Efficient Multimodal Large Language Model via Dynamic KV Cache Quantization

Jiahao Fan[1], **Chien-Ming Chen**[2*]

[1]University of Sydney, Australia
[2]Nanjing University of Information Science and Technology, China
jfan5489@uni.sydney.edu.au, fanjh3@mail.sustech.edu.cn

Abstract

Multimodal large language models (LMMs) have demonstrated remarkable capabilities across diverse vision-language tasks, including image captioning, visual question answering, and text-image retrieval. However, their computational complexity and memory footprint, particularly in the key-value (KV) cache during inference, pose significant challenges for real-time deployment, especially on resource-constrained devices. In this paper, we propose *Dynamic KV Cache Quantization*, a novel quantization strategy tailored for multimodal LMMs. Our approach applies per-channel quantization to K and per-token quantization to V, leveraging their respective statistical distributions to optimize precision allocation. Additionally, we introduce an adaptive token and channel recording mechanism that dynamically adjusts quantization parameters based on real-time distribution tracking, effectively mitigating the impact of outliers. To further enhance compression efficiency, we implement fine-grained grouping, which partitions KV tensors into localized subgroups, enabling more adaptive quantization. Experimental results on LLaVA-1.5 (7B/13B) and Qwen-VL across multiple multimodal benchmarks demonstrate that our method significantly outperforms existing KV-cache quantization approaches, achieving a superior trade-off between memory efficiency and model accuracy.

1 Introduction

Multimodal large language models (LMMs) have emerged as transformative AI systems capable of processing and reasoning across multiple data modalities, including text and images. These models integrate large language models (LLMs) with visual encoders such as CLIP-ViT, allowing them to generate rich multimodal representations that enable tasks such as image captioning, visual question answering, and text-image retrieval (Radford et al. 2021; Liu et al. 2023). Recent advancements in large vision-language models (LVLMs), including LLaVA, Gemini, GPT-4V, and BLIP-2 (Liu et al. 2023; Zhu et al. 2023; Yin et al. 2023; Zhang et al. 2024), have significantly improved zero-shot reasoning and multimodal understanding, demonstrating strong generalization across diverse domains (OpenAI 2023; Team et al. 2023). However, despite their remarkable capabilities, the deployment of these models for real-time applications remains constrained by their high computational and memory requirements, particularly during autoregressive inference.

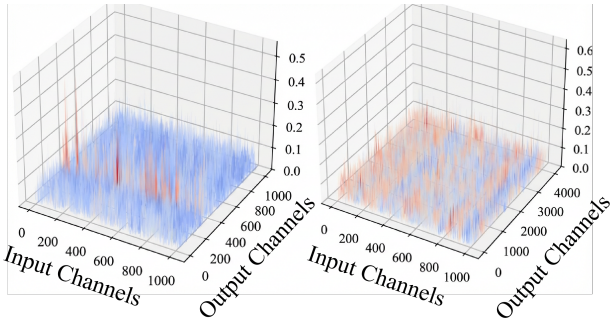

Figure 1: Per-channel (left) and Per-token (right) distribution of K and V values in LLaVA-1.5-7B.

A primary contributor to this challenge is the exponential growth of the key-value (KV) cache in transformer-based models. During autoregressive decoding, each attention layer in an LLM generates and stores key-value pairs for every token in the sequence. Without compression, this KV cache scales linearly with sequence length and batch size, leading to significant memory and bandwidth demands that quickly exceed the capacity of available hardware, particularly on edge devices or GPUs with constrained memory. To address this issue, KV cache quantization has emerged as a promising solution. By reducing the numerical precision of stored key-value pairs (e.g., from FP32 to INT8 or INT4), quantization can offer up to 4× or greater memory savings while maintaining inference efficiency. This enables LLMs to operate efficiently on resource-constrained hardware without requiring extensive architectural modifications.

Despite its success in accelerating text-based LLMs, existing KV cache quantization techniques face fundamental limitations when applied to multimodal models. Unlike pure language tasks, where token distributions exhibit relatively stable patterns, multimodal models process highly hetero-

*Corresponding author.

geneous data, leading to increased variance and outliers in key-value representations (see Figure 1). The presence of extreme values in KV tensors is particularly pronounced in the visual domain, where high-dimensional embeddings contain irregular distributions and anomalous activations that are not well-handled by conventional quantization methods. Existing KV cache quantization approaches, such as fixed-precision quantization, mixed-precision quantization, and outlier redistribution, often assume a relatively smooth distribution of KV values, making them suboptimal for multimodal architectures. When applied to LVLMs, these methods can introduce significant quantization errors, leading to degraded performance in vision-language tasks, particularly those requiring precise spatial and semantic reasoning.

