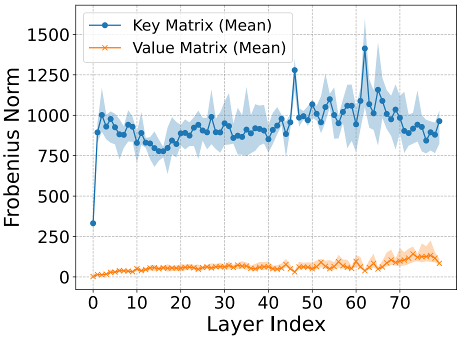


Figure 2: Frobenius norm of key and value cache across layers for the Llama 3.3 70B model on the C4 dataset. The x-axis represents the layer index. The shaded region shows the min-max range across attention heads within each layer, and the solid line indicates the mean norm per layer. Key matrices exhibit significantly higher norms compared to value matrices, highlighting their stronger representational capacity.

2.4 Total Energy of KV Cache (MSE Perspective)

The Frobenius norm of a matrix, denoted by $\|M\|_F$, is defined as the square root of the sum of the squared entries:

$$\|M\|_F = \sqrt{\sum_{i,j} M_{i,j}^2}$$

Equivalently, it can be viewed as the square root of the sum of the eigenvalues of $M^\top M$, meaning it captures the total "energy" of M. In the context of mean-squared error (MSE), the Frobenius norm provides a direct indicator of how much overall distortion might occur when quantizing the matrix.

Quantization Error and the Frobenius Norm. When a matrix M is quantized to a lower-precision representation $\widetilde{M}$, the quantization error in MSE terms is often measured by the squared Frobenius norm:

$$\|M - \widetilde{M}\|_F^2 = \sum_{i,j} (M_{i,j} - \widetilde{M}_{i,j})^2.$$

A common assumption is that this error scales with both the range of the entries in M and the granularity determined by the bit precision. Formally, one may write

$$\mathbb{E}\big[\|M - \widetilde{M}\|_F^2\big] \;\propto\; \|M\|_F^2 \times 2^{-2b},$$

where b denotes the number of bits used per entry.[1] For matrices with the same dimensions (e.g., key and value cache), the MSE and squared Frobenius norm are directly proportional by a factor of $1/mn$. Consequently, when analyzing quantization error for matrices of identical size:

- Minimizing $\|M - \widetilde{M}\|_F^2$ is equivalent to minimizing MSE$(M, \widetilde{M})$, as the proportionality constant $1/mn$ does not affect optimization or relative comparisons.
- The scaling relationship $\mathbb{E}\big[\|M - \widetilde{M}\|_F^2\big] \propto \|M\|_F^2 \times 2^{-2b}$ holds equivalently for MSE, since both metrics share the same dependence on b and $\|M\|_F^2$.

Thus, the Frobenius norm of the quantization and mean square of the quantization error can be used interchangeably for key and value cache, as they encode the same error structure up to a constant scaling factor.

Larger Frobenius Norms Require Higher Precision. As we discussed in 2.1 empirical measurements across multiple transformer models reveal that *key matrices* have significantly larger Frobenius norms than *value matrices*. This disparity suggests that if key matrices are quantized too aggressively (e.g., with fewer bits), the MSE incurred could be disproportionately high. Formally, for two matrices K (key) and V (value), if

$$\|K\|_F \gg \|V\|_F,$$

then allocating the same limited bit precision to both K and V would cause

$$\|K - \widetilde{K}\|_F^2 \gg \|V - \widetilde{V}\|_F^2,$$

since the error term for K is amplified by its larger energy. Furthermore, errors introduced in earlier layers propagate through subsequent layers in a multi-layer transformer architecture, compounding the negative impact of coarse quantization.

3 Method

3.1 Adaptive KV Cache Quantization (*KV-AdaQuant*)

Building on the theoretical findings that matrices with higher spectral norms propagate quantization error more strongly and that larger Frobenius norms incur greater quantization error, along with empirical evidence indicating that key matrices exhibit higher norms than value matrices, a straightforward strategy emerges: quantize keys with higher precision and values with fewer bits. In particular, allocating 4-bit precision to keys and 2-bit precision to values effectively controls the overall mean squared error (Figure 4), reducing memory usage while mitigating error propagation through successive transformer layers.

3.2 Differences between *KV-AdaQuant* and Other Methods

The Hugging Face Transformers library implements a KV cache quantization technique that applies uniform precision to KV cache. Although straightforward, this approach can be suboptimal due to the distinct sensitivities of keys and values to quantization errors. Consequently, assigning the same bit-width to both key and value often introduces notable performance degradation. Furthermore, increasing the bit-width for value matrices offers minimal or no additional benefits, indicating that uniform quantization is not an ideal choice.

