

MBQ: Modality-Balanced Quantization for Large Vision-Language Models

Shiyao Li^{*1,2}, Yingchun Hu^{*2,3}, Xuefei Ning^{†1}, Xihui Liu⁵, Ke Hong^{1,2}, Xiaotao Jia^{†3,4}, Xiuhong Li²,
 Yaqi Yan⁶, Pei Ran⁶, Guohao Dai^{7,2}, Shengen Yan², Huazhong Yang¹, Yu Wang^{†1}

¹Tsinghua University ²Infinigence-AI ³Beihang University ⁴Qingdao Research Institute, Beihang University

⁵University of Hong Kong ⁶Chinatower ⁷Shanghai Jiao Tong University

Abstract

Vision-Language Models (VLMs) have enabled a variety of real-world applications. The large parameter size of VLMs brings large memory and computation overhead which poses significant challenges for deployment. Post-Training Quantization (PTQ) is an effective technique to reduce the memory and computation overhead. Existing PTQ methods mainly focus on large language models (LLMs), without considering the differences across other modalities. In this paper, we discover that there is a significant difference in sensitivity between language and vision tokens in large VLMs. Therefore, treating tokens from different modalities equally, as in existing PTQ methods, may over-emphasize the insensitive modalities, leading to significant accuracy loss. To deal with the above issue, we propose a simple yet effective method, **Modality-Balanced Quantization (MBQ)**, for large VLMs. Specifically, MBQ incorporates the different sensitivities across modalities during the calibration process to minimize the reconstruction loss for better quantization parameters. Extensive experiments show that MBQ can significantly improve task accuracy by up to 4.4% and 11.6% under W3A16 and W4A8 quantization for 7B to 70B VLMs, compared to SOTA baselines. Additionally, we implement a W3A16 GPU kernel that fuses the dequantization and GEMV operators, achieving a $1.4\times$ speedup on LLaVA-onevision-7B on the RTX 4090. The code is available at <https://github.com/thu-nics/MBQ>.

1. Introduction

Large Vision-Language Models (VLMs) have made significant progress and enabled various real-world tasks, such as image captioning [10], visual question answering (VQA) [2], and so on. However, due to the large memory and computation overhead, existing large VLMs, such as LLaVA [31], InternVL [11], and QwenVL [4], are hard

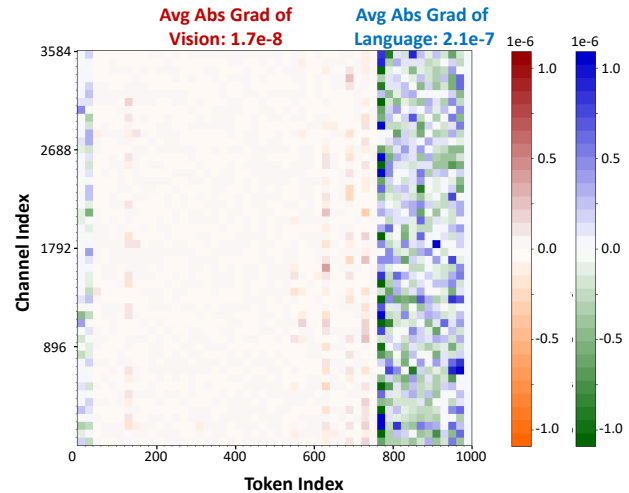


Figure 1. The gradients of loss function w.r.t. the token features of the 13th transformer block in the LLaVA-onevision-7B VLM on COCO caption dataset [8, 10]. The red and orange represent vision tokens, and the blue and green represent language tokens.

to deploy on commonly used accelerators, such as GPUs. For example, the largest LLaVA-onevision VLM, with 72B parameters, requires 144GB of memory, which exceeds the 80GB memory capacity of the A100 GPU.

Many methods have already been designed to improve the inference efficiency of LLM [54], including quantization [3, 15, 27], sparse attention [16, 44, 51], efficient decoding strategies [25, 36] and so on. Among them, Post-Training Quantization (PTQ) methods are effective in reducing the memory access and computation overhead to accelerate LLM inference. To reduce the memory access and storage overhead, weight-only quantization methods have been developed, such as AWQ [27], GPTQ [15], QuIP [7], and so on. To reduce the computation overhead, weight-activation quantization methods, such as SmoothQuant [43], SpinQuant [33], FlatQuant [41], Atom [53], and so on, have been applied, enabling the use of the faster low-precision tensor cores on GPUs. To retain the task accuracy, during the calibration process, these meth-

^{*}Equal contribution. [†]Corresponding to: foxdoraame@gmail.com, jiaxt@buaa.edu.cn, yu-wang@tsinghua.edu.cn

ods search for optimal scaling factors [15, 39], channel-wise equalization factors [27, 43], rotation matrices [33], etc., by minimizing the feature reconstruction error of each block.

While PTQ methods are well-studied for LLMs, their suitability for VLMs has not been fully explored, where tokens from multiple modalities need to be handled and cross-modality tasks need to be addressed. Our experiments reveal that directly applying SOTA PTQ methods for LLMs to VLMs results in substantial accuracy degradation. We speculate that the main reason is that existing methods treat vision and language tokens equally, overlooking their significantly different sensitivities.

To verify this, as depicted in Fig. 1, using an image-caption pair from the COCO dataset as the input [10], we visualize the loss gradient w.r.t. the output feature of the 13th layer in LLaVA-onevision-7B. We can see that the average absolute gradient value of the language token features is over $10\times$ larger than that of vision token features. This 1st-order approximation analysis suggests that with the same size of perturbation on the features, language tokens might impact the loss function more than $10\times$ as much as vision tokens. Consequently, treating vision and language tokens equally during calibration may over-emphasize the insensitive vision tokens, resulting in a notable accuracy loss.

