

CASP: Compression of Large Multimodal Models Based on Attention Sparsity

Mohsen Gholami, Mohammad Akbari, Kevin Cannons, and Yong Zhang
Huawei Technologies Canada Co., Ltd.

{mohsen.gholami1, mohammad.akbari, kevin.cannons, yong.zhang3}@huawei.com

Abstract

In this work, we propose an extreme compression technique for Large Multimodal Models (LMMs). While previous studies have explored quantization as an efficient post-training compression method for Large Language Models (LLMs), low-bit compression for multimodal models remains under-explored. The redundant nature of inputs in multimodal models results in a highly sparse attention matrix. We theoretically and experimentally demonstrate that the attention matrix’s sparsity bounds the compression error of the Query and Key weight matrices. Based on this, we introduce CASP, a model compression technique for LMMs. Our approach performs a data-aware low-rank decomposition on the Query and Key weight matrix, followed by quantization across all layers based on an optimal bit allocation process. CASP is compatible with any quantization technique and enhances state-of-the-art 2-bit quantization methods (AQLM and QuIP#) by an average of 21% on image- and video-language benchmarks. The code is available [here](#)¹.

1. Introduction

Large multimodal Models (LMMs) have garnered significant attention in recent years due to their impressive performance in image- and video-language comprehension. Despite their substantial applications, LMMs are computationally expensive, which limits their broader use. For instance, 70B models like LLaVa-Onevision [20] require 140GB of GPU memory to operate at 16-bit precision. Moreover, the inference of LMMs requires significant electricity consumption, which raises concerns about environmentally friendly AI [40]. To address this problem, compression techniques such as knowledge distillation [12, 14], quantization [7, 8, 22], and low-rank factorization [41, 47] have been proposed.

Recent studies predominantly focus on compressing large language models (LLMs), while the LMMs compression

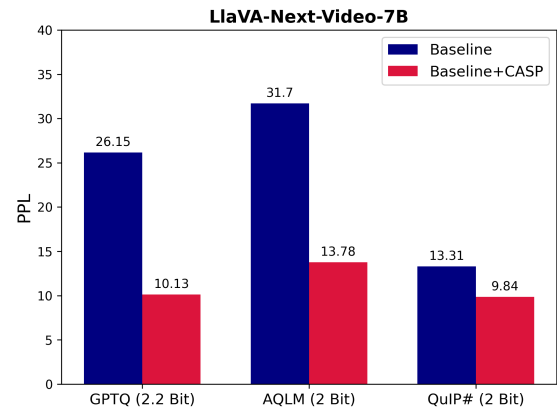


Figure 1. CASP considers the specific properties of LMMs and offers significant improvement over state-of-the-art model quantization methods. PPL: perplexity.

remains less explored. Although LLM compression techniques can be applied to LMMs, there are fundamental differences between the two. Popular LMMs convert visual modalities into hundreds to thousands of tokens, which are then concatenated with text tokens and fed to an underlying LLM. Unlike textual data, multimodal inputs are typically highly redundant. This redundancy is reflected in the attention layers, which can effectively be considered during LMMs compression.

Fig. 2 illustrates the attention maps of LLaVa-Next-Video-7B compared to Llama2-7B for the same layer (i.e., layer 15). Despite LLaVa-Next-Video-7B utilizing Llama2-7B as its underlying LLM, the attention of the two models differs significantly. This observation raises two under-explored questions: 1) *How do the activations and attentions of LMMs differ from those of LLMs?* 2) *What insights should be considered to optimize LMMs compression?*

In LMMs, visual tokens typically receive much less attention compared to text tokens. For example, in LLaVa-1.5 [24], vision tokens approximately receive ten times less attention compared with text tokens. This disparity is even more pronounced in LLaVa-Next-Video with greater number of visual tokens. The cumulative attention on text tokens leads to a sparse attention map (Fig. 2). We hypothesize

¹<https://github.com/vbdi/casp>

that the sparse attention maps can be recovered by smaller attention weights, allowing for low-bit compression. This phenomenon is well-documented in the compressed sensing literature [1, 5], which shows that through optimization, a signal’s sparsity can be leveraged to reconstruct it from far fewer samples than those required by the Nyquist–Shannon sampling theorem. While our problem differs from those addressed in compressed sensing, it is closely related.

In addition, although existing post-training model compression methods nearly match the original model’s performance with up to 3-bit precision, lower bits (e.g., 2 bits) result in a significant accuracy drop. This effect remains an open problem for both LLMs and LMMs. AQLM [7] and QuIP# [38] are state-of-the-art techniques for 2-bit LLM quantization, but they both rely on time-consuming fine-tuning to restore the performance of compressed models. For instance, quantizing a 70B model with AQLM can take 10-14 days. In this paper, we tackle this issue with a fine-tuning-free, post-training model compression approach for LMMs in the low-bit precision scenario. To the best of our knowledge, this is the first study to explore 2-bit LMM compression.

Specifically, we propose CASP, a model Compression method for LMMs based on Attention SParsity. Motivated by our empirical observations, we theoretically show that the compression error of the attention weights (W_q and W_k in Eq. 1) is bounded by the sparsity of the attention maps. This implies that with sparser maps, greater compression of the attention weights can be achieved with negligible performance drop. Consequently, we introduce a joint quantization and low-rank decomposition technique that compresses the attention weights to 6% of their original size, equivalent to 1 bit. Initially, we perform low-rank decomposition on the attention weight matrices. Following this, we propose an optimal bit allocation strategy for each layer in the model to assign more bits to critical layers so that an average target bit rate can be achieved. Our method is orthogonal to any quantization technique. CASP significantly enhances the performance of state-of-the-art 2-bit quantization methods, including AQLM and QuIP#, by an average of 35% and 7% on different image- and video-language benchmarks. Additionally, we demonstrate that CASP can be applied to LLMs, improving AQLM and QuIP# by an average of 11% and 2.7% on language-only benchmarks.

Contributions. Our major contributions are as follows:

1. Providing both theoretical and experimental insights showing the compression error of the attention weight matrices is bounded by the attention maps sparsity.
2. Proposing CASP, a novel low-bit LMM compression method based on attention sparsity.
3. CASP is a finetuning-free approach, that is compatible with any quantization method and also applicable to both LMMs and LLMs.