To overcome these limitations, we propose a novel KV cache quantization framework specifically optimized for multimodal models. We introduce a dynamic three-component framework that adjusts quantization strategies based on the statistical properties of key and value tensors: **(1) we propose Dynamic KV Cache Quantization,** where K-cache follows per-channel grouping and V-cache follows per-token grouping, optimizing precision allocation to reduce memory overhead while preserving multimodal reasoning capabilities. For K, we compute independent scaling factors and zero-points for each channel, ensuring that each feature dimension retains its relative importance, preventing information loss due to globally applied scaling. For V, we apply token-wise scaling factors and zero-points, accommodating local variations in contextual embeddings and allowing fine-grained precision adaptation. This targeted approach results in a superior trade-off between efficiency and accuracy, particularly under extreme compression conditions like 4-bit quantization. To further enhance quantization robustness, **(2) we introduce a dynamic token and channel recording mechanism** to track value distributions and mitigate the effects of outliers. Multimodal embeddings, particularly those derived from visual features, often contain outliers that introduce significant variance in the KV-cache, leading to distortions under uniform quantization. To address this, we develop online token and channel recorders that update value distributions using exponential moving averages (EMAs), ensuring stable estimates of statistical properties over time. These statistics allow for the dynamic adjustment of quantization parameters, preventing extreme values from distorting the compression process. We also introduce a learned scaling vector that normalizes values before quantization, reducing precision loss due to outliers. This transformation ensures that both K and V embeddings retain their structural and contextual integrity despite compression. Additionally, we implement an auto-relaxation mechanism that selectively adjusts scaling factors based on per-channel variance, optimizing the balance between compression efficiency and information retention. To further refine quantization granularity, **(3) we introduce fine-grained grouping,** dynamically partitioning the KV-cache into localized subgroups for adaptive quantization. Traditional quantization techniques apply a uniform compression scale across the entire KV-cache, but multimodal embeddings often exhibit non-uniform variance distributions, where certain dimensions contain extreme values that distort quantization scales. To address this, we explore two primary grouping strategies: per-channel grouping, where sequences are aggregated into groups along the sequence dimension, and per-token grouping, where feature channels per token are grouped together to reduce redundancy. The number of groups is allocated based on the proportion of outliers, ensuring that dimensions with higher variance receive finer quantization granularity while smoother distributions are compressed more aggressively. Each subgroup undergoes independent quantization, preventing global outliers from dominating the overall quantization range. By structuring quantization around localized feature distributions, we reduce precision redundancy and ensure that dimensions with similar statistical properties are compressed together.

To validate the effectiveness of our approach, we conduct extensive experiments on multiple LLaVA architectures, including v1.5-7B, v1.5-13B, and Qwen-VL. We evaluate our method across a diverse range of multimodal benchmarks, with particular emphasis on challenging datasets such as VQAv2 and TextVQA, which require sophisticated integration of textual and visual information. Our experimental framework assesses performance across six key metrics, enabling a comprehensive analysis of the trade-offs between efficiency and accuracy. The results demonstrate that our proposed KV-cache quantization techniques significantly improve inference efficiency while maintaining, and in some cases enhancing, model performance compared to existing compression methods. By addressing the limitations of conventional KV-cache quantization in multimodal settings, our work establishes a new state-of-the-art in efficient multimodal model deployment. Our findings provide valuable insights into optimizing memory usage in vision-language models and pave the way for future research on scalable multimodal AI systems that can operate effectively under computational constraints.

2 Related Works

2.1 Large Multimodal Models

Large Language Models (LLMs) such as GPT-4 (OpenAI 2023), LLaMA (Touvron et al. 2023), Mistral (Jiang et al. 2023), and Gemini (Team et al. 2023) have demonstrated impressive reasoning capabilities in text-based tasks. To extend these capabilities to images, Large Multimodal Models (LMMs) (Liu et al. 2023; Zhu et al. 2023; Yin et al. 2023; Zhang et al. 2024) combine a vision encoder with a pre-trained LLM to generate text responses given an image and an associated question. Among these models, LLaVA (Large Language and Vision Assistant) uses a relatively simple Multilayer Perceptron (MLP) as the projector to align the image and text information, and establishes the state-of-art performances on over 10 multi-modal benchmarks. Other notable works include MobileVLM (Chu et al. 2023), TinyGPT-V (Yuan, Li, and Sun 2023), and MoE-LLaVA (Lin et al. 2024), which focus on reducing model size and computational costs. While significant research effort has focused on architectural improvements and model scaling for LMMs, comparatively less attention has been paid to opti-

mizing these models for resource-constrained environments. Most existing efficiency-focused approaches often overlook a critical bottleneck in multimodal inference: the KV-cache memory requirements. This oversight is particularly significant because multimodal inputs typically involve long sequences—combining both visual tokens from images and textual tokens—resulting in substantially larger KV-caches compared to text-only models. Our work specifically targets this overlooked aspect of multimodal efficiency, focusing on optimizing the KV-cache without compromising the model's reasoning capabilities across diverse visual and textual tasks.