Other methods, such as SKVQ (Duanmu et al., 2024) and QAQ (Dong et al., 2024), also propose low-bit quantization for the KV cache, leveraging mixed-precision settings. SKVQ, for example, quantizes the key cache to 2 bits and the value cache to 1.5 bits, demonstrating minimal accuracy loss; however, it relies on fixed hyperparameter configurations that do not deeply analyze the key and value cache' internal characteristics. QAQ similarly introduces quality-adaptive quantization to highlight the heightened sensitivity of key matrices to quantization errors, yet does not thoroughly investigate the underlying matrix properties, making generalization across diverse model architectures and tasks less certain.

By contrast, the proposed approach offers a more general framework of cache quantization that explicitly accounts for the distinct characteristics of key and value cache. This strategy reduces error propagation across transformer layers while achieving considerable memory savings. Additionally, it remains adaptable to further enhancements, such as calibration, grouping, and clustering, allowing the integration of complementary techniques to improve KV cache compression. This adaptability renders the framework orthogonal to existing methods, providing a more comprehensive solution to

[1] The proportionality depends on the quantizer design and the distribution of entries in M. For more details see Appendix A.

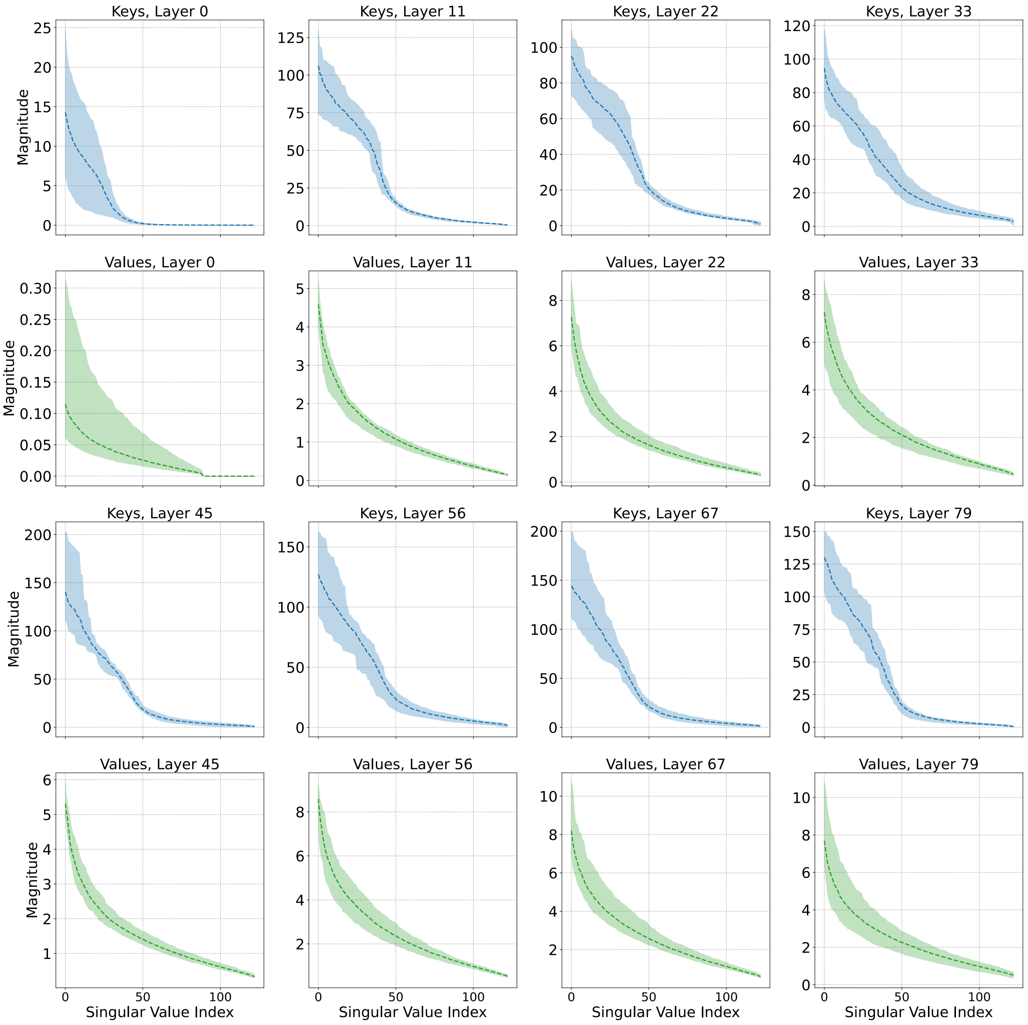


Figure 3: **Singular value distribution of key and value cache** for the Llama 3.3 70B model on the C4 dataset. The x-axis represents the singular value indices, ordered from the 5th largest to the smallest, while the y-axis denotes the magnitude of the singular values. The shaded region indicates the minimum-to-maximum range across attention heads within each layer, and the solid lines represent the mean singular value magnitude at each index. This zoomed-in view reveals that, beyond the largest singular value (spectral norm), key matrices exhibit consistently higher singular values even in the lower ranges, underscoring their greater representational significance compared to value matrices. Consequently, key matrices should be quantized with less aggressive compression. See Figure 7 in Appendix C for the complete range.