4. Validating CASP’s performance across a wide range of image-language, video-language, and language-only benchmarks using 5 different LMMs and LLMs, achieving state-of-the-art results.

2. Related Works

LLM compression methods fall into two main categories: training-aware and post-training. Training-aware methods, like quantization aware training (QAT) [32, 43], integrate quantization with network training, making models more suitable for quantization. However, due to the high time and computational costs of training modern LLMs, these methods are less favored. Post-training compression methods are more practical as they compress pre-trained models in one-shot without additional training [6, 8, 21, 27, 38, 41].

In post-training LLM compression, the main methods are quantization [2, 6, 8, 38], pruning [8, 27], knowledge distillation [13, 16], and low-rank decomposition [17, 21, 41, 47]. Post-training quantization (PTQ) is particularly notable for its efficiency, as it quantizes network parameters after training, requiring less computation than quantization aware training (QAT) while still achieving competitive performance [6, 38].

Early works on PTQ existed before the rise of LLMs [11, 31]. Initial PTQ methods for LLMs used round-to-nearest projections, adjustable for different memory/accuracy needs [4, 34, 45]. GPTQ [8] introduced a data-aware approach, minimizing l_2 errors on a calibration dataset for each network layer using a large-scale solver. Recent methods like QuIP [2] and QuIP# [38] use a two-step PTQ process involving weight smoothing and subsequently mapping weights onto a lattice codebook. AQLM [6], similar to QuIP, uses a data-driven codebook but with additive weight encoding and omitting the smoothing step. PTQ has not yet been applied to LMMs, which will be addressed by our proposed solution.

Another post-training model compression method is low-rank decomposition, such as singular value decomposition (SVD), which approximates matrices with lower-rank matrices to compress the model [17, 21, 41, 47]. Despite its potential, SVD for LLM compression is relatively unexplored. ASVD [47] was one of the first approaches but suffered from performance degradation at high compression ratios. SVD-LLM [41] improved model accuracy at high compression by using a data whitening strategy to account for the impact of each singular value on the compression loss. MoDeGPT [21] uses a different strategy by grouping matrices into larger modules and applying 3 types of module-dependent matrix approximations. Despite these initial efforts, it appears that such methods have not been considered for LMMs.

The literature has mainly focused on applying single post-training compression methods to pre-trained LLMs.

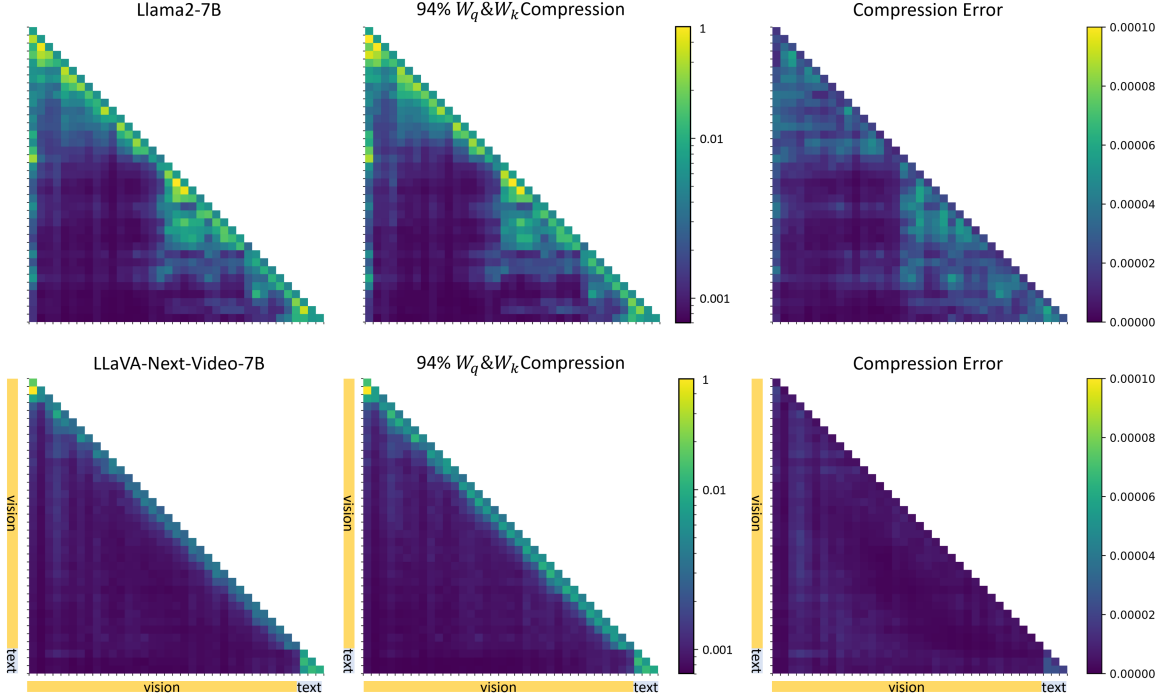


Figure 2. Left column: Comparison of LLaVA-Next-Video-7B and Llama2-7B attention maps (S in Eq. 1) at Layer 15. Despite LLaVA-Next-Video using Llama2 as its base LLM, there is a notable difference in their maps, with LLaVA showing high sparsity. Middle column: Attention maps when W_q and W_k (i.e., attention weights) are 94% compressed (equivalent to 1 bit). Right column: Compression errors (E in Eq. 3). The sparsity in LLaVA’s map results in smaller errors when compressing W_q and W_k .

However, pruning, quantization, and low-rank decomposition can be combined. LQ-LoRA [15] is a recent work that combines these strategies by performing matrix decomposition to create a high precision low-rank component and a memory-efficient quantized component. No prior work has applied PTQ, low-rank decomposition, or their combination to LMMs, which is the focus of our proposed approach.

3. Method

In this section, we first describe the attention mechanism, its sparsity, and low-rank features, which motivate us to perform low-rank decomposition on the attention weights. Following that, we prove the theorem that the compression error is bounded by the attention map sparsity. Next, we explain the low-bit quantization process as the second phase of our method, followed by a proof showing that our bit allocation is optimal.