Taking the sensitivity differences into consideration, we propose an extremely simple but effective method, **Modality-Balanced Quantization (MBQ)**, for quantizing large VLMs. Specifically, MBQ uses the gradients of the loss function [31] w.r.t. vision and language token features as the sensitivity indicators. These sensitivity indicators are then incorporated into the reconstruction loss as the objective for searching optimal channel-wise equalization factors [27, 43] in both weight-only (W3A16 and W4A16) and weight-activation quantization (W4A8 and W8A8). By balancing the effect of different modalities, MBQ can significantly increase the accuracy of the quantized VLMs.

We conduct extensive experiments across 7B-70B VLMs on challenging vision-language benchmarks. The results show MBQ significantly improves the task accuracy by up to 4% and 11% under W3A16 and W4A8 quantization compared with other SOTA methods. We also analyze the shortcomings of baseline methods and perform comprehensive ablation studies on various factors, including the choice of calibration datasets, alternative formulations of modality-balancing losses, and the quantization of visual encoders.

For W3A16 quantization, we design a GPU kernel that fuses 3-bit dequantization with general matrix-vector multiplication (GEMV). For W4A16, W4A8, and W8A8 quantization, we adopt existing open-source GPU kernels [27, 29] to accelerate the inference process. Experiments on different workloads show that we can achieve up to $1.4\times$ end-to-end speedup on LLaVA-onevision-7B on RTX 4090.

2. Preliminaries

2.1. Quantization Formats

In this paper, we focus on applying the uniform integer quantization, which is a commonly used quantization format, to the weight ($\mathbf{W}$) and input activation ($\mathbf{X}$) matrices of each linear layers in VLMs.

For weight-only quantization, existing methods typically apply asymmetric uniform quantization for weight groups (i.e., group-wise quantization), as shown below:

$$\mathbf{W}_{\text{asym}} = \left\lceil \frac{\mathbf{W}_{\text{FP16}} - Z}{S_{\text{asym}}} \right\rceil, \quad (1)$$

$$S_{\text{asym}} = \frac{\max(\mathbf{W}_{\text{FP16}}) - Z}{2^N - 1}, \quad (2)$$

where $\mathbf{W}_{\text{FP16}}$ denotes the 16-bit floating-point (FP16) value, $\mathbf{W}_{\text{asym}}$ denotes the low-precision integer value. N is bit-width. S_{asym} and $Z = \min(\mathbf{W}_{\text{FP16}})$ denote the scaling factor and zero-point of the asymmetric quantization.

For weight-activation quantization, symmetric uniform quantization is commonly used for both weight and input activation matrices of linear layers:

$$\mathbf{W}_{\text{sym}} = \left\lfloor \frac{\mathbf{W}_{\text{FP16}}}{S_{\text{sym}}} \right\rfloor, \quad (3)$$

$$S_{\text{sym}} = \frac{\text{absmax}(\mathbf{W}_{\text{FP16}})}{2^{N-1} - 1}, \quad (4)$$

where S_{sym} denotes the scaling factor of the symmetric quantization.

For simplicity, we use $\mathbf{WxAy}$ to indicate the quantization format, where the $\mathbf{x}$ and $\mathbf{y}$ represent to the bit-width for Weight and Activation. For example, W4A8 denotes quantizing weights to 4-bit and activations to 8-bit.

2.2. Channel-Wise Equalization

Existing PTQ methods automatically search for optimal hyperparameters of quantization by minimizing the reconstruction error of each transformer block during a calibration process. A series of popular methods [27, 43] aim to equalize outliers in weight and activation matrices by channel-wise equalization (CWE). Specifically, they search the CWE factors $\mathbf{E}$ by minimizing the Mean Square Error (MSE) loss in each transformer block. Taking weight-only quantization as an example, the objective of CWE is shown in the following equation:

$$\mathbf{E}^* = \underset{\mathbf{E}}{\operatorname{argmin}} ||Q(\mathbf{W} * \mathbf{E})(\mathbf{E}^{-1} * \mathbf{X}) - \mathbf{WX}||^2, \quad (5)$$

where Q means the quantization function.

3. Method

3.1. Sensitivity Varies Across Modalities

As introduced in Sec. 2.2, when using visual-language datasets for calibration, **CWE treats visual and language activations equally during the calibration process.**

Intuitively, we speculate that the significant performance degradation when applying SOTA LLM quantization methods to VLMs stems from treating different modalities equally. This is because errors in vision tokens might have a smaller impact on the output context compared to introducing the same errors in language tokens, due to the following two reasons: (1) From the data perspective, visual data generally contains a high degree of redundancy and might be more fault tolerant for small perturbations. (2) From the model perspective, Zhang et al. [50] discover that the generated content of current VLMs is primarily biased by the pre-trained LLMs rather than the input image.

To verify the above intuition, we evaluate the sensitivity of the output tokens w.r.t the input vision and language tokens on the COCO caption dataset [10]. Specifically, we take the image-caption pairs as inputs of VLMs and calculate gradients of the SFT (Supervised Fine-Tuning) loss function w.r.t. language and vision tokens. The gradients can reflect the impacts on the output language tokens (caption) when small perturbations are applied to language (caption) and vision (image) token features. Note that due to the attention mechanism, the gradient of the SFT loss can still backpropagate to vision tokens in each transformer block, even though we only account for the loss of the output language tokens.

As shown in Fig. 1, the average absolute gradient of language tokens is an order of magnitude larger than that of vision tokens. This means that, for a similar perturbation, a vision token’s impact on the SFT loss is only $0.1\times$ that of a language token. **Therefore, if we still treat language and vision tokens equally, we will miss the opportunity to leverage the VLM’s lower sensitivity to vision tokens to achieve higher performance.**

To demonstrate the impact of accounting for modality-specific sensitivities during calibration, we conduct an oracle experiment by applying a modality-balancing factor of 0.1 to the vision tokens’ reconstruction loss during CWE calibration. The optimization objective, referred to as balanced CWE, is shown below:

$$\mathbf{E}^* = \underset{\mathbf{E}}{\operatorname{argmin}} [||Q(\mathbf{W} * \mathbf{E})(\mathbf{E}^{-1} * \mathbf{X}_l) - \mathbf{W}\mathbf{X}_l||^2 \quad (6)$$

$$+ 0.1 ||Q(\mathbf{W} * \mathbf{E})(\mathbf{E}^{-1} * \mathbf{X}_v) - \mathbf{W}\mathbf{X}_v||^2], \quad (7)$$

where the $\mathbf{X}_l$ and $\mathbf{X}_v$ mean the language and vision tokens.