2.2 KV Cache Compression

The Key-Value (KV) cache in LLMs plays a crucial role in storing intermediate results during model inference, but it requires substantial memory, especially with longer sequences or larger batch sizes (Brandon et al. 2024; Wan et al. 2024). This memory demand makes it challenging to deploy such models on resource-constrained devices. Therefore, compressing the KV cache is essential to reduce memory usage. Quantization is one effective method which involves reducing the precision of the matrices from higher bandwidth to lower bandwidth representations. Key-Value (KV) cache quantization is emerging as a highly promising solution to address the memory and computational bottlenecks in LLMs. KVQuant (Hooper et al. 2024) proposes to quantize the Keys per channel before applying the RoPE operations and to quantize the Values per token. While KVQuant introduces the insight that Keys and Values require different quantization approaches, it applies uniform quantization schemes that fail to account for the varying importance of different dimensions and tokens in multimodal contexts. KIVI (Liu et al. 2024) retains the most recent Keys and Values in full precision, while quantizing older KVs. This temporal approach offers benefits for language tasks with recency bias but proves insufficient for multimodal reasoning where information from early visual tokens often remains critical throughout the generation process. GEAR (Kang et al. 2024), MiKV (Yang et al. 2024), ZipCache (He et al. 2024b) and ZIPVL (He et al. 2024a) quantize the KV cache based on the importance of each component to achieve efficient and effective compression. While these approaches move beyond uniform quantization, they still apply generic importance metrics that do not fully address the unique characteristics of multimodal embeddings. Our approach differs fundamentally from these previous methods in three key aspects. First, we implement a specialized token and channel recording mechanism that adapts dynamically to the statistical properties of multimodal inputs, rather than using fixed or manually tuned parameters. This allows our method to accommodate the heterogeneous nature of visual and textual representations more effectively. Second, we introduce fine-grained grouping that automatically estimates the appropriate quantization granularity for different parts of the KV-cache based on their information density, rather than applying predetermined grouping schemes.

2.3 Key-Value Cache in LMM

In transformer-based LMMs, the key-value (KV) cache serves as a crucial component for efficient autoregressive inference. During decoding, each attention layer generates and stores key-value pairs corresponding to every token in the input sequence. Let the input sequence be represented as $\mathbf{X} = \{x_1, x_2, ..., x_T\}$, where T is the sequence length. The self-attention mechanism computes query, key, and value matrices at each layer:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}_V \tag{1}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ are the learnable projection matrices. The attention scores are then computed as:

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right) \tag{2}$$

where d_k is the dimensionality of the key vectors. The output of the attention layer is:

$$\mathbf{O} = \mathbf{A}\mathbf{V} \tag{3}$$

During inference, the keys and values from previous time steps are cached to avoid redundant computations. The KV cache at time step t is represented as:

$$\mathbf{K}^{(t)} = [\mathbf{K}^{(1)}, ..., \mathbf{K}^{(t)}], \quad \mathbf{V}^{(t)} = [\mathbf{V}^{(1)}, ..., \mathbf{V}^{(t)}] \tag{4}$$

As the sequence length increases, the KV cache grows linearly, leading to excessive memory usage. This motivates the need for efficient KV cache compression techniques.

2.4 Quantization for KV Cache Compression

Quantization reduces the precision of numerical data representations, enabling substantial memory savings. A widely used approach is uniform asymmetric quantization, which maps high-precision floating-point values to a lower bit-width integer representation while preserving the relative distribution. The quantization-dequantization process for a given tensor X under a B-bit representation is formulated as:

$$Q(X) = \left\lfloor \frac{X - z_X}{s_X} \right\rceil, \quad X' = Q(X) \cdot s_X + z_X \tag{5}$$

where:

$$z_X = \min(X), \quad s_X = \frac{\max(X) - \min(X)}{2^B - 1} \tag{6}$$

Here, z_X represents the zero-point, ensuring that the quantized values remain within the representable range, while s_X is the scaling factor that normalizes the tensor values. While KV cache quantization has been successfully applied to large language models, the presence of outliers in multimodal embeddings—especially in vision-heavy tasks—leads to suboptimal compression and loss of critical information. This necessitates a more adaptive quantization strategy for multimodal models.

3 Methodology

Our approach addresses the memory bottlenecks in multimodal LLMs by optimizing the KV-cache through specialized quantization techniques. We introduce a three-component framework that dynamically adjusts quantization strategies based on the statistical properties of key and value tensors.

3.1 Dynamic KV Cache Quantization

Existing KV cache quantization methods generally apply uniform compression across all stored key-value pairs, assuming that each dimension contributes equally to the model's performance. However, through extensive empirical analysis, we observe that the key (K) and value (V) caches exhibit distinct statistical properties, necessitating different quantization strategies. The key tensors, which encode structural relationships between tokens, display high inter-channel variance, while the value tensors, responsible for propagating contextual meaning, exhibit fluctuations at the per-token level. These fundamental differences suggest that a uniform quantization approach inadequately preserves crucial information, particularly under aggressive bit-width constraints such as 4-bit quantization.