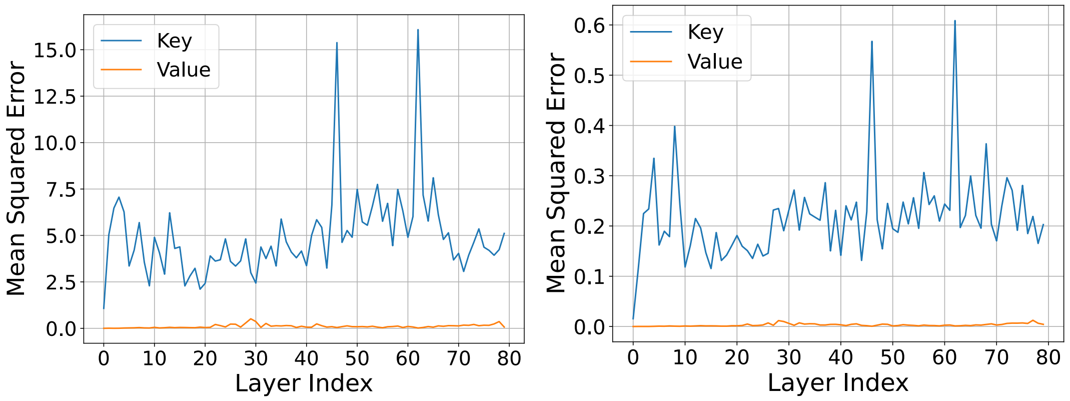
the challenges associated with KV cache quantization.

4 Experiments

A series of experiments is designed to investigate singular value patterns in KV cache and to demonstrate how these patterns can be leveraged to enhance practical quantization strategies. The experimental testbed consists of a diverse set of transformer models. Two distinct data setups are employed: one for analyzing the singular value distributions and another for evaluating the effectiveness of the proposed quantization approach.

4.1 Experiment Settings

Singular Value and Norm Analysis This section examines structural differences between key and value cache by analyzing their singular values and norms. The process begins by feeding random batches (10 samples per batch) from the C4, MMLU, and GSM8K datasets into the model for each query, where every sample is padded to match the longest sequence to facilitate observations across diverse inputs. For each newly generated token (up to 1,000 tokens), the KV cache from each attention mechanism is collected, yielding matrices $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{d_{\text{head}} \times \text{seq\_len}}$ for each head and layer. Singular value decomposition is then per-



(a) Mean4squared quantization error across layers for 2-bit quantization.

(b) Mean squared quantization error across layers for 4-bit quantization.

Figure 4: Quantization error (mean squared error) of key and value cache for the Llama 3.3 70B model on the C4 dataset across all transformer layers. The x-axis represents the layer index, and the y-axis indicates the mean squared quantization error. Key matrices exhibit consistently higher quantization error compared to value matrices, with larger discrepancies observed in lower-bit quantization.

formed offline in `float32` precision using the PyTorch `torch.linalg.svd` operator, and the resulting singular values, along with the spectral norms ($\|\cdot\|_2$) and Frobenius norms ($\|\cdot\|_F$), are computed and averaged across all heads and layers to capture global patterns. This approach reveals how $\|\mathbf{K}\|_2$ and $\|\mathbf{V}\|_2$ evolve in deeper layers, sheds light on the total "energy" in these matrices, and clarifies the distribution of larger and smaller singular values, offering insights into their susceptibility to quantization errors.

Quantization Error Hardware typically supports 2-bit, 4-bit, and 8-bit quantization, but in theory, any bit-width can balance compression and accuracy. We measure quantization error across bit-widths from 2 to 8 by quantizing and dequantizing key-value matrices, finding that MSE (and thus the Frobenius norm) behaves similarly across all bit-widths. As shown in Figure 5, the key matrix consistently exhibits higher error.

Bit-Width Configurations. The effect of different bit-widths used to represent each entry in $\mathbf{K}$ and $\mathbf{V}$ is quantified. Uniform settings include bit-widths ranging from 2 to 8 bits for *both* keys and values (e.g., $(2, 2)$, $(4, 4)$, or $(8, 8)$). Mixed-precision settings compare $(2, 4)$ and $(4, 2)$, assigning 2 bits to either values or keys and 4 bits to the other.