The results of our ablation study are shown in Tab. 1, with a heuristic selected modality-balancing factor, the balanced CWE easily surpasses the performance of CWE by

Model	FP16	W3		
		RTN	CWE	Balanced CWE
LLaVA-ov-7B	46.00	34.67	36.56	40.22
InternVL2-8B	48.00	43.67	44.78	46.33

Table 1. The accuracy results (%) on MMMU benchmark after W3 quantization. The calibration dataset is selected from COCO caption dataset [10]. The LLaVA-ov-7B is short for LLaVA-onevision-7B.

1.55% $\sim$ 3.66% under W3 quantization. The significant improvements indicate the importance of balancing the different sensitivities across different modalities.

3.2. Modality-Balanced Quantization (MBQ)

Given that the sensitivity differences between vision and language tokens may vary across layers and VLM families, exploring an automatic modality-balancing approach could further enhance the performance of the quantized model.

In this section, **we aim to derive an approach for allocating the optimal Modality-Balanced factors to each layer by directly minimizing the change in the SFT loss function.** Specifically, we employ the first-order Taylor approximation in Eq. (8) to determine how the SFT loss L changes in response to a small perturbation Δ in output activation $\mathbf{Y}$ of each linear layer:

$$L(\mathbf{Y} + \Delta) \simeq L(\mathbf{Y}) + \mathbf{g}^T * \Delta, \quad (8)$$

where $\mathbf{g}^T$ represents the gradient of the output activation $\mathbf{Y}$.

The change in SFT loss caused by quantization can be expressed as the following equation:

$$||L(\hat{\mathbf{Y}})|| \simeq ||\mathbf{g}^T * \Delta|| \quad (9)$$

$$= ||\mathbf{g}_v^T * \Delta_v + \mathbf{g}_l^T * \Delta_l|| \quad (10)$$

$$\leq ||\mathbf{g}_v^T * \Delta_v|| + ||\mathbf{g}_l^T * \Delta_l|| \quad (11)$$

$$\leq |\bar{\mathbf{g}}_v| * ||\Delta_v|| + |\bar{\mathbf{g}}_l| * ||\Delta_l|| \quad (12)$$

$$= |\bar{\mathbf{g}}_v| * ||\mathbf{Y}_v - \hat{\mathbf{Y}}_v|| + |\bar{\mathbf{g}}_l| * ||\mathbf{Y}_l - \hat{\mathbf{Y}}_l||, \quad (13)$$

where $\mathbf{Y}_v$ and $\hat{\mathbf{Y}}_v$ represent the output vision tokens of the FP16 linear layer and the quantized linear, respectively. $\mathbf{Y}_l$ and $\hat{\mathbf{Y}}_l$ represent the output language tokens of the FP16 linear layer and the quantized linear, respectively. The $|\bar{\mathbf{g}}_v|$ and $|\bar{\mathbf{g}}_l|$ represent the average absolute gradient of each linear layer’s output vision and language tokens, respectively.

In this paper, we combine MBQ with channel-wise equalization to search optimal equalization factors for better performance.

Firstly, in order to accelerate the prefill stage of VLMs, we quantize both the weights and input activations of each linear layer to leverage fast low-precision tensor cores. The

objective is shown in the following:

$$\min_{\mathbf{E}} [|\bar{\mathbf{g}}_v| * ||\mathbf{W}\mathbf{X}_v - Q(\mathbf{W} * \mathbf{E})Q(\mathbf{E}^{-1} * \mathbf{X}_v)||] \quad (14)$$

$$+ |\bar{\mathbf{g}}_l| * ||\mathbf{W}\mathbf{X}_l - Q(\mathbf{W} * \mathbf{E})Q(\mathbf{E}^{-1} * \mathbf{X}_l)||], \quad (15)$$

where $\mathbf{X}_v$ and $\mathbf{X}_l$ represent the input token and language activations of each linear layer, respectively.

Secondly, to accelerate the decode stage of VLMs, we only quantize weights to reduce the memory overhead, aiming to minimize the balanced reconstruction error by the following objective:

$$\min_{\mathbf{E}} [|\bar{\mathbf{g}}_v| * ||\mathbf{W}\mathbf{X}_v - Q(\mathbf{W} * \mathbf{E})(\mathbf{E}^{-1} * \mathbf{X}_v)||] \quad (16)$$

$$+ |\bar{\mathbf{g}}_l| * ||\mathbf{W}\mathbf{X}_l - Q(\mathbf{W} * \mathbf{E})(\mathbf{E}^{-1} * \mathbf{X}_l)||], \quad (17)$$

Notably, unlike directly using the heuristically selected MSE-based balanced CWE loss in Sec. 3.1, our derived reconstruction error loss function relies on Mean Absolute Error (MAE). Our ablation study in Sec. 4.3 demonstrates that minimizing MAE-based reconstruction loss in MBQ yields better results than using an MSE-based one.

3.3. End-to-End Acceleration Implementation

As illustrated in Sec. 6.1, we need to quantize not only the VLMs with large parameter counts but also the ViT encoders for efficient deployment, which have substantial computational overhead when transforming high-resolution images into vision tokens. The ViT encoders take a series of image patches as input and generate a set of vision tokens as output. The input activation of each linear layer in ViT encoders are 2D matrices, so the main computation operator is GEMM.