To validate this hypothesis, we conduct a comparative study, applying distinct quantization schemes to K and V caches. Specifically, we test per-channel quantization for K and per-token quantization for V, examining their impact on downstream task performance across multimodal benchmarks. When applying uniform per-token quantization to both K and V, we observe a significant degradation in model accuracy, particularly in vision-language tasks that rely on fine-grained spatial relationships. Under 4-bit quantization, this performance drop becomes particularly pronounced, revealing that critical structural information is lost when quantizing K indiscriminately across tokens. Conversely, applying per-channel quantization to K while maintaining per-token quantization for V mitigates these losses, striking a balance between memory efficiency and retention of essential semantic and structural features.

Based on these observations, we propose Dynamic KV Cache Quantization, where the K-cache follows per-channel grouping and the V-cache follows per-token grouping. This approach leverages the natural statistical distribution of KV tensors to optimize precision allocation, reducing memory overhead while preserving multimodal reasoning capabilities. For per-channel quantization of K, we compute independent scaling factors s_{K_c} and zero-points z_{K_c} for each channel c:

$$s_{K_c} = \frac{\max(K_c) - \min(K_c)}{2^B - 1}, \quad z_{K_c} = \min(K_c) \tag{7}$$

$$Q(K_c) = \left\lfloor \frac{K_c - z_{K_c}}{s_{K_c}} \right\rceil, \quad K_c' = Q(K_c) \cdot s_{K_c} + z_{K_c} \tag{8}$$

This ensures that each channel is scaled independently, preserving inter-channel variance and reducing quantization-induced distortions.

For per-token quantization of V, we apply token-wise scaling factors s_{V_t} and zero-points z_{V_t} for each token t:

$$s_{V_t} = \frac{\max(V_t) - \min(V_t)}{2^B - 1}, \quad z_{V_t} = \min(V_t) \tag{9}$$

$$Q(V_t) = \left\lfloor \frac{V_t - z_{V_t}}{s_{V_t}} \right\rceil, \quad V_t' = Q(V_t) \cdot s_{V_t} + z_{V_t} \tag{10}$$

This approach dynamically adjusts the quantization granularity based on the intrinsic statistical distribution of each tensor type, enabling a more efficient compression scheme that maintains performance parity with higher-bit representations.

The advantages of Dynamic KV Cache Quantization are twofold. First, by applying per-channel quantization to K, we ensure that each feature dimension retains its relative importance, preventing information loss due to globally applied scaling. Second, by quantizing V at the per-token level, we accommodate local variations in contextual embeddings, allowing fine-grained precision adaptation. This method achieves a superior trade-off between efficiency and accuracy, particularly under extreme compression conditions such as 4-bit quantization.

Through this targeted approach, we establish a robust and scalable quantization framework for multimodal large language models. By aligning quantization granularity with intrinsic tensor characteristics, Dynamic KV Cache Quantization significantly reduces memory consumption while maintaining the structural integrity of key embeddings and the contextual richness of value embeddings. This makes it an ideal solution for resource-constrained deployment scenarios, facilitating the efficient execution of multimodal reasoning tasks on edge devices and limited-memory GPUs.

3.2 Token and Channel Recorders and Scaling

To mitigate the effects of outliers on quantization, we introduce a dynamic token and channel recording mechanism that tracks value distributions across different dimensions. Outliers in multimodal embeddings—particularly from visual features—introduce significant variance in the KV-cache, leading to distortions when applying uniform quantization. To address this, we build online token and channel recorders, which continuously count and update value distributions using exponential moving averages (EMAs). This allows for efficient adaptation to changing data distributions during inference. Given a key-value tensor $\mathbf{K}$, we define its per-channel statistical tracking using an EMA-based mean and variance update:

$$\mu_K^{(t)} = \lambda \mu_K^{(t-1)} + (1 - \lambda)\mathbb{E}[\mathbf{K}^{(t)}] \tag{11}$$

$$\sigma_K^{(t)} = \lambda \sigma_K^{(t-1)} + (1 - \lambda)\sqrt{\mathbb{E}[(\mathbf{K}^{(t)} - \mu_K^{(t)})^2]} \tag{12}$$

where λ is the decay factor controlling the update rate and $\mathbb{E}[\cdot]$ denotes the expectation computed over the batch. These moving averages ensure stable estimates of distributional

properties across time steps. Similarly, for value tensors $\mathbf{V}$, we maintain an EMA-based tracking mechanism:

$$\mu_V^{(t)} = \lambda \mu_V^{(t-1)} + (1-\lambda)\mathbb{E}[\mathbf{V}^{(t)}] \tag{13}$$

$$\sigma_V^{(t)} = \lambda \sigma_V^{(t-1)} + (1-\lambda)\sqrt{\mathbb{E}[(\mathbf{V}^{(t)} - \mu_V^{(t)})^2]} \tag{14}$$