3.1. Attention Weights Compression

The scaled-dot-product (SDP) attention in transformer-based models [39] operates on queries $\mathbf{Q} = XW_q \in \mathbb{R}^{N \times d}$ and keys $\mathbf{K} = XW_k \in \mathbb{R}^{N \times d}$, where X represents the input activations from the previous layer, W_q and W_k are the model weights, d is the hidden dimension, and N is the number of tokens. The attention maps are then defined as:

$$\mathbf{S} = \text{Softmax} \left(\frac{XW_qW_k^\top X^\top}{\sqrt{d}} \right) = \text{Softmax}(Y), \quad (1)$$

where $\mathbf{S} \in \mathbb{R}^{N \times N}$. This operation has three specific criteria that are crucial in the context of LMM compression. *First*, unlike other weight matrices in transformers (such as those in MLPs, attention-value, and attention-output layers), W_q and W_k exhibit a highly low-rank structure [47]. *Second*, while activation outputs in transformers are generally not sparse, the attention map \mathbf{S} , representing the activation output of the layer, is sparse. *Third*, the sparsity of \mathbf{S} is even more pronounced in LMMs because vision tokens receive significantly less attention.

Given the above motivations, we perform a data-aware low-rank decomposition on W_q and W_k to obtain two low-rank weight matrices $W'_q = A_qB_q$ and $W'_k = A_kB_k$. First, following [41], we compute the covariance matrix \mathbf{C} of the calibration data and perform Cholesky decomposition to obtain the lower triangular matrix \mathbf{L} . The whitening matrix \mathcal{A} is then derived as \mathbf{L}^{-1} . By applying \mathcal{A} to the calibration data, we transform the data into a whitened space. Subsequently, we perform low-rank decomposition on the whitened data to obtain W' . The approximated attention map is then calculated as:

$$\mathbf{S}' = \text{Softmax} \left(\frac{XW'_qW'^\top_k X^\top}{\sqrt{d}} \right) = \text{Softmax}(Y'). \quad (2)$$

The error of the low-rank approximation is calculated using the Frobenius norm of the difference between the original and approximated attention maps as follows:

$$E = \|\mathbf{S}' - \mathbf{S}\| = \|\text{Softmax}(Y') - \text{Softmax}(Y)\|. \quad (3)$$

Let \mathcal{S} denote the attention map sparsity, calculated as the proportion of elements in \mathbf{S} with values above a small threshold η , divided by the total number of elements in \mathbf{S} . In the following theorem, we demonstrate that the attention compression error is bounded by the sparsity of the attention map. Consequently, the compression error decreases as the attention map becomes sparser.

3.1.1 Theorem and Proof

Theorem 1. *The compression error of the W_q and W_k (Eq. 3) is bounded by the sparsity of the attention map \mathbf{S} (Eq. 1).*

Proof. The compression error can be written as:

$$E = \|\text{Softmax}(Y + \delta Y) - \text{Softmax}(Y)\|, \quad (4)$$

where $Y, \delta Y \in \mathbb{R}^{N \times N}$ and $\|\cdot\|$ denotes the Frobenius norm. Since Softmax is applied to each row of Y independently, we can write the above equation as:

$$E = \sum_{i=1}^N \|\text{Softmax}(Y_i + \delta Y_i) - \text{Softmax}(Y_i)\|, \quad (5)$$

where $Y_i \in \mathbb{R}^{N \times 1}$ and $\delta Y_i \in \mathbb{R}^{N \times 1}$ are the i th row of Y_i and δY_i , respectively. The difference between the Softmax of $(Y_i + \delta Y_i)$ and (Y_i) can be approximated using a first-order Taylor expansion around (Y) :

$$\text{Softmax}(Y_i + \delta Y_i) \approx \text{Softmax}(Y_i) + \nabla \text{Softmax}(Y_i) \cdot \delta Y_i, \quad (6)$$

where $\nabla \text{Softmax}(Y_i) \in \mathbb{R}^{N \times N}$ is the Jacobian matrix of the Softmax function. Substituting this approximation into Eq. 5, we get:

$$E = \sum \|\nabla \text{Softmax}(Y_i) \cdot \delta Y_i\| + \epsilon_i. \quad (7)$$

where ϵ_i represents the approximation error introduced by the first-order Taylor expansion. This simplification shows that E is approximately equal to the Frobenius norm of the product between the Jacobian of the Softmax function and the perturbation δY_i . Given the above equation, we can define a lower bound for E :

$$E \leq \sum \|\nabla \text{Softmax}(Y_i)\| \cdot \|\delta Y_i\| + \epsilon_i. \quad (8)$$

The Jacobian matrix $J = \nabla \text{Softmax}(Y_i)$ of the Softmax function is an $N \times N$ matrix where each element J_{jm} is the partial derivative of the j th output with respect to the m th input. The elements of the Jacobian are:

$$J_{jm} = \frac{\partial \text{Softmax}(Y_i)_j}{\partial z_m}, \quad (9)$$

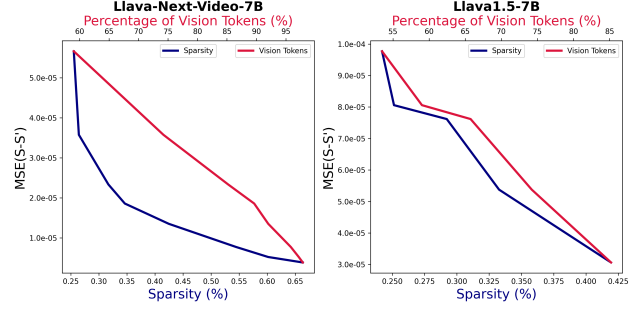


Figure 3. Compression error E (Eq. 3) for LLaVa-Next-Video and LLaVa1.5 decreases when the percentage of visual tokens increases (i.e., more sparse attention map).