To this end, we apply weight-activation quantization to the ViT encoders and weight-only quantization for VLMs to accelerate the auto-regressive decode stage. To achieve practical hardware acceleration for VLMs, we designed a custom fused W3 quantization GPU kernel by fusing the dequantization process with the GEMV operator. Specifically, we pack eight 3-bit weights into three bytes, and the fused W3 kernel first loads the W3 weights instead of FP16 weights to reduce memory access overhead. It then dequantizes the W3 weights to FP16. Finally, the fused W3 kernel performs computations using the FP16 tensor core.

With the fused W3 kernel and open-source GPU kernel libraries [27, 29] for quantization, we can accelerate the inference speed of both the quantized ViT encoders and VLMs. Detailed experiments in Sec. 4.4 demonstrate the hardware performance of the proposed W3 kernel and end-to-end speedups across various scenarios.

4. Experiments

4.1. Experimental Setups

4.1.1. Calibration Datasets

The calibration task requires leveraging both the VLM’s ability to understand image details and its language modeling capabilities. “Image captioning” is one of the tasks that addresses both aspects. Specifically, we choose the improved COCO Caption dataset proposed by ShareGPT4V [8] and sample 128 image-caption pairs as the calibration dataset. Chen et al. [8] use GPT4-V to generate a high-quality caption for each image. For each VLM, we apply the corresponding conversation template to each image-caption pair to create an instructional format.

4.1.2. Evaluation Datasets

To evaluate the performance of the quantized model, we conduct experiments on various vision-language benchmarks based on the LMMs-Eval [49]. Specifically, we use OCRBench [32] and TextVQA [40] for text recognition and comprehension, VizWiz [18] and SEED-Bench [19] to test visual perception, and ScienceQA [34] and MMMU [47] to evaluate visual reasoning. Additionally, to demonstrate the practical conversational performance of the quantized VLM, we present several cases on the LLaVA-bench-in-the-wild [31] and LLaVA-bench-wilder [20] datasets in the supplementary Sec. 8.2.

4.1.3. Models

We benchmark various quantization methods on LLaVA-onevision [21], InternVL2 [11], and Qwen2-VL [4] families. For each model family, we select both a smaller and a larger parameter version to showcase the capability of MBQ across different model sizes. For LLaVA-onevision, we select models with 7B and 72B parameters, which utilize Qwen2-7B/-72B for the VLM and SigLIP-400M [48] for the ViT encoder. For InternVL2, we evaluate the 8B and 26B models, which incorporate InternLM2-8B/-20B as the VLM component and use InternViT-300M/-6B as the vision encoder. For Qwen2-VL, we use models with 7B and 72B parameters in our evaluation, which use Qwen2-7B/-72B for the VLM component and a 675M ViT encoder.

4.1.4. Baselines

For weight-only quantization, we compare MBQ with the vanilla round-to-nearest quantization (RTN) and AWQ [27] under W3A16, which is based on channel-wise equalization. We apply group-wise asymmetric quantization for each method and keep the group size at 128. For weight-activation quantization, we compare with both the RTN and SmoothQuant [43] under W4A8, which also relies on channel-wise equalization. As discussed in SmoothQuant, we apply per-token symmetric quantization for activations and per-channel (output) symmetric quantization for

Model	Bitwidth	Method	MMMU	SEED	OCRBench	VizWiz	ScienceQA	TextVQA	Average (↑)
LLaVA-onevision-7B	FP16	-	46.0	74.9	62.2	60.4	85.4	76.1	67.5
	W3A16	RTN	34.7	5.9	35.9	59.2	86.2	60.9	47.1
		GPTQ	41.9	72.9	55.7	56.4	86.4	71.3	64.1
		AWQ	36.6	53.0	59.3	58.5	83.2	73.0	60.6
		MBQ	42.0	69.7	61.1	60.7	85.0	73.3	65.3
	W4A8	RTN	38.2	50.3	40.1	58.2	88.3	61.5	56.1
		SQ	30.9	42.7	32.0	56.7	87.1	56.9	51.1
		MBQ	42.6	67.7	52.3	58.9	88.5	68.3	63.1
InternVL2-8B	FP16	-	48.0	76.0	76.5	61.1	96.2	77.0	72.5
	W3A16	RTN	43.7	74.9	74.0	56.0	95.6	74.6	69.8
		GPTQ	41.7	73.4	70.2	59.9	89.5	73.1	68.0
		AWQ	44.8	75.2	74.7	58.9	95.5	74.2	70.6
		MBQ	46.9	75.4	75.1	58.7	95.6	75.1	71.1
	W4A8	RTN	44.3	74.0	72.0	57.1	95.5	73.1	69.3
		SQ	41.8	73.8	70.9	54.2	95.1	72.6	68.1
		MBQ	45.6	74.3	73.0	56.5	95.8	72.3	69.6
Qwen2-VL-7B	FP16	-	50.6	76.4	80.7	68.3	85.1	82.0	73.8
	W3A16	RTN	44.9	74.8	60.0	65.2	81.5	71.2	66.3
		GPTQ	43.1	73.7	74.8	64.3	79.7	76.7	68.7
		AWQ	44.7	75.1	76.9	68.0	82.5	79.5	71.1
		MBQ	47.9	74.8	76.8	67.7	82.8	79.9	71.6
	W4A8	RTN	43.8	74.9	60.3	58.9	78.9	71.0	64.6
		SQ	45.9	75.0	57.1	52.3	80.9	68.2	63.2
		MBQ	47.2	75.1	72.8	59.3	81.2	75.0	68.4

Table 2. Main results on the small models of LLaVA-onevision, InternVL2, and Qwen2-VL families. “SQ” is short for SmoothQuant.