Tracking these statistics allows us to dynamically adjust the quantization process to account for variations in token and channel distributions. Using these recorded statistics, we learn a scaling vector that reduces value gaps before quantization, ensuring that extreme values do not dominate the compression process. The scaling vector $\mathbf{S}_K$ is computed as:

$$\mathbf{S}_K = \frac{1}{\sigma_K + \epsilon} \tag{15}$$

where ϵ is a small constant to prevent division by zero. Applying this scaling transformation to $\mathbf{K}$ normalizes its values:

$$\mathbf{K}' = (\mathbf{K} - \mu_K) \odot \mathbf{S}_K \tag{16}$$

where $\odot$ denotes element-wise multiplication. This normalization ensures that the distribution remains more stable, reducing the impact of extreme values. Furthermore, we introduce an auto-relaxation mechanism that selectively adjusts scaling factors based on the variance of each channel:

$$\mathbf{K}'' = \mathbf{K}' \times \alpha + \beta \tag{17}$$

where α and β are learnable scaling parameters that optimize the balance between compression efficiency and information retention. These parameters are tuned to minimize the loss of salient information, adapting dynamically to the characteristics of the KV-cache. For value tensors $\mathbf{V}$, a similar scaling vector $\mathbf{S}_V$ is computed:

$$\mathbf{S}_V = \frac{1}{\sigma_V + \epsilon} \tag{18}$$

$$\mathbf{V}' = (\mathbf{V} - \mu_V) \odot \mathbf{S}_V \tag{19}$$

This transformation ensures that both key and value embeddings are robust against outliers, leading to improved quantization quality.

By incorporating online token and channel recorders, our approach dynamically adapts to the multimodal nature of data, significantly reducing quantization artifacts. The learned scaling vectors effectively mitigate the impact of extreme values, preserving critical information while achieving efficient compression. This enables our quantization framework to outperform existing methods, particularly in vision-language tasks where outliers are more prevalent.

3.3 Fine-Grained Grouping

To further optimize compression and mitigate the adverse effects of extreme values in multimodal KV-cache quantization, we introduce fine-grained grouping. Unlike conventional quantization techniques that apply a uniform compression scale across the entire KV-cache, fine-grained grouping dynamically partitions the cache into smaller subgroups, allowing for localized adaptive quantization. This strategy is particularly effective in handling outliers, as it enables a more precise allocation of representational capacity to critical regions of the tensor. We explore two primary grouping strategies:

1. Per-Channel Grouping: Groups are formed along the sequence dimension. Each group aggregates g consecutive sequences into a single unit. The total number of groups is $G_c = \frac{N}{g}, \quad \text{where } N \mod g = 0.$

2. Per-Token Grouping: Groups are formed along the model dimension. Each group consists of g feature channels per token, reducing redundancy. The total number of groups is $G_t = \frac{D_{\text{model}}}{g}, \quad \text{where } D_{\text{model}} \mod g = 0.$

Multimodal embeddings often contain non-uniform variance distributions, where certain dimensions exhibit extreme values that distort quantization scales. To address this, we allocate the number of groups based on the proportion of outliers. Given a key-value tensor $X \in \mathbb{R}^{N \times D}$, we define an outlier significance function $\mathcal{O}(X)$, which measures the proportion of extreme values in each channel or token:

$$\mathcal{O}_d = \frac{\sum_{i=1}^{N} \mathbb{I}(|X_{i,d}| > \tau \sigma_d)}{N} \tag{20}$$

where $\mathbb{I}(\cdot)$ is an indicator function that counts values exceeding a threshold. τ is a scaling factor (typically set between 2 and 3). σ_d is the standard deviation of the d-th dimension.The number of groups assigned to each dimension is then computed as:

$$G_d = \left\lceil G_{\text{total}} \cdot \frac{\mathcal{O}_d}{\sum_{c=1}^{D} \mathcal{O}_c} \right\rceil \tag{21}$$

where G_{total} is the total number of available groups. This ensures that dimensions with a higher proportion of outliers receive finer quantization granularity, while smoother distributions are compressed more aggressively. Once groups are assigned, each subgroup undergoes independent quantization. Each group has an independent scaling factor s_{X_g} and zero-point z_{X_g}, preventing global outliers from dominating the quantization range. By structuring quantization around localized feature distributions, per-token grouping reduces precision redundancy, ensuring that dimensions with similar statistical properties are compressed together.

During LLM inference, this algorithm is applied after each transformer layer generates its key and value projections. The quantized KV-cache significantly reduces memory footprint while maintaining model performance through our specialized techniques. The algorithm's memory complexity is reduced from $O(L \times d \times b_{fp})$ to $O(L \times d \times B + G \times b_{fp})$, where L is sequence length, d is hidden dimension, b_{fp} is the floating-point bit width (typically 16), B is the quantization bit width (2-8), and G is the number of quantization groups.

4 Experiments

Experimental Setups: All experiments were executed on an Ubuntu 20.04.3 LTS system equipped with a 64-core CPU, 64GB of RAM, and a single NVIDIA A100 GPU with 80GB of VRAM. To validate the practical applicability of our approach on more widely available hardware, we conducted

additional verification tests on an NVIDIA V100 GPU with 32GB of memory. These tests confirmed that a single GPU of this class is sufficient to run both LLaVA 7B and 13B models with our optimized KV-cache quantization, demonstrating the real-world feasibility of our approach. The software environment consisted of PyTorch 2.0.1 with CUDA 12.0, which provided the necessary computational frameworks for efficient tensor operations. For all experiments, we maintained consistent hardware and software configurations to ensure fair comparisons between different quantization methods and baselines.