which can be expressed as:

$$J_{jm} = \text{Softmax}(Y_i)_j (\delta_{jk} - \text{Softmax}(Y_i)_j), \quad (10)$$

where δ_{jm} is the Kronecker delta, which is 1 if $j = m$ and 0 otherwise. In matrix form, the Jacobian J can be written as: $J = \text{diag}(\mathbf{z}) - \mathbf{z}\mathbf{z}^T$, where $\mathbf{z} = \text{Softmax}(Y_i)$ and $\text{diag}(\mathbf{z})$ is a diagonal matrix with the elements of \mathbf{z} on the diagonal. Define the density \mathcal{D} of Y as the proportion of non-zero elements to the total number of elements, in which case the norm of the Jacobian J can be approximated as: $\|J\| \approx (1 - \frac{1}{N\mathcal{D}})^2$. Therefore, based on Eq. 8, we have:

$$\begin{aligned} E &\leq (1 - \frac{1}{N\mathcal{D}})^2 \sum \|\delta Y_i\| + \epsilon_i \\ \Rightarrow E &\leq (1 - \frac{1}{N\mathcal{D}})^2 \|\delta Y\| + \epsilon. \end{aligned} \quad (11)$$

Therefore, the upper bound of compression error E decreases as the density \mathcal{D} decreases, which corresponds to an increase in sparsity \mathcal{S} . Note that the proof is intended to be general and applies to any approximated attention map. Moreover, in the context of low-rank compression, as the sparsity of attention map increases (i.e., smaller \mathcal{D}), we can have lower ranks and truncate more eigenvalues from W_q and W_k , while maintaining the same upper bound for E in Eq. 11

Fig. 3 shows that sparser attention maps (i.e., higher ratio of vision tokens) result in lower error E in both LLaVA-Next-Video and LLaVA-1.5. This suggests that the empirical evidence is aligned with the theoretical insight in Eq. 11.

3.2. Quantization with Optimal Bit Allocation

PTQ methods minimize the layer-wise reconstruction $\|XW - X\hat{W}\|$, where \hat{W} is the quantized weight matrix and X is the activation. State-of-the-art low-bit PTQ methods [7, 38] use vector quantization and quantize a group of g weights as a g dimensional vector. In n -bit vector quantization, a vector is quantized to one of 2^{ng} vectors in \mathbb{R}^g , forming a $2^{ng} \times g$ codebook C [7, 38].

As we compress the model by low-rank factorization of W_q and W_k matrices, we can quantize important layers to higher bits and obtain the same average quantization bits as uniform quantization. We adopt the Block Influence score [29] to measure the importance of each layer as:

$$s_l = 1 - \mathbb{E} \frac{X_{in}^\top X_{out}}{\|X_{in}\| \|X_{out}\|}, \quad (12)$$

where X_{in}/X_{out} are input/output activations of layer l . We then try to maximize the sum of importance scores, weighted by the number of parameters remaining in each layer. To mitigate the adverse effects of excessive sparsity, we utilize entropic regularization to achieve smoothing. The formulation of this constrained optimization problem is as follows:

$$\max_{b_1:b_L} \sum_{l=1}^L s_l b_l p_l + \mu \sum_{l=1}^L H(b_l), \quad \text{s.t.} \quad \frac{1}{P} \sum_{l=1}^L b_l p_l = B_{avg}, \quad (13)$$

where $H(b_l) = -b_l \log(b_l)$ is the entropy, μ is the regularization parameter, and L is the number of layers of the model. b_l and p_l are the quantization bits and the number of parameters after low-rank decomposition of l th layer of the model. Also, P is the total number of parameter of the original model and B_{avg} is the target required quantization bit. The optimal layer bit distribution can be computed as:

$$b_l = \frac{1}{p_l} P B_{avg} \times \text{Softmax}(s_l p_l / \mu), \quad (14)$$

3.2.1 Theorem and Proof

Theorem 2. *Given the objective function in Eq. 13, the bit allocation in Eq. 14 is optimal.*

Proof. The Lagrangian for the constrained optimization problem in Eq. 13 can be written as:

$$\mathcal{L}(b, \lambda) = \sum_{l=1}^L (s_l b_l p_l + \mu H(b_l)) + \lambda \left(B_{avg} - \sum_{l=1}^L \frac{b_l p_l}{P} \right), \quad (15)$$

where λ is the Lagrange multiplier. To find the optimal b_l , we take the partial derivative of the Lagrangian with respect to b_l and set it to zero:

$$\frac{\partial \mathcal{L}}{\partial b_l} = s_l p_l - \mu \log(b_l) - \mu - \lambda \frac{p_l}{P} = 0. \quad (16)$$

Solving for b_l :

$$b_l = \exp \left(\frac{s_l p_l - \mu - \lambda \frac{p_l}{P}}{\mu} \right). \quad (17)$$

After substituting b_l into the constraint in Eq. 13:

$$B_{avg} = \sum_{l=1}^L \exp \left(\frac{s_l p_l - \mu - \lambda \frac{p_l}{P}}{\mu} \right) \frac{p_l}{P}. \quad (18)$$

Therefore, having Eq. 17 and 18, we can calculate b_l by:

$$b_l = \frac{B_{avg} P}{p_l} \text{Softmax} \left(\frac{s_l p_l - \mu - \lambda \frac{p_l}{P}}{\mu} \right). \quad (19)$$

Since we used the same rank for the low-rank decomposition of W_q and W_k in different layers, p_l is constant. As the Softmax function is invariant to adding constant values, we can simplify the above equation to $b_l = \frac{P B_{avg}}{p_l} \times \text{Softmax}(s_l p_l / \mu)$ that is given in Eq. 14.

4. Experiments

In this section, we analyze the performance of the proposed method compared to baselines across different image-language, video-language, and language-only models and benchmarks. The ablation studies consider the effects of the calibration dataset type, higher bit precision quantization, optimal bit allocation, and also the impact of the ratio of text vs. vision tokens.

4.1. Experimental Setting

Models. For the experiments on image-language benchmarks, we use three different LMMs namely LLaVA1.5-7B² [24], LLaVA1.5-13B³, and LLaVA-Next-7B⁴ (i.e., LLaVA1.6-7B) [23]. We also use LLaVA-Next-Video-7B⁵ [50] for the video-language experiments. All the above-mentioned models use Llama2-7B [37] or 13B as their underlying LLM (depending on the model size) and CLIP [35] as their vision encoder. LLaVA1.5 and LLaVA-Next encode the input image to 576 and a dynamic number of visual tokens, respectively, while LLaVA-Next-Video uses 144 visual tokens for each frame in the video.