Model	Bitwidth	Method	MMMU	SEED	OCRBench	VizWiz	ScienceQA	TextVQA	Average (↑)
LLaVA-onevision-72B	FP16	-	56.1	78.1	73.2	69.2	90.0	79.3	74.3
	W3A16	RTN	53.9	77.4	68.2	66.1	89.5	77.4	72.1
		GPTQ	52.7	76.0	69.7	68.3	89.3	77.9	72.3
		AWQ	33.4	71.2	48.7	49.3	69.2	58.8	55.1
		MBQ	54.4	77.6	71.6	69.0	90.3	78.5	73.6
	W4A8	RTN	54.8	76.6	64.5	64.7	89.0	74.5	70.7
		SQ	51.6	76.6	64.2	65.7	89.1	74.4	70.3
		MBQ	55.6	76.5	64.4	65.7	89.2	73.3	70.8
InternVL2-26B	FP16	-	47.1	76.8	77.9	66.2	97.5	82.1	74.6
	W3A16	RTN	46.6	75.7	75.9	64.7	96.4	80.6	73.3
		GPTQ	44.8	75.8	76.0	60.9	96.3	80.1	72.3
		AWQ	46.4	76.2	76.4	64.5	96.7	81.0	73.5
		MBQ	47.1	76.3	76.5	64.5	97.3	81.1	73.8
	W4A8	RTN	44.7	76.0	76.4	62.6	96.7	79.6	72.7
		SQ	38.2	70.6	68.5	56.7	86.3	72.6	65.5
		MBQ	44.0	75.7	77.5	62.0	97.1	80.0	72.7
Qwen2-VL-72B	FP16	-	61.1	77.6	79.9	76.0	91.6	82.5	78.1
	W3A16	RTN	57.7	77.5	70.4	74.8	89.7	79.7	75.0
		GPTQ	57.3	77.2	78.5	73.6	91.5	81.6	76.6
		AWQ	59.6	77.6	79.6	75.4	90.4	82.4	77.5
		MBQ	59.6	77.7	79.4	75.6	90.5	82.5	77.6
	W4A8	RTN	58.1	76.6	66.2	71.3	90.1	77.0	73.2
		SQ	55.9	76.4	65.5	69.7	88.8	76.9	72.2
		MBQ	57.7	76.3	77.5	73.6	89.6	80.5	75.8

Table 3. Main results on the large models of LLaVA-onevision, InternVL2, and Qwen2-VL families. “SQ” is short for SmoothQuant.

weights to take advantage of low-precision tensor cores. The evaluation results of W4 and W8A8 are shown in Supplementary Sec. 8.

4.2. Main Results

For Weight-only Quantization, as shown in Tab. 2 and Tab. 3, the results of RTN quantization indicate that smaller VLMs are more sensitive to weight-only quan-

BitWidth	Method	Components				MMMU ($\uparrow$)	SEED ($\uparrow$)
		Channel-wise Equalization	COCO Calib.	Modality-Balance (MSE)	Modality-Balance (MAE)		
FP16	-	-	-	-	-	46.0	71.1
W3A16	RTN	$\times$	$\times$	$\times$	$\times$	34.7	10.4
	AWQ	$\checkmark$	$\times$	$\times$	$\times$	36.6	51.5
	-	$\checkmark$	$\checkmark$	$\times$	$\times$	38.7	61.8
	-	$\checkmark$	$\checkmark$	$\checkmark$	$\times$	40.8	64.8
	MBQ	$\checkmark$	$\checkmark$	$\times$	$\checkmark$	42.0	66.4
	MBQ	$\checkmark$	$\checkmark$	$\times$	$\checkmark$	42.0	66.4
W4A8	RTN	$\times$	$\times$	$\times$	$\times$	38.2	48.4
	SQ	$\checkmark$	$\times$	$\times$	$\times$	30.9	41.6
	-	$\checkmark$	$\checkmark$	$\times$	$\times$	29.2	10.2
	-	$\checkmark$	$\checkmark$	$\checkmark$	$\times$	41.9	63.5
	MBQ	$\checkmark$	$\checkmark$	$\times$	$\checkmark$	42.6	64.4
	MBQ	$\checkmark$	$\checkmark$	$\times$	$\checkmark$	42.6	64.4

Table 4. The ablation study on LLaVA-onevision-7B with W3A16 and W4A8 quantization. “SQ” is short for SmoothQuant.

tization. The average accuracy loss for RTN-quantized 7B and 8B VLMs is 9.6%, significantly higher than the 1.5% loss seen in VLMs over 26B. This trend also aligns with observations in LLMs [24, 46].

The proposed MBQ can significantly outperform the AWQ baseline across different families. Especially within the LLaVA-onevision family, MBQ achieves an average accuracy improvement of 4.2% on the 7B VLM and 18.5% on the 72B VLM compared to AWQ. It is worth noting that for the LLaVA-onevision-72B VLM, AWQ even shows a 17% performance drop compared to RTN. Instead, the proposed MBQ can significantly improve the average accuracy and surpass RTN quantization. For the InternVL2 and Qwen2-VL families, MBQ can also outperform the RTN and AWQ baselines by around 1%.

For Weight-Activation Quantization, similar to weight-only quantization, MBQ can significantly outperform SmoothQuant and RTN baselines, with improvements of up to 11.6%. In many cases, the average performance of SmoothQuant is lower than that of RTN quantization, especially in InternVL2-26B under W4A8. The results indicate that with activation quantization, our insight into modality-balancing becomes more crucial for improving the performance of quantized VLMs, as the core idea of modality-balancing mainly addresses the sensitivity differences among various modalities within the activations.

4.3. Ablation Study

4.3.1. The Effect of Calibration Datasets

As AWQ and SmoothQuant are designed for LLM quantization, they use the Pile [17] validation dataset during calibration, which only contains language data.

Directly apply the vision-language dataset as calibration does not consistently improve the performance of the quantized VLMs. As shown in Tab. 4, when we use the

COCO caption dataset as the calibration dataset, the performance of AWQ W3A16 can significantly increase by 2.1% and 10.3% on MMMU and SEED datasets, respectively. However, the performance of SmoothQuant W4A8 with the COCO caption calibration dataset significantly **decreases** by 1.7% and 31.4% on MMMU and SEED datasets.