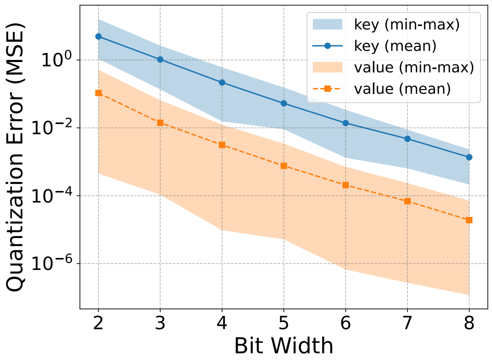


Figure 5: Mean squared error (MSE) of KV cache quantization for the Llama 3.3 70B model on the C4 dataset, using quantization bit-widths ranging from 2 to 8. The x-axis represents the quantization bit-width, while the y-axis shows the MSE on a logarithmic scale. A logarithmic scale is used to highlight 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.

Quantization Procedure. Hugging Face's *Optimum* library is used for low-level CUDA kernels. No additional outlier or chunk-based grouping techniques are applied, so as to isolation of the effect of $\mathbf{K}$-vs-$\mathbf{V}$ bit allocation.

Error Metric: MSE. To measure the quantization distortion, we record

$$\mathrm{MSE}(\mathbf{K}, \widetilde{\mathbf{K}}) = \frac{1}{d_{\text{head}} \times \text{seq\_len}} \|\mathbf{K} - \widetilde{\mathbf{K}}\|_F^2,$$

and likewise for $\mathbf{V}$. In addition to per layer and per head comparison, we average MSE across all

Table 1: Quantization error (mean ± std) for key and value cache at 2-bit, 3-bit, and 4-bit quantization (K_i and V_i) across different datasets.

Dataset	Model	2-bit		3-bit		4-bit	
		K_2	V_2	K_3	V_3	K_4	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
	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
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

heads, layers, and tokens in a batch. This approach yields a global view of how severely each bit-width compresses $\mathbf{K}$ vs. $\mathbf{V}$.

Experimental Protocol. For each model and dataset subset (C4, MMLU, GSM8K):

1. Run a forward pass to produce unquantized KV caches in bfloat16.
2. Quantize to a chosen bit-width, then immediately dequantize to bfloat16.
3. Compare the dequantized version $\widetilde{\mathbf{K}}/\widetilde{\mathbf{V}}$ to the original $\mathbf{K}/\mathbf{V}$ in MSE terms.
4. Repeat for all bit-width settings of interest.

Downstream Evaluation. While MSE is a reasonable proxy for overall distortion, the ultimate question is whether the quantized KV cache preserves model quality on realistic tasks. We therefore evaluate the final model performance on GSM8K in two prompting modes: 1-shot and 8-shot prompting using 4 quantization settings:

- $\mathbf{K}_{(\mathbf{2bit})}\mathbf{V}_{(\mathbf{2bit})}$: Aggressive uniform quantization.
- $\mathbf{K}_{(\mathbf{4bit})}\mathbf{V}_{(\mathbf{4bit})}$: More conservative uniform quantization.
- $\mathbf{K}_{(\mathbf{2bit})}\mathbf{V}_{(\mathbf{4bit})}$ *vs.* $\mathbf{K}_{(\mathbf{4bit})}\mathbf{V}_{(\mathbf{2bit})}$: Mixed-precision assignments, giving 2 bits to either $\mathbf{K}$ or $\mathbf{V}$ and 4 bits to the other.

4.2 Experiment Results

Table 1 shows that across datasets and models, quantization error for the key matrices is consistently and substantially higher than that for the value matrices. For instance, in the MMLU dataset, Llama3.2-1B exhibits an error of 4.85 ± 1.04 for $\mathbf{K}_2$ versus only 0.13 ± 0.10 for $\mathbf{V}_2$. This trend persists for both the 3-bit and 4-bit settings, where increasing the bit-width reduces the error for both matrices, yet the keys still incur a notably higher error. Similar patterns are evident in the C4 and GSM8K datasets, highlighting that key matrices are inherently more sensitive to quantization noise. Furthermore, Figure 6 in the Appendix B illustrates the quantization error per attention head in each layer for the Llama3.1-8B model on C4, MMLU, and GSM8K, reinforcing our observation (see Appendix D for extended results). Table 2 shows ac-

Table 2: GSM8K 1-shot and 8-shot performance of various models under different key-value quantization settings. $\mathbf{K}_i\mathbf{V}_j$ denotes i-bit quantization for the K cache and j-bit for the V cache. The $\mathbf{K}_{(\mathbf{4bit})}\mathbf{V}_{(\mathbf{2bit})}$ column, highlighted below, represents our main finding.