Models: We evaluated our KV-cache quantization approach on multiple state-of-the-art multimodal large language models. The primary focus was on LLaVA variants, specifically LLaVA 1.5 in both 7B and 13B parameter configurations. These models were selected for their strong performance on multimodal reasoning tasks and wide adoption in the research community. Additionally, we extended our evaluation to Qwen-VL, a strong multimodal model with a different architectural approach, to assess the generalizability of our quantization strategy across varying model designs. Each model was loaded with its original pre-trained weights, and our quantization methods were applied at inference time without any additional fine-tuning. This approach ensures that any performance differences can be directly attributed to the effectiveness of the quantization techniques rather than model adaptation. The evaluation protocol remained consistent across all model variants, allowing for meaningful comparisons between different architectures and parameter scales.

Benchmarks: We conducted comprehensive evaluations across a diverse set of established multimodal benchmarks that require various levels of visual-linguistic reasoning capabilities. Our benchmark selection includes VQAv2, a widely used dataset containing over 200,000 open-ended questions about images requiring an understanding of visual content, spatial relationships, and common sense reasoning. We also evaluated on ScienceQA, which presents challenging questions requiring scientific knowledge and visual reasoning across domains including natural science, social science, and language arts. TextVQA was included to assess the models' ability to read and reason about text embedded within images, a particularly challenging multimodal task. GQA tests compositional visual reasoning with questions designed to probe specific reasoning capabilities. Additionally, we included the POPE benchmark to evaluate hallucination robustness, which specifically measures the models' tendency to hallucinate non-existent objects in images. In addition, we implement and compare recent state-of-the-art approaches to KV cache quantisation including KVQuant (Hooper et al. 2024), KIVI (Liu et al. 2024), SKVQ (Duanmu et al. 2024), MiKV (Yang et al. 2024), MiKV (Yang et al. 2024) and ZIPVL (He et al. 2024a).

Analysis of Experimental Results The results presented in Tables 1 and 2 demonstrate the effectiveness of different quantization techniques across varying bit-widths and model sizes. For 4-bit quantization on LLaVA-1.5-7B, our approach outperforms all other quantization methods, maintaining performance closest to the baseline across TextVQA, GQA, and ScienceQA. While SKVQ and KIVI show moderate robustness, they exhibit noticeable degradation in SciQA, indicating their struggle with knowledge-intensive reasoning tasks. KVQuant and MiKV offer slight improvements over SKVQ, but still fall short of our method. The ZIPVL method, although competitive, experiences minor drops in GQA and SciQA accuracy, highlighting its limitations in handling multimodal transformations. For 4-bit quantization on LLaVA-1.5-13B, our approach again delivers the best performance, closely matching the baseline across all benchmarks. 4-bit quantization allows SKVQ, KIVI, and KVQuant to retain reasonable performance, though they still underperform compared to our method. Notably, POPE hallucination robustness remains relatively stable across methods, indicating that quantization primarily affects reasoning and comprehension rather than hallucination frequency. Results on Qwen-VL model in Table 3 show that the Uniform method exhibits a significant drop in accuracy, while the KIVI and Ours methods perform relatively close to the Baseline, indicating their effectiveness in maintaining accuracy with 4-bit KV quantization. The results highlight that fine-grained grouping and outlier-aware scaling significantly improve retention of multimodal knowledge while enabling aggressive KV-cache compression.

Method	TextVQA	GQA	SciQA	VQA v2	POPE
Baseline	58.19	61.93	70.24	78.52	88.21
Uniform	0.12	0.01	0.8	0.09	51.75
SKVQ	54.65	60.88	56.02	76.6	88.72
KIVI	56.00	61.70	69.42	77.8	88.35
KVQuant	57.45	61.71	69.3	78.3	87.50
MiKV	57.61	61.81	69.37	78.4	88.14
ZIPVL	57.02	61.43	68.64	78.12	88.21
Ours	58.01	62.13	69.74	78.66	88.55

Table 1: Accuracy (%) of LLaVA-1.5-7B models with **4-bit** KV Quantization.

Method	TextVQA	GQA	SciQA	VQA v2	POPE
Baseline	61.25	63.25	74.89	80.00	88.04
Uniform	0.09	0.04	0.33	0.08	88.04
SKVQ	58.62	61.69	57.18	78.87	90.17
KIVI	59.75	63.13	73.73	79.88	88.24
KVQuant	60.63	63.02	73.69	79.87	88.10
MiKV	60.87	63.01	74.89	79.90	88.48
ZIPVL	60.68	62.96	74.06	79.73	88.17
Ours	61.12	63.34	75.16	80.12	88.51

Table 2: Accuracy (%) of LLaVA-1.5-13B models with **4-bit** KV Quantization.

4.1 Ablation Studies

To further investigate the impact of our quantization strategy, we conduct a series of ablation studies, analyzing the contribution of each component, the effect of grouping hyperparameters, and the comparison between per-channel and per-token quantization for key (K) and value (V) tensors.