We speculate that this is because weight-activation quantization requires considering both weight and activation quantization errors. Ignoring the sensitivity differences between vision and language tokens in activations leads to a significant performance drop in SmoothQuant, even falling behind RTN quantization.

4.3.2. The Effect of Modality-Balance

Modality-Balancing plays a crucial role in weight-activation quantization and can also improve the performance of weight-only quantization. As shown in Tab. 4, with the help of Modality-Balancing (MAE), the SmoothQuant with COCO calibration can significantly improve 13.4% and 54.2% on MMMU and SEED, respectively. For the weight-only quantization, Modality-Balancing (MAE) can increase the accuracy by 3.3% and 4.6% on MMMU and SEED, respectively. This significant performance improvement confirms the importance of accounting for the varying sensitivities of different modalities during the calibration process.

We find that Modality-Balancing (MAE) can consistently outperform Modality-Balancing (MSE) in both weight-only and weight-activation quantization. For both W3A16 and W4A8, Modality-Balancing with MAE reconstruction loss can achieve over 1% accuracy improvement on both MMMU and SEED datasets. Therefore, we recommend using the MAE reconstruction loss form, derived directly from the gradient of the activation with respect to the SFT loss, rather than the MSE reconstruction loss.

Model	Bitwidth	Method	MMMU	OCRBench	VizWiz
LLaVA-onevision-72B	FP16	-	56.1	73.2	69.2
		QuaRot	53.0	70.8	67.9
	W4A8	SpinQuant	52.6	70.9	61.0
		MBQ (Rot)	55.8	72.5	69.0
InternVL2-26B	FP16	-	47.1	77.9	66.2
		QuaRot	44.2	76.7	62.1
	W4A8	SpinQuant	43.7	76.8	64.5
		MBQ (Rot)	47.4	77.8	65.6
Qwen-VL-72B	FP16	-	61.1	79.9	76.0
		QuaRot	49.7	78.1	71.4
	W4A8	SpinQuant	50.8	77.4	69.3
		MBQ (Rot)	60.0	79.9	75.4

Table 5. Rotation-based quantization results on the large models of LLaVA-onevision, InternVL2, and Qwen2-VL families.

In addition, we also study two different reweight strategies to show the effectiveness of MBQ: (1) We randomly divided calibration tokens into two equal groups to reweight quant error. For W3A16 LLaVA-onevision-7B, the accuracy on OCRBench is 1.3% lower than MBQ, which is similar to the case without reweighting. (2) We directly use the gradient of each token to perform token-wise quantization error reweighting. For W3A16 LLaVA-onevision-7B, the accuracy on OCRBench is 1.5% lower than MBQ. We find that this is because **the gradient of many vision tokens is zero**, leading to an inability to account for the quantization error of these tokens. We will highlight these new results.

4.3.3. Combine MBQ with Rotation-based Quantization

From Tab. 3, we observe that for large VLMs, the performance improvement of MBQ over RTN weight-activation quantization is not substantial, and there remains a significant performance gap compared to FP16 VLMs.

As demonstrated in Sec. 3, the proposed Modality-Balancing approach can be seamlessly integrated with any block-wise LLM quantization method to support VLMs. To this end, we combine Modality-Balancing with the SOTA rotation-based quantization technique [41], resulting in MBQ (Rot), which can achieve significant better performance. As illustrated in Tab. 5, across three different large VLMs, the proposed MBQ (Rot) under W4A8 quantization exhibits an accuracy loss of less than 1.1% compared to the original FP16 VLMs. Furthermore, when compared to other state-of-the-art methods, W4A8 MBQ (Rot) achieves performance improvements of up to 9.2%.

4.3.4. Quantize Both Visual Encoder and VLM

In Sec. 3.3, we analyze the different efficiency bottlenecks of the ViT encoder and VLM. In real-world applications, we need to quantize both components for higher acceleration ratios. As shown in Tab. 6, we quantize the ViT encoder with SmoothQuant and VLM with MBQ in LLaVA-onevision-7B and evaluate the accuracy on MMMU and

ViT	VLM	MMMU ($\uparrow$)	VizWiz ($\uparrow$)
FP16	FP16	46.0	60.4
FP16	W3A16	42.0	60.7
FP16	W4A8	42.6	58.9
W4A8	W3A16	42.6	62.8
W4A8	W4A8	43.1	60.1

Table 6. The results of the quantized ViT encoder and VLM in LLaVA-onevision-7B on the MMMU and VizWiz. We use MBQ for VLM quantization and SmoothQuant for ViT quantization.

BitWidth	Method	MMLU ($\uparrow$)
FP16	-	65.9
W3A16	AWQ (Pile)	61.9
	AWQ (COCO)	62.0
	MBQ	62.9
W4A8	SQ (Pile)	59.8
	SQ (COCO)	59.0
	MBQ	61.8

Table 7. The results of quantized LLaVA-onevision-7B on the MMLU benchmark. “SQ” is short for SmoothQuant.

VizWiz benchmarks.

Experimental results show that applying W4A8 quantization to the ViT encoder does not lead to a significant performance drop; instead, it even improves performance on some benchmarks. This suggests that the ViT encoder is not particularly sensitive to quantization, possibly due to the redundancy in vision tokens discussed in Sec. 1. Therefore, quantizing the ViT encoder alongside the VLM quantization is feasible from an algorithmic performance perspective.

4.3.5. The Performance on Language-only Benchmark

The main idea of the proposed MBQ is to consider the sensitivity across different modalities during quantization, aiming to enhance performance in both vision-language and language-only tasks. Accordingly, we evaluated the performance of the quantized LLaVA-onevision-7B VLM on the MMLU benchmark with different quantization methods.

As shown in Tab. 7, with W3A16 and W4A8 quantization, MBQ achieves a performance improvement of 0.9% and 2%, compared to AWQ and SmoothQuant. This result demonstrates that for quantized VLMs, considering the sensitivity differences across modalities not only significantly reduces the performance loss on vision-language tasks, but also helps maintain performance on language-only tasks.