	1-shot				8-shot			
Model	$\mathbf{K_2V_2}$	$\mathbf{K_2V_4}$	$\mathbf{K_4V_2}$	$\mathbf{K_4V_4}$	$\mathbf{K_2V_2}$	$\mathbf{K_2V_4}$	$\mathbf{K_4V_2}$	$\mathbf{K_4V_4}$
Llama3.2-1B-it	0.033	0.035	0.338	0.357	0.031	0.031	0.289	0.369
Llama3.1-8B-it	0.511	0.547	0.752	0.754	0.408	0.441	0.770	0.782
Phi4-14B	0.759	0.783	0.913	0.923	0.771	0.815	0.927	0.931
DeepSeekR1Q-14B	0.772	0.775	0.865	0.867	0.763	0.792	0.876	0.875

curacy on GSM8K under 1-shot and 8-shot prompting. Uniform $\mathrm{K}_{(2\mathrm{bit})}\mathrm{V}_{(2\mathrm{bit})}$ settings degrade accuracy noticeably, especially in the longer 8-shot context. By contrast, $\mathrm{K}_{(4\mathrm{bit})}\mathrm{V}_{(2\mathrm{bit})}$ nearly matches the unquantized baseline, confirming that providing higher precision to keys preserves model quality while still compressing values aggressively. The reversed assignment $\mathrm{K}_{(2\mathrm{bit})}\mathrm{V}_{(4\mathrm{bit})}$ consistently underperforms, underscoring that allocating fewer bits to key matrices adversely affects accuracy.

5 Related Work

Model Quantization. for large models such as LLMs (Wan et al., 2024), which is why post-training quantization (PTQ) techniques are generally preferred for quantizing these models.

Quantization reduces the computational cost of neural network inference by lowering model bit-precision (Han et al., 2016; Nagel et al., 2021). It can be divided into Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT). PTQ requires no re-training or labeled data (Jacob et al., 2017; Nagel et al., 2019) and is preferred for large models like LLMs, whereas QAT demands fine-tuning with labeled data (Nagel et al., 2021; Gholami et al., 2022) but scales poorly for such models (Wan et al., 2024).

LLMs Quantization. Researchers consider three post-training quantization settings for LLMs: weight-only, weight-activation, and KV cache quantization. Weight-only quantization focuses solely on weights (Frantar et al., 2023a,b; Lin et al., 2024), while weight-activation quantization targets both weights and activations (Dettmers et al., 2022; Xiao et al., 2023; Shao et al., 2024). Since longer inputs inflate the KV cache's memory overhead (Pope et al., 2023), KV cache quantization offers memory savings similar to weight-activation methods while maintaining performance near that of weight-only quantization (Yue et al., 2024).

KV Cache Quantization. Recent KV cache quantization methods reduce LLM memory and accelerate inference, falling into three categories: outlier redistribution, fixed-precision, and mixed-precision (Li et al., 2025). Outliers in KV cache are addressed by redistributing them or applying smoothing transformations (Xiao et al., 2023; Lin et al., 2024; Shao et al., 2024). Fixed-precision methods use a uniform bit-width (Yao et al., 2022; Sheng et al., 2023; Liu et al., 2024) but can overlook token importance and outliers. In contrast, mixed-precision allocates higher precision to critical tokens and lower precision elsewhere (Hooper et al., 2024; Yue et al., 2024; Li et al., 2024; Dong et al., 2024; Duanmu et al., 2024). Existing methods like SKVQ and QAQ rely on specific engineering solutions or predefined parameters, whereas *KV-AdaQuant* uses a principled, analysis-driven framework that leverages the intrinsic properties of key and value caches for a more robust and adaptable quantization approach.

6 Conclusion

In this work, we systematically analyze KV cache matrices from the perspectives of linear algebra and information theory to uncover distinct patterns in the information carried by these matrices. Building on the singular value distribution of the KV cache, we propose an information-aware quantization framework for KV cache compression. We evaluate the effectiveness of our approach on two widely used datasets, GSM8K, demonstrating its practical impact. While our current study focuses on the per-token quantization methods, the proposed framework is general and can be applied to any key-value quantization technique.

Limitations

Although our information-aware quantization framework is effective, there are aspects that warrant further consideration. First, our experiments utilize a maximum context length of 2K tokens. While this setting aligns with many practical inference scenarios, it may not fully capture the demands of tasks requiring larger context windows (e.g., 4K or 8K tokens). Extending our approach to support longer contexts could reveal additional challenges, such as increased quantization sensitivity and greater memory management overhead.

Ethical Considerations

Our research aims to reduce the memory footprint and computational overhead of LLM in inference time through KV cache quantization. Such advancements can potentially lower energy consumption and expand access to resource-constrained communities, contributing to more environmentally sustainable and inclusive deployment of language modeling based productions. However, the use of proposal KV cache compression method must be carefully monitored to ensure that accuracy tradeoffs do not disproportionately affect critical applications such as healthcare, legal, and financial contexts.

References

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. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

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.

Google DeepMind. 2025. Gemini 2.0 flash. Accessed via DeepMind Technologies.

DeepSeek-AI. 2023. Deepseek-v3 technical report. DeepSeek-V3 scales language models to over 600 billion parameters, pushing the limits of language understanding and generation capabilities.

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. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*.