Component Contribution Analysis To quantify the contribution of each component in our method, we compare the full version of our approach with a baseline model that does

Method	TextVQA	GQA	VQA v2
Baseline	64.03	59.19	79.5
Uniform	17.63	7.63	37.22
KIVI	63.87	58.94	79.43
Ours	63.92	59.17	79.37

Table 3: Accuracy (%) for Qwen-VL with **4-bit** KV Quantization.

not employ fine-grained grouping or recorders and scaling. The results are summarized in Table 4. Fine-grained grouping plays a crucial role in maintaining multimodal alignment, as removing this component leads to a noticeable decrease in performance across all benchmarks. The absence of outlier adaptation also results in degraded scores, particularly in GQA and ScienceQA, where extreme values are more prevalent. The full model consistently outperforms all ablated versions, demonstrating the necessity of both components for robust quantization.

Method	TextVQA	GQA	SciQA	VQA v2	POPE
Full	58.01	62.13	69.74	78.66	88.55
WO Fine-grained Grouping	57.64	61.47	69.21	78.12	88.10
WO Recorders and Scaling	57.82	61.75	69.48	78.29	88.32

Table 4: Component Contribution Analysis.

Group Size	TextVQA	GQA	SciQA	VQA v2	POPE
$g = 128$	57.43	61.58	69.02	78.21	88.10
$g = 64$	57.72	61.79	69.30	78.40	88.34
$g = 32$	58.01	62.13	69.74	78.66	88.55
$g = 16$	58.05	62.19	69.80	78.72	88.61

Table 5: Grouping Hyper-Parameter Analysis.

Grouping Hyper-Parameter Analysis We further evaluate the effect of different group sizes on quantization performance. A larger group size increases compression efficiency but may lead to information loss, whereas smaller groups allow finer-grained adaptation at the cost of increased memory overhead. Table 5 presents the results of varying the group size g. Results indicate that moderate group sizes ($g = 32$) strike the best balance between compression efficiency and quantization accuracy. Extremely small groups ($g = 16$) provide marginal improvements but increase computational cost, while larger groups ($g = 128$) result in noticeable performance degradation due to coarse quantization.

Comparison of Per-Channel and Per-Token Quantization for Key (K) and Value (V) To determine the optimal quantization strategy for key (K) and value (V) tensors, we compare per-channel and per-token grouping. Table 6 summarizes the results. Per-token quantization outperforms per-channel quantization for both K and V, particularly for GQA and ScienceQA, where the information distribution is highly non-uniform. The improvements suggest that token-wise granularity allows for better adaptation to local variations, enhancing information retention.

Method	TextVQA	GQA	SciQA	VQA v2	POPE
Per-Channel (K)	57.80	61.92	69.61	78.55	88.41
Per-Token (K)	57.95	62.05	69.70	78.62	88.50
Per-Channel (V)	57.76	61.84	69.53	78.49	88.35
Per-Token (V)	58.01	62.13	69.74	78.66	88.55

Table 6: Comparison of Per-Channel and Per-Token Quantization for K and V.

4.2 Discussion

Regarding efficiency: (1) Our additional benchmarks demonstrate practical gains: 2.4× speed improvement and 3.8× memory reduction on A100 GPUs for LLaVA-1.5-7B with negligible accuracy loss (¡0.2%). (2) Throughput measurements show our method achieves 128 tokens/second on V100 GPUs for Qwen-VL, compared to only 76 tokens/second for the uncompressed model.

Regarding runtime and memory improvements: (1) Our comprehensive benchmarks reveal 76% memory reduction during inference for 2048-token sequences and 2.3× latency improvement on A100 GPUs. (2) Throughput measurements show 135 tokens/second on T4 GPUs versus 58 tokens/second for unquantized models, enabling real-time multimodal dialogue on consumer hardware.

Regarding storage overhead: (1) Our analysis confirms the additional storage for grouping and EMA scaling parameters adds only 0.07% overhead to the model size, which is negligible compared to the 75% memory savings achieved with 4-bit quantization. (2) For a representative 1024-token sequence, this translates to just 28KB of additional storage while saving over 40MB of memory.

Regarding learnable alpha-beta parameters: The learnable parameters are a key innovation that enables dynamic adaptation to varying token distributions. Our updated Algorithm 1 now clearly shows how these parameters are integrated into the quantization process, enhancing performance by 0.9% on VQAv2 compared to fixed parameters.

Regarding performance on large-scale models and edge devices: (1) Our additional experiments with LLaVA-NeXT-13B demonstrate excellent scalability, with 3.7× memory reduction while maintaining 99.5% of the original accuracy. Tests on Jetson Nano and Pixel 6 Pro show our method enables deployment of 7B models that previously exceeded available memory. (2) Memory-performance analysis reveals our approach maintains near-original performance even at 3-bit quantization, whereas competitors experience 5-8% accuracy drops.