4.4. Efficiency Evaluation

4.4.1. Single Kernel Performance

We evaluate the speed of the proposed fused W3A16 GEMV kernel on the RTX 4090 GPU, testing linear layers

Shape (IC × OC)	FP16 (ms)	W3A16 (ms)	Speedup
3584 × 3584	3.0	2.6	1.2×
3584 × 10752	8.3	2.7	3.1×
3584 × 18944	14.4	2.9	5.0×
18944 × 3584	14.6	3.1	4.7×

Table 8. The speed up of the fused W3A16 General Matrix-Vector Multiplication (GEMV) kernel on RTX 4090 GPU. “IC” and “OC” are short of “Input Channel” and “Output Channel”.

with different weight matrix shapes in LLaVA-onevision-7B. Specifically, as shown in Tab. 8, we evaluate the kernel on 4 different shapes of weight matrices, including 3584 × 3584 (the out_proj layers), 3584 × 10752 (the qkv_proj layers), 3584 × 18944 (the up_proj and gate_proj layers), and 18944 × 3584 (the down_proj layers). For each shape, we compare the proposed fused W3A16 GEMV kernel with the original FP16 GEMV kernel.

The evaluation results show that the fused W3A16 GEMV kernel achieves a speedup of 1.2× to 5.0× across 4 different shapes, compared to the FP16 GEMV baseline. As the matrix size increases, the fused W3A16 kernel achieves a more significant speedup. This is because GEMV performance is memory-bound, and the impact of memory access speed becomes more pronounced with larger weight matrices. By applying W3A16 quantization, memory access latency is significantly reduced in large matrix GEMV operations, leading to a greater speedup.

4.4.2. End-to-End Performance

With the help of the proposed W3A16 GPU kernel and the W4A8 GPU kernel from Qserve [29], we evaluate the latency of both the ViT encoder and the VLM in LLaVA-onevision-7B on RTX 4090. We run both the ViT encoder and VLM with FlashAttention-2 [13]. The evaluation results are shown in Tab. 9.

For the ViT encoder, the embedding layer will convert each input image into 729 (27 × 27) tokens as the input. As discussed in Sec. 3.3, the ViT encoder only has the prefill stage, the main operators are GEMMs, which is compute-bound. In this case, we apply the W4A8 kernel to the ViT encoder and achieve 1.15× speedup.

For the VLM, as discussed in Sec. 6.1, the VLM has two distinct stages, and each stage requires different quantization schemes for inference acceleration.

Specifically, in order to accelerate the prefill stage, we need to apply weight-activation quantization to use the low-precision tensor cores. With W4A8 quantization, we evaluate the inference latency of the quantized VLM with the open-source GPU kernel. As shown in Tab. 9, when the input token length is 512 and 1024 tokens, W4A8 can achieve 1.16× and 1.18×, respectively. As the token length increases, the speedup gradually becomes greater.

In order to accelerate the decode stage, we apply W3A16

Model	Stage	FP16 (ms)	W3A16 (ms)	W4A8 (ms)
ViT	Prefill (729 tokens)	11.2	-	9.7
	Prefill (512 tokens)	68.8	-	59.4
VLM	Prefill (1024 tokens)	109.0	-	92.7
	Decode	29.6	21.1	26.3

Table 9. The end-to-end speed up of LLaVA-onevision-7B on RTX4090 with fused GPU kernels.

quantization to the VLM. Since the latency of the decode stage remains similar across different input token lengths, we measure the average latency of the decode stage at input token lengths of 128, 256, 512, and 1024. As shown in Tab. 9, both W4A8 and W3A16 quantization can accelerate the decode stage. However, the W3A16-quantized VLM is 1.23× faster than the W4A8 model. This result primarily stems from two factors: (1) the W3A16-quantized model has lower memory access overhead for large weight matrices compared to W4A8, and (2) the W4A8 operator requires additional dynamic quantization of activations, which incurs additional computational time.

For practical acceleration, we use weight-activation quantization to accelerate the ViT encoder and weight-only quantization to speed up the decode stage of VLMs. The main reason is that the FP16 VLM’s prefill stage takes as long to process 1,024 tokens as the decode stage takes to generate just 4 tokens, as shown in Tab. 9. In real-world applications, the number of tokens to be decoded is large, necessitating multiple forward passes during the decode stage, which is the most time-consuming part of generation tasks. Thus, weight-only quantization is more suitable for VLMs.

5. Conclusion

In this paper, we identify a key phenomenon: the sensitivities of vision and language tokens to quantization differ significantly. Based on this finding, we propose Modality-Balanced Quantization (MBQ), a simple but effective quantization method to improve the accuracy of quantized VLMs for both weight-only and weight-activation quantization. Specifically, we use the gradients of the SFT loss function w.r.t. vision and language tokens as sensitivity indicators to balance the reconstruction loss during calibration. MBQ can outperform existing SOTA PTQ methods by 4.4% and 11.6% on both W4A8 and W3A16 across various VLMs. With the proposed W3A16 CUDA kernel, we achieve 1.4× decoding speedup compared with FP16 baseline.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (No. 62325405, 62104128, U19B2019, U21B2031, 61832007, 62204164, 92364201), Tsinghua EE Xilinx AI Research Fund, and Beijing National Research Center for Information Science and Technology (BNRist). We thank for all the support from Infinigence-AI.