Metrics and Benchmarks. Our main evaluation metric is perplexity (PPL), which is the metric commonly used for quantization methods in the literature [8, 22, 38]. PPL is defined as the exponentiation of the average negative log-likelihood of a sequence of tokens [18]. We further evaluate CASP on different downstream tasks that use their specific metrics. Except for the PPL results, all other results on downstream datasets are obtained using the LMMs-eval framework⁶ [49]. In order to measure PPL, we use LiveBench (LiveB) [42] and LLaVA-Bench-Wilder (LWilder) [18] as open-ended QA datasets and LLaVA-Bench-COCO (LCOCO) [18] as an image-captioning dataset. For downstream task performance analysis, multi-choice QA benchmarks such as SEED-Bench [19], MMU [48], ScienceQA (SQA) [26], MME [9], and MMBench [25] benchmarks are used. For video-language tasks, VideoDetailCaption [18] and VideoChatGPT (temporal) [28] are used as representative open-ended video

²<https://huggingface.co/llava-hf/llava-1.5-7b-hf>

³<https://huggingface.co/llava-hf/llava-1.5-13b-hf>

⁴<https://huggingface.co/llava-hf/llava-v1.6-vicuna-7b-hf>

⁵<https://huggingface.co/llava-hf/LLaVA-NeXT-Video-7B-hf>

⁶<https://github.com/EvolvingLMMs-Lab/lmms-eval/tree/v0.2.0>

	Method	Bit	LiveB (PPL↓)	LWilder (PPL↓)	LCOCO (PPL↓)	SEEDB (Acc↑)	SQA (EM↑)	MMM (Acc↑)	MMB (Acc↑)	MME (Cognition↑)	Avg. Rel. Improv.
LLaVA1.5-7B	Original	16	5.5	4.2	4.5	65.4	67.9	35.6	62.9	323	
	GPTQ	2.2	38.1	138.7	7.5	3.0	9.17	23.7	4.6	175	
	CASP _{GPTQ}	2.2	8.2	10.5	5.6	24.2	43.9	25.7	22.5	231	+125.6%
	AQLM	2	14.9	23.3	10.8	41.7	36.7	24.5	25.6	241	
	CASP _{AQLM}	2	7.9	8.2	5.7	52.2	50.8	27.4	38.9	298	+38.7%
	QuIP#	2	9.2	6.7	5.3	61.4	60.3	29.5	52.3	243	
CASP _{QuIP#}	2	7.1	6.6	5.3	63.1	61.6	32.1	51.8	263	+5.7%	
LLaVA1.5-13B	Original	16	5.1	4.0	4.2	67.7	71.6	36.5	68.1	312	
	GPTQ	2.2	13.7	36.2	5.5	52.9	24.6	26.5	32.5	247	
	CASP _{GPTQ}	2.2	8.6	16.0	6.0	59.7	59.7	29.7	51.5	304	+40.0%
	AQLM	2	10.2	18.1	7.9	56.4	57.7	28.2	38.3	207	
	CASP _{AQLM}	2	6.2	7.3	5.6	64.4	67.9	33.1	58.7	261	+32.0%
	QuIP#	2	6.0	6.1	4.7	66.7	68.3	33.8	63.4	270	
CASP _{QuIP#}	2	6.0	5.4	4.7	66.7	71.2	33.8	62.6	293	+2.7%	
LLaVA-Next-7B	Original	16	6.0	3.8	4.9	69.9	70.1	36.1	66.9	312	
	GPTQ	2.2	29.9	211.0	88.7	5.9	9.42	25.1	9.36	151	
	CASP _{GPTQ}	2.2	10.6	10.5	6.3	37.4	37.5	27.1	10.0	199	+141.3%
	AQLM	2	19.3	26.6	9.5	30.3	32.7	24.4	12.2	162	
	CASP _{AQLM}	2	9.3	7.7	6.4	60.7	53.5	29.5	43.3	219	+78.6%
	QuIP#	2	7.7	5.6	5.3	65.6	60.6	31.2	56.7	263	
CASP _{QuIP#}	2	7.3	4.8	5.1	66.4	61.1	32.4	55.4	241	+2.3%	

Table 1. Comparison results of our proposed CASP and different baselines (including GPTQ, AQLM, and QuIP#) on LLaVA models across image-language understanding datasets. All the results are in 2-bit precision, except GPTQ (i.e., 2.2-bit). The average relative improvement of CASP over the baselines is also provided.

QA datasets, which are evaluated using PPL, ROUGE-L (RG-L), and OpenAI’s GPT score with GPT-4o-mini (i.e., out of 5) [33]. Moreover, we perform experiments over three multiple-choice QA benchmarks, NextQA [44], VideoMME (VMME) [10], and ActivityNetQA [46], which are evaluated in terms of Exact Match (EM) or accuracy.

Baselines. We use 3 recent quantization methods, GPTQ [8], AQLM [7], and QuIP# [38], as our baselines. GPTQ is designed for the quantization of 3-bit or higher. For 2-bit, GPTQ employs a group size of 128, which is equivalent to 2.2-bit. AQLM offers three different schemes of “Number of Codebooks” \times “Codebook Size” for low-bit compression: 1×16 , 8×8 , and 1×8 . Among these, only 1×8 is equivalent to 2-bit quantization, which we utilize in this paper. QuIP# uses the E8P12 codebook for 2-bit quantization. Although both QuIP# and AQLM use time-consuming fine-tuning, for a fair comparison with our method, we report the results without fine-tuning. In all the experiments in this paper including the baselines and our method (for both low-rank factorization and quantization), we use a calibration dataset consisting of 1024 samples from RedPajama [3], each with a sequence length of 4096.

Configuration. W_q and W_k are low-rank compressed to 25% of their original size for GPTQ and AQLM (i.e., 75% of eigenvalues are removed). For QuIP#, this number is 50%. 3-bit quantization is then performed for more important layers chosen by the optimal bit allocation procedure in Section 3.2. For the rest of the layers, 2-bit is used.