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, 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)*.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2022. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*. Chapman and Hall/CRC.

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.

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.

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)*.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2017. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*.

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

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.

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.

Mistral-AI. 2023. Mistral large 2. Mistral Large is a state-of-the-art language model reportedly featuring 30 billion parameters.

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.

Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. 2019. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2019)*.

NVIDIA. 2025. Blackwell architecture.

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.

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).

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.

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, Daniel Y. Fu, Zhiqiang Xie, Beidi Chen, Clark Barrett, Joseph E. Gonzalez, Percy Liang, Christopher Ré, Ion Stoica, and Ce Zhang. 2023. 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)*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

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.

Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, Mosharaf Chowdhury, and Mi Zhang. 2024. Efficient large language models: A survey. *Transactions on Machine Learning Research (TMLR 2024)*.

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)*.

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.

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.

A Quantization and Dequantization Error Bounds

In this appendix we establishes 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 the 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.

Preliminaries and Notation

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

$$\begin{aligned} Q_{ij} &= \operatorname{round}\!\Big(\frac{A_{ij}}{\alpha}\Big), \\ \text{with}\quad Q_{ij} &\in \{-2^{b-1},\ldots,2^{b-1}-1\}. \end{aligned} \tag{3}$$

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.

Spectral Norm Bound

Theorem 1 (Spectral Norm Quantization Error)**.** *Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{N}$ be given. Consider two's-complement quantization with scale factor*

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

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:

$$\begin{aligned} M \le \|A-\widehat{A}\|_2 &\le \sqrt{m\,n}\,\frac{M}{2(2^{b-1}-1)} \\ &\le \sqrt{m\,n}\,\frac{\|A\|_2}{2(2^{b-1}-1)}. \end{aligned} \tag{4}$$

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

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

□

Frobenius Norm Bound

Theorem 2 (Frobenius Norm Quantization Error). *Under the same setup, the Frobenius norm of the quantization error satisfies*

$$\begin{aligned}\|A - \widehat{A}\|_F \;&\le\; \sqrt{m\,n}\,\tfrac{M}{2(2^{b-1}-1)} \\ &\le\; \sqrt{m\,n}\,\tfrac{\|A\|_F}{2(2^{b-1}-1)}.\end{aligned} \tag{5}$$

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:

$$\begin{aligned}\|A - \widehat{A}\|_2 &\lesssim \frac{\sqrt{m\,n}}{2^b}\,\|A\|_2, \\ \|A - \widehat{A}\|_F &\lesssim \frac{\sqrt{m\,n}}{2^b}\,\|A\|_F.\end{aligned} \tag{6}$$

Quantizing KV Cache

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

$$\begin{aligned} \|\mathbf{E}_V\|_2 &\lesssim \frac{\sqrt{L\, d_{\text{head}}}}{2^{b_V}} \|\mathbf{V}\|_2, \\ \|\mathbf{E}_K\|_2 &\lesssim \frac{\sqrt{L\, d_{\text{head}}}}{2^{b_K}} \|\mathbf{K}\|_2. \end{aligned} \tag{7}$$

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

Similarly, the Frobenius norm error bounds are given by

$$\begin{aligned} \|\mathbf{E}_V\|_F &\lesssim \frac{\sqrt{L\, d_{\text{head}}}}{2^{b_V}} \|\mathbf{V}\|_F, \\ \|\mathbf{E}_K\|_F &\lesssim \frac{\sqrt{L\, d_{\text{head}}}}{2^{b_K}} \|\mathbf{K}\|_F. \end{aligned} \tag{8}$$

To achieve comparable Frobenius errors,

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

which implies

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

and therefore,

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

Again, because $\|\mathbf{V}\|_F < \|\mathbf{K}\|_F$, it follows that $b_K > b_V$.

B Quantization Error Across Layers and Heads

Figure 6 displays the MSE of the quantized key and value caches in the Llama 3.1 8B model, evaluated layer by layer across 32 layers and differentiated by attention head. Results are presented for three datasets: C4, MMLU, and GSM8k. Each subfigure illustrates the 2-bit and 4-bit quantization configurations for the key and value caches. The error patterns vary across layers, suggesting that certain layers or heads may be more sensitive to aggressive quantization; however, the overall pattern indicates that assigning 4-bit to the key cache is comparable to assigning 2-bit to the value cache.