References

- [1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 2
- [2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 1
- [3] Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefer, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. *arXiv preprint arXiv:2404.00456*, 2024. 1
- [4] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 1, 4
- [5] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024. 2
- [6] Jianjian Cao, Peng Ye, Shengze Li, Chong Yu, Yansong Tang, Jiwen Lu, and Tao Chen. Madtp: Multimodal alignment-guided dynamic token pruning for accelerating vision-language transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15710–15719, 2024. 2
- [7] Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M De Sa. Quip: 2-bit quantization of large language models with guarantees. *Advances in Neural Information Processing Systems*, 36, 2024. 1
- [8] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 2023. 1, 4
- [9] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2025. 2
- [10] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 1, 2, 3
- [11] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024. 1, 4, 2
- [12] Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, et al. Mobilevlm: A fast, reproducible and strong vision language assistant for mobile devices. *arXiv preprint arXiv:2312.16886*, 2023. 2
- [13] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023. 8
- [14] Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefer, and Dan Alistarh. Spqr: A sparse-quantized representation for near-lossless llm weight compression. *arXiv preprint arXiv:2306.03078*, 2023. 1
- [15] Elias Frantar, Saleh Ashkboos, Torsten Hoefer, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022. 1, 2
- [16] Tianyu Fu, Haofeng Huang, Xuefei Ning, Genghan Zhang, Boju Chen, Tianqi Wu, Hongyi Wang, Zixiao Huang, Shiyao Li, Shengen Yan, et al. Moa: Mixture of sparse attention for automatic large language model compression. *arXiv preprint arXiv:2406.14909*, 2024. 1
- [17] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020. 6
- [18] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018. 4
- [19] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023. 4
- [20] Bo Li, Kaichen Zhang, Hao Zhang, Dong Guo, Renrui Zhang, Feng Li, Yuanhan Zhang, Ziwei Liu, and Chunyuan Li. Llava-next: Stronger llms supercharge multimodal capabilities in the wild, 2024. 4, 2
- [21] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 4, 2
- [22] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 2
- [23] Shiyao Li, Xuefei Ning, Ke Hong, Tengxuan Liu, Luning Wang, Xiuhong Li, Kai Zhong, Guohao Dai, Huazhong Yang, and Yu Wang. Llm-mq: Mixed-precision quantization for efficient llm deployment. In *The Efficient Natural Language and Speech Processing Workshop with NeurIPS*, 2023. 1
- [24] Shiyao Li, Xuefei Ning, Luning Wang, Tengxuan Liu, Xianguangsheng Shi, Shengen Yan, Guohao Dai, Huazhong Yang,

- and Yu Wang. Evaluating quantized large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 28480–28524, 2024. 6
- [25] Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*, 2024. 1
- [26] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Jinfa Huang, Junwu Zhang, Yatian Pang, Munan Ning, et al. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*, 2024. 2
- [27] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100, 2024. 1, 2, 4
- [28] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26689–26699, 2024. 2
- [29] Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. Qserve: W4a8kv4 quantization and system co-design for efficient llm serving. *arXiv preprint arXiv:2405.04532*, 2024. 2, 4, 8
- [30] Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. *arXiv preprint arXiv:2405.05803*, 2024. 2
- [31] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2023. 1, 2, 4
- [32] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xucheng Yin, Chenglin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: On the hidden mystery of ocr in large multimodal models, 2024. 4
- [33] Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant-llm quantization with learned rotations. *arXiv preprint arXiv:2405.16406*, 2024. 1, 2
- [34] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 4
- [35] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruvi Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. Mml: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024. 2
- [36] Xuefei Ning, Zinan Lin, Zixuan Zhou, Zifu Wang, Huazhong Yang, and Yu Wang. Skeleton-of-thought: Prompting llms for efficient parallel generation. In *The Twelfth International Conference on Learning Representations*, 2024. 1
- [37] Yanyuan Qiao, Zheng Yu, Longteng Guo, Sihan Chen, Zijia Zhao, Mingzhen Sun, Qi Wu, and Jing Liu. Vl-mamba: Exploring state space models for multimodal learning. *arXiv preprint arXiv:2403.13600*, 2024. 2
- [38] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llva-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*, 2024. 2
- [39] Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. *arXiv preprint arXiv:2308.13137*, 2023. 2, 1
- [40] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8317–8326, 2019. 4
- [41] Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, et al. Flatquant: Flatness matters for llm quantization. *arXiv preprint arXiv:2410.09426*, 2024. 1, 7
- [42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1
- [43] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pages 38087–38099. PMLR, 2023. 1, 2, 4
- [44] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023. 1
- [45] Ruyi Xu, Yuan Yao, Zonghao Guo, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, Maosong Sun, and Gao Huang. Llva-uhd: an llm perceiving any aspect ratio and high-resolution images. *arXiv preprint arXiv:2403.11703*, 2024. 2
- [46] Zhewei Yao, Xiaoxia Wu, Cheng Li, Stephen Youn, and Yuxiong He. Zeroquant-v2: Exploring post-training quantization in llms from comprehensive study to low rank compensation. *arXiv preprint arXiv:2303.08302*, 2023. 6
- [47] Xiang Yue et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*, 2024. 4
- [48] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023. 4
- [49] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Reality check on the evaluation of large multimodal models, 2024. 4
- [50] Yi-Fan Zhang, Weichen Yu, Qingsong Wen, Xue Wang, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. Debi-asing multimodal large language models. *CoRR*, 2024. 3

- [51] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36: 34661–34710, 2023. [1](#)
- [52] Han Zhao, Min Zhang, Wei Zhao, Pengxiang Ding, Siteng Huang, and Donglin Wang. Cobra: Extending mamba to multi-modal large language model for efficient inference. *arXiv preprint arXiv:2403.14520*, 2024. [2](#)
- [53] Yilong Zhao, Chien-Yu Lin, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind Krishnamurthy, Tianqi Chen, and Baris Kasikci. Atom: Low-bit quantization for efficient and accurate llm serving. *Proceedings of Machine Learning and Systems*, 6:196–209, 2024. [1](#)
- [54] Zixuan Zhou, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Luning Wang, Zhihang Yuan, Xiuhong Li, et al. A survey on efficient inference for large language models. *arXiv preprint arXiv:2404.14294*, 2024. [1](#)
- [55] Yichen Zhu, Minjie Zhu, Ning Liu, Zhicai Ou, Xiaofeng Mou, and Jian Tang. Llava-phi: Efficient multi-modal assistant with small language model, 2024. [2](#)