4.2. Image-Language Understanding

Tab. 1 presents the numerical results of LLaVA1.5-7B, LLaVA1.5-13B, and LLaVA-Next-7B LMMs compressed using different baselines (i.e., GPTQ, AQLM, and QuIP#) with and without CASP on the image-language benchmarks. The relative improvement (averaged over all the benchmarks) of CASP compared to each quantization technique is highlighted in blue in the last column. Note that except for GPTQ and CASP_{GPTQ} (i.e., 2.2 bit), all other models use average 2-bit quantization.

As shown in Tab. 1, compared to CASP_{QuIP#}, the relative improvements achieved by CASP_{GPTQ} and CASP_{AQLM} are more significant due to the large performance gap between these quantization techniques and the original model. For example, on LLaVA1.5-7B, CASP_{GPTQ} and CASP_{AQLM} show relative improvements of 125% and 38%, respectively. Although QuIP# has a smaller margin compared to the original model, CASP_{QuIP#} still achieves a relative improvement of 5.7%. It is important to note that the theoretical upper bound for relative improvement with QuIP# is 22.6%, assuming the original model represents the upper limit. Therefore, the 5.7% improvement achieved by CASP_{QuIP#} is still quite significant.

For LLaVA1.5-13B, the performance drop of all the baselines compared to the original model is smaller, as it has more redundant parameters compared to LLaVA1.5-7B. Still, CASP_{GPTQ} and CASP_{AQLM} respectively show a large relative performance improvement of 40% and 32%

Method	Bit	VideoChatGPT			VideoDetailCaption			ActivityNet		NextQA	VMME	Avg. Rel. Improv.
		(RG-L \uparrow)	(PPL \downarrow)	(Score \uparrow)	(RG-L \uparrow)	(PPL \downarrow)	(Score \uparrow)	(Score \uparrow Acc \uparrow)	(EM \uparrow)	(Acc \uparrow)		
LLaVA-Next-Video-7B	Original	16	0.280	7.1	1.76	0.239	6.9	2.60	2.58 46.36	56.61	35.63	
	GPTQ	2.2	0.196	26.2	0.40	0.206	19.1	0.52	1.00 15.02	21.50	12.74	
	CASP _{GPTQ}	2.2	0.258	10.0	0.68	0.251	9.5	1.15	2.20 38.30	25.48	19.41	+69.8%
	AQLM	2	0.198	31.7	0.48	0.245	24.3	0.42	1.13 22.54	19.83	11.70	
	CASP _{AQLM}	2	0.241	13.8	0.82	0.238	11.8	0.92	1.73 31.30	22.97	20.26	+159.0%
	QuIP#	2	0.245	13.3	1.23	0.179	11.2	1.82	2.23 40.39	23.15	19.4	
	CASP _{QuIP#}	2	0.286	9.8	1.67	0.254	9.2	2.47	2.40 43.12	24.70	24.2	+21.9%

Table 2. Comparison results of our proposed CASP and different baselines (including GPTQ, AQLM, and QuIP#) on LLaVA-Next-Video-7B model across video-language understanding datasets. All the results are in 2-bit precision, except GPTQ (i.e., 2.2-bit). The average relative improvement of CASP over the baselines is also provided.

Method	Bit	C4 (PPL \downarrow)	WikiText2 (PPL \downarrow)	Avg. Rel. Improv.	
Llama2-7B	Original	16	5.12	6.63	
	GPTQ	2.2	35.01	43.64	
	CASP _{GPTQ}	2.2	26.04	21.88	+37.7%
	AQLM	2	12.57	9.74	
	CASP _{AQLM}	2	11.32	8.52	+11.2%
	QuIP#	2	11.00	8.22	
CASP _{QuIP#}	2	10.54	8.10	+2.7%	

Table 3. Comparison results of our proposed CASP and different baselines on Llama2-7B model across C4 and WikiText2 datasets. All the results are in 2-bit precision, except GPTQ (i.e., 2.2-bit).

over GPTQ and AQLM. CASP_{QuIP#} obtains an improvement of 2.7% while the theoretical upper bound for relative improvement with QuIP# is only 12.0% here.

LLaVA-Next-7B generally outperforms LLaVA1.5-7b and LLaVA1.5-13b by increasing the input image resolution, which provides approximately 3-4X more visual tokens compared to the LLaVA1.5 models [23]. As also discussed in the proposed method section, this potentially suggests higher sparsity and lower compression error. Thus, compared to the other models, a higher relative improvement of 141% and 78% is achieved by CASP_{GPTQ} and CASP_{AQLM} on LLaVA-Next-7B. Moreover, CASP_{QuIP#} obtains an average relative improvement of 2.3%, while the theoretical upper bound with QuIP# is only 16.9%.

4.3. Video-Language Understanding

The comparison results of the baselines and our method on LLaVA-Next-Video-7B and video-language benchmarks are presented in Tab. 2. As video-language tasks are generally more challenging, a larger performance drop after introducing low-bit compression is expected. On the other hand, since a higher ratio of visual tokens is generated for videos, the sparsity of the attention scores increases, which results in lower attention compression error (see Section 3.1). This feature can be seen in the results in Tab. 2, especially for CASP_{AQLM} and CASP_{QuIP#} with a substantial relative improvement of 159% and 21%.

Q, K Compression	Bit Alloc	LiveB (PPL \downarrow)	LWilder (PPL \downarrow)	LCOCO (PPL \downarrow)	Avg. (PPL \downarrow)
		38.08	138.68	7.53	61.43
	✓	27.54	112.91	7.41	49.28
✓		9.10	10.26	5.63	8.33
✓	✓	8.18	10.46	5.57	8.07

Table 4. Effect of different components on CASP_{GPTQ} performance on LLaVA1.5-7B.

4.4. Language-Only Tasks

To further validate the generality of the proposed CASP, we also perform experiments on Llama2-7B⁷ and two language datasets including C4 [36] and WikiText2 [30]. Although the motivation of sparse attention scores behind our method is more present in LMMs, the phenomenon is also observed for LLMs, which is illustrated in Fig. 2. As summarized by the results in Tab. 3, the superiority of CASP with a relative improvement of 37%, 11%, and 2.7% over GPTQ, AQLM, and QuIP#, respectively, can be seen.