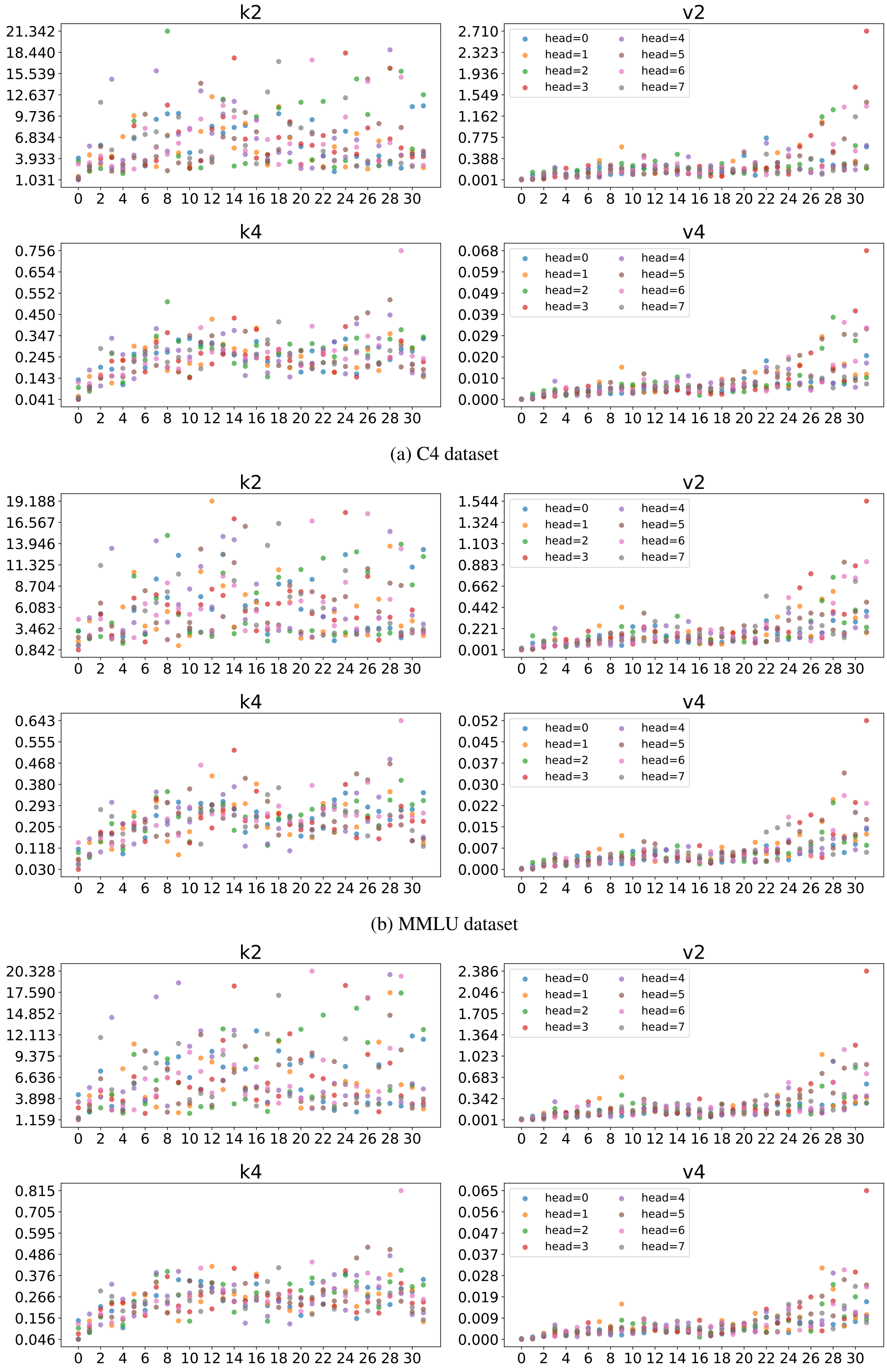
C Extended Visualization of Singular Value Distributions

Figure 7 provides the full-range view of the singular value distribution for the key and value cache at various layers of the Llama 3.3 70B model on the C4 dataset. The horizontal axis enumerates the singular values in descending order, beginning with the largest (i.e., the spectral norm), and the vertical axis measures the corresponding magnitudes. The shaded region captures the minimum-to-maximum range of singular values across the attention heads within each layer, while the solid curves depict the mean singular value at each rank.

This figure underscores two main observations. First, key matrices systematically exhibit higher leading singular values than value matrices. In particular, the first singular value (the spectral norm) for key matrices is consistently larger, confirming that the key matrices possess a higher spectral norm. Second, this behavior (significant gap) remains evident across the entire spectrum of singular values, implying that the sum of the squares of these values (i.e., the Frobenius norm) is also larger for key matrices. Consequently, the stronger representational capacity of key matrices, as reflected by both their spectral and Frobenius norms, renders them more sensitive to quantization errors.

D Extended Quantization Error Results

Tables 3, 4, and 5 present extended quantization error results across a wider range of models and settings for the MMLU, C4, and GSM8K datasets, respectively. These tables offer a detailed breakdown of the error metrics under various bit-width configurations, complementing the analysis presented in the main text and in Table 1.