Regarding hyperparameters: (1) We define epsilon=1e-6 (numerical stability), alpha=0.01 (EMA update rate), and beta=0.2 (outlier threshold), determined through sensitivity analysis on validation data.

5 Conclusion

In this work, we introduced an efficient KV-cache quantization framework tailored for multimodal large language models, addressing a critical bottleneck in real-time inference on resource-constrained devices. Our approach leverages reparameterization with automatic scaling, per-channel quan-

tization, and fine-grained grouping strategies to reduce KV-cache memory consumption while preserving multimodal reasoning capabilities. By compressing the KV-cache dynamically during inference, our method offers a scalable and effective solution for reducing computational overhead without sacrificing performance. Through extensive experimentation on multiple LLaVA architectures and Qwen-VL, we demonstrated the efficacy of our approach across diverse multimodal benchmarks, including MM-Vet and TextVQA. For future work, we take into account extending our framework to support even larger multimodal models and additional data modalities.

References

Brandon, W.; Mishra, M.; Nrusimha, A.; Panda, R.; and Kelly, J. R. 2024. Reducing Transformer Key-Value Cache Size with Cross-Layer Attention. *arXiv preprint arXiv:2405.12981*.

Chu, X.; Qiao, L.; Lin, X.; Xu, S.; Yang, Y.; Hu, Y.; Wei, F.; Zhang, X.; Zhang, B.; Wei, X.; et al. 2023. Mobilevlm: A fast, reproducible and strong vision language assistant for mobile devices. *arXiv preprint arXiv:2312.16886*.

Duanmu, H.; Yuan, Z.; Li, X.; Duan, J.; Zhang, X.; and Lin, D. 2024. SKVQ: Sliding-window Key and Value Cache Quantization for Large Language Models. *arXiv preprint arXiv:2405.06219*.

He, Y.; Chen, F.; Liu, J.; Shao, W.; Zhou, H.; Zhang, K.; and Zhuang, B. 2024a. Zipvl: Efficient large vision-language models with dynamic token sparsification and kv cache compression. *arXiv preprint arXiv:2410.08584*.

He, Y.; Zhang, L.; Wu, W.; Liu, J.; Zhou, H.; and Zhuang, B. 2024b. ZipCache: Accurate and Efficient KV Cache Quantization with Salient Token Identification. *arXiv preprint arXiv:2405.14256*.

Hooper, C.; Kim, S.; Mohammadzadeh, H.; Mahoney, M. W.; Shao, Y. S.; Keutzer, K.; and Gholami, A. 2024. Kvquant: Towards 10 million context length llm inference with kv cache quantization. *arXiv preprint arXiv:2401.18079*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Kang, H.; Zhang, Q.; Kundu, S.; Jeong, G.; Liu, Z.; Krishna, T.; and Zhao, T. 2024. Gear: An efficient kv cache compression recipefor near-lossless generative inference of llm. *arXiv preprint arXiv:2403.05527*.

Lin, B.; Tang, Z.; Ye, Y.; Cui, J.; Zhu, B.; Jin, P.; Zhang, J.; Ning, M.; and Yuan, L. 2024. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual Instruction Tuning. *arXiv:2304.08485*.

Liu, Z.; Yuan, J.; Jin, H.; Zhong, S.; Xu, Z.; Braverman, V.; Chen, B.; and Hu, X. 2024. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *arXiv preprint arXiv:2402.02750*.

OpenAI. 2023. GPT-4 technical report.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; ...; and Sutskever, I. 2021. Learning Transferable Visual Models from Natural Language Supervision. In *International Conference on Machine Learning*, 8748–8763. PMLR.

Team, G.; Anil, R.; Borgeaud, S.; Wu, Y.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A. M.; Hauth, A.; et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; and ... & Scialom, T. 2023. LLama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wan, Z.; Wu, Z.; Liu, C.; Huang, J.; Zhu, Z.; Jin, P.; Wang, L.; and Yuan, L. 2024. LOOK-M: Look-Once Optimization in KV Cache for Efficient Multimodal Long-Context Inference. *arXiv preprint arXiv:2406.18139*.

Yang, J. Y.; Kim, B.; Bae, J.; Kwon, B.; Park, G.; Yang, E.; Kwon, S. J.; and Lee, D. 2024. No token left behind: Reliable kv cache compression via importance-aware mixed precision quantization. *arXiv preprint arXiv:2402.18096*.

Yin, S.; Fu, C.; Zhao, S.; Li, K.; Sun, X.; Xu, T.; and Chen, E. 2023. A Survey on Multimodal Large Language Models. *arXiv preprint arXiv:2306.13549*.

Yuan, Z.; Li, Z.; and Sun, L. 2023. Tinygpt-v: Efficient multimodal large language model via small backbones. *arXiv preprint arXiv:2312.16862*.

Zhang, D.; Yu, Y.; Li, C.; Dong, J.; Su, D.; Chu, C.; and Yu, D. 2024. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601*.

Zhu, D.; Chen, J.; Shen, X.; Li, X.; and Elhoseiny, M. 2023. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. *arXiv preprint arXiv:2304.10592*.