4.5. Ablations

Components of CASP. Tab. 4 summarizes the effect of the two main components of CASP including Q, K low-rank factorization and optimal bit allocation. As shown in the table, the compressed LLaVA1.5-7B has a very poor performance when none of the above-mentioned components is applied (i.e., equal to the GPTQ baseline with 2.2 bit). Bit allocation in isolation is somewhat effective. However, the main improvements come from compressing attention weights. Incorporating the low-rank compression of Q, K followed the quantization of random layers to higher bits (3rd row, Tab. 4) significantly improves the performance. Moreover, applying the optimal bit allocation along with Q, K compression (4th row, Tab. 4) provides even more improvement. We argue that the proposed bit allocation has limitations to address in future work including determining the optimal bits for each weight instead of each layer.

Optimal Bit Allocation. Fig. 4 demonstrates the returned optimal bit by Eq. 14 for different values for the regular-

⁷<https://huggingface.co/meta-llama/Llama-2-7b-hf>

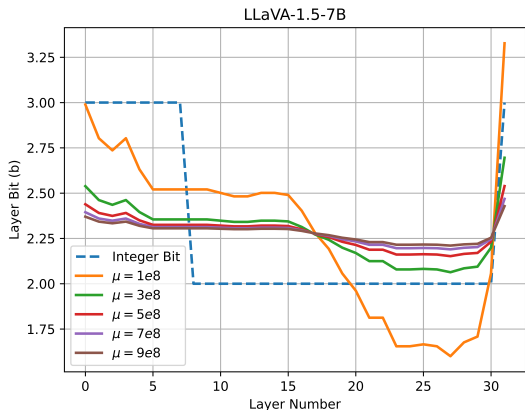


Figure 4. Optimal bit computation by Eq. 14 for different μ values. Note that the layer bit (i.e. b_l in Eq. 14) only accounts for the compression obtained from quantization, not the low-rank decomposition. Therefore, the average layer bit from the above plot is not the actual average bit of the model (i.e. B_{avg} in Eq. 14).

Calibration Dataset	LiveB (PPL↓)	LWilder (PPL↓)	LCOCO (PPL↓)	Avg. (PPL↓)
RedPajama	7.86	8.24	5.70	7.26
C4	6.82	8.96	5.70	7.16
LLaVA-Instruct-150K	7.84	13.68	5.72	9.08

Table 5. $\text{CASP}_{\text{AQLM}}$ performance with different calibration datasets on LLaVA1.5-7B.

ization parameter μ . This study was done on LLaVA1.5-7B model. The non-integer optimal bit can be allocated by changing the group size for GPTQ and by changing the codebook size and number of codebooks for AQLM and QuIP#. For simplicity, we convert the optimal bit to integer values (dashed lines in Fig. 4) and use the integer bit in our experiments. Generally, we observe that the first layers and the last layer are allocated higher bits that show the importance of these layers in the model.

Calibration Dataset. In this study, we analyze the impact of using multi-modal vs. text-only calibration datasets on the performance of the $\text{CASP}_{\text{AQLM}}$ with LLaVA1.5-7B. Specifically, we use LLaVA-Instruct-150K as a multi-modal dataset [24], and RedPajama [3] and C4 [36] as text-only datasets (1024 samples from each). The corresponding results in Tab. 5 indicate that optimizing the quantization procedure with text-only calibration datasets provides lower PPL on average. We argue that since the underlying LLM has initially been pre-trained with a significant amount of language-only data, it is more aligned with similar types of calibration data for quantization.

Higher Quantization Bits. The comparison results of CASP and the baseline quantization techniques with 3-bit precision are summarized in Tab. 6. It is shown that the performance drop of the baselines compared to the original model is marginal when it comes to 3-bit. Similarly, the

Method	Bit	LiveB (PPL↓)	LWilder (PPL↓)	LCOCO (PPL↓)	Avg. Rel. Improv.
Original	16	5.45	4.24	4.51	
GPTQ	3	9.07	6.28	5.03	
$\text{CASP}_{\text{GPTQ}}$	3	6.07	4.85	4.80	+20.1%
AQLM	3	7.64	4.54	4.72	
$\text{CASP}_{\text{AQLM}}$	3	6.06	4.88	4.86	+3.4%
QuIP#	3	5.68	4.53	4.62	
$\text{CASP}_{\text{QuIP\#}}$	3	5.69	4.51	4.63	+0.01%

Table 6. Experiments with 3-bit compression.

Attention Compression Ratio	Llama2 (C4) #VT=0	LLaVA1.5 (LWilder) #VT=575	LLaVA1.6 (LWilder) #VT≈2500	LLaVA-Next-Video (VideoChatGPT) #VT=4600
Original	7.28	4.24	3.77	7.09
50%	7.57 (↓ 4%)	4.38 (↓ 3%)	3.88 (↓ 2%)	7.14 (↓ 0.7%)
75%	8.20 (↓ 12%)	4.72 (↓ 11%)	4.13 (↓ 9%)	7.40 (↓ 4%)

Table 7. Experiments on the effect of number of visual tokens in different models LLM and LMMs in terms of PPL.

relative improvements of CASP is smaller since the theoretical upper bound for relative improvements is only 7% for QuIP#. Compared to QuIP#, CASP has no improvement as both have almost reached the original model performance.

Effect of Vision Tokens Ratio. Tab. 7 shows the experimental results for different models with various number of vision tokens (#VT). In this section, we only apply the first phase of CASP, that is low-rank decomposition of W_q and W_k and removing 50% and 75% of the eigenvalues. The results in the table show that as the number of vision tokens (sparsity of the attention map) increases, less performance drop after compression is observed. For example, Llama2 has the highest drop with 0 visual tokens, while LLaVA-Next-Video with 4.6K visual tokens has the lowest performance drop in both 50% and 75% compression ratios.

More Analysis and Results. The computational complexity analysis, the effect of calibration dataset size, further numerical experiments on more datasets, and qualitative results are given in the supplementary materials.

5. Conclusion

In this work, we proposed a low-bit model compression technique for LMMs. Our insights for CASP arise from empirical observations that the attention maps in LMMs are highly sparse. We theoretically and experimentally showed that the Query and Key weight matrices can be compressed with negligible performance drop in the model. We also proposed an optimal bit allocation approach to obtain an average target bit outperformed state-of-the-art low-bit model compression techniques. Our extensive experimental results on different LMMs and benchmarks for image- and video-language understanding showed the effectiveness and generality of the proposed method. The insight from this work has a broader impact on the architecture design of LMMs resulting in more efficient attention mechanism.

References

- [1] Emmanuel J Candes and Yaniv Plan. A probabilistic and riplless theory of compressed sensing. *IEEE transactions on information theory*, 57(11):7235–7254, 2011. 2
- [2] J. Chee, Y. Cai, V. Kuleshov, and C. De Sa. QuIP: 2-bit quantization of large language models with guarantees. 2023. 2
- [3] Together Computer. Redpajama: An open source recipe to reproduce llama training dataset, 2023. 6, 8
- [4] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. 2022. 2
- [5] David L Donoho. For most large underdetermined systems of linear equations the minimal ℓ_1 -norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(6):797–829, 2006. 2
- [6] V. Egiazarian, A. Panferov, D. Kuznedev, E. Frantar, A. Babenko, and D. Alistarh. Extreme compression of large language models via additive quantization. 2024. 2
- [7] Vage Egiazarian, Andrei Panferov, Denis Kuznedev, Elias Frantar, Artem Babenko, and Dan Alistarh. Extreme compression of large language models via additive quantization, 2024. 1, 2, 4, 6
- [8] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *ICLR*, 2023. 1, 2, 5, 6
- [9] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2024. 5
- [10] Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024. 6
- [11] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer. A survey of quantization methods for efficient neural network inference, 2021. 2
- [12] Mohsen Gholami, Mohammad Akbari, Tianxi Hu, Vaden Masrani, Z. Wang, and Yong Zhang. GOLD: Generalized knowledge distillation via out-of-distribution-guided language data generation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4365–4380, Mexico City, Mexico, 2024. Association for Computational Linguistics. 1
- [13] Y. Gu, L. Dong, F. Wei, and M. Huang. MiniLLM: Knowledge distillation of large language models. In *ICLR*, 2024. 2
- [14] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models, 2024. 1
- [15] H. Guo, P. Greengard, E. P. Xing, and Y. Kim. LQ-LoRA: Low-rank plus quantized matrix decomposition for efficient language model finetuning. In *ICLR*, 2024. 3
- [16] C. Y. Hsieh, C. L. Li, C. K. Yeh, H. Nakhost, Y. Fujii, A. Ratner, R. Krishna an C. Y. Lee, and T. Pfister. Distilling step-by-step! Outperforming larger language models with less training data and smaller model sizes. 2023. 2
- [17] Y. C. Hsu, T. Hua, S. Chang, Q. Lou, Y. Shen, and H. Jin. Language model compression with weighted low-rank factorization. In *ICLR*, 2022. 2
- [18] Frederick Jelinek, Robert L. Mercer, Lalit R. Bahl, and Janet M. Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *Journal of the Acoustical Society of America*, 62, 1977. 5
- [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. 5
- [20] 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. 1
- [21] C. H. Lin, S. Gao, J. S. Smith, A. Patel, S. Tuli, Y. Shen, H. Jin, and Y. C. Hsu. MoDeGPT: Modular decomposition for large language model compression, 2024. 2
- [22] 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 llm compression and acceleration. In *MLSys*, 2024. 1, 5
- [23] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 5, 7
- [24] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 5, 8
- [25] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player? *arXiv:2307.06281*, 2023. 5
- [26] 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. 5
- [27] X. Ma, G. Fang, and X. Wang. LLM-Pruner: On the structural pruning of large language models. 2023. 2
- [28] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, 2024. 5
- [29] Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. Shortgpt: Layers in large language models are more redundant than you expect, 2024. 5
- [30] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2022. 7
- [31] M. Nagel, R. A. Amjad, M. v. Baalen, C. Louizos, and T. Blankevoort. Up or down? Adaptive rounding for post-training quantization. 2020. 2

- [32] M. Nagel, M. Fournarakis, Y. Bondarenko, and T. Blankevoort. Overcoming oscillations in quantization-aware training. 2022. 2
- [33] OpenAI. GPT-4o mini: advancing cost-efficient intelligence. <https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence>. Accessed: 2024-11-14. 6
- [34] G. Park, B. Park, M. Kim, S. Lee, J. Kim, B. Kwon, S. J. Kwon, B. Kim, Y. Lee, and D. Lee. nuQmm: Quantized matmul for efficient inference of large-scale generative language models, 2022. 2
- [35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 5
- [36] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, 2019. 7, 8
- [37] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 5
- [38] A. Tseng, J. Chee, Q. Sun, V. Kuleshov, and C. De Sa. QuIP#: Even better LLM quantization with hadamard incoherence and lattice codebooks. 2024. 2, 4, 5, 6
- [39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 3
- [40] Maolin Wang, Yao Zhao, Jiajia Liu, Jingdong Chen, Chenyi Zhuang, Jinjie Gu, Ruo Cheng Guo, and Xiangyu Zhao. Large multimodal model compression via iterative efficient pruning and distillation. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 235–244, 2024. 1
- [41] X. Wang, Y. Zheng, Z. Wan, and M. Zhang. SVD-LLM: Truncation-aware singular value decomposition for large language model compression, 2024. 1, 2, 3
- [42] Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Siddhartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. Livebench: A challenging, contamination-free llm benchmark. 2024. 5
- [43] H. Xi, C. Li, J. Chen, and J. Zhu. Training transformers with 4-bit integers. 2023. 2
- [44] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: next phase of question-answering to explaining temporal actions, 2021. 6
- [45] Z. Yao, R. Y. Aminabadi, M. Zhang, X. Wu, C. Li, and Y. He. ZeroQuant: Efficient and affordable post-training quantization for large-scale transformers. 2022. 2
- [46] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering, 2019. 6
- [47] Z. Yuan, Y. Shang, Y. Song, Q. Wu, and G. Sun Y. Yan. ASVD: Activation-aware singular value decomposition for compressing large language models, 2023