(a) C4 dataset

(b) MMLU dataset

(c) GSM8k dataset

Figure 6: Quantization error (MSE) of K and V caches in the Llama 3.1 8B model across 32 layers for the (a) C4, (b) MMLU, and (c) GSM8k datasets. The top row in each plot shows 2-bit quantization (k2, v2), while the bottom row shows 4-bit quantization (k4, v4). Each point represents a different attention head in the corresponding layer. The x-axis indicates the layer index, and the y-axis shows the quantization error in MSE.

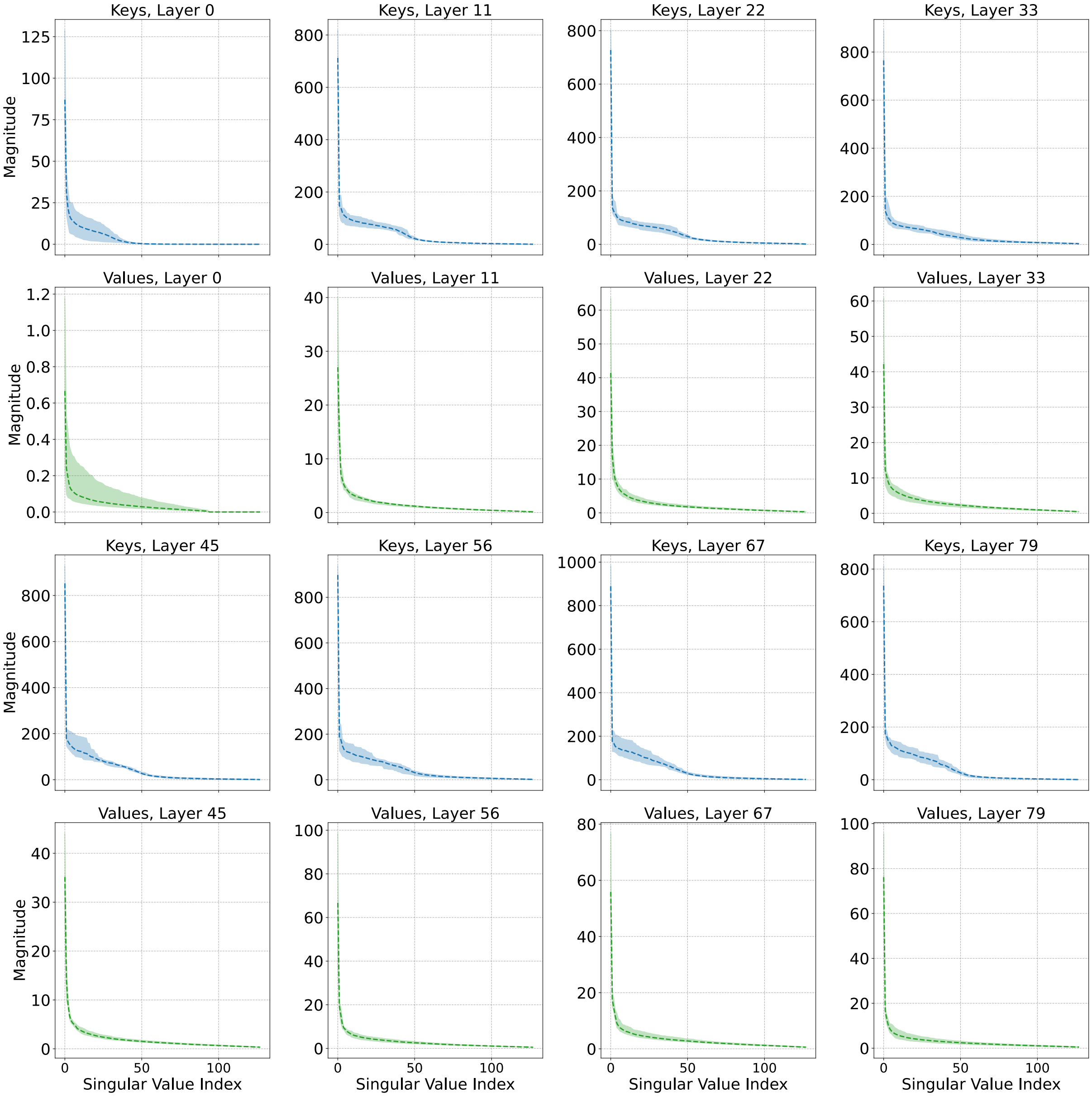


Figure 7: **Complete singular value distribution of key and value cache** for the Llama 3.3 70B model on the C4 dataset. The x-axis denotes the singular value indices, ordered from the largest (spectral norm) to the smallest, while the y-axis represents the corresponding magnitudes. The shaded region 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 3: 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 MMLU dataset.

Model	K2	V2	K3	V3	K4	V4
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
Memotron3.1-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 4: 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 C4 dataset.

Model	K2	V2	K3	V3	K4	V4
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
Memotron3.1-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 5: 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.

Model	K2	V2	K3	V3	K4	V4
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
Memotron3.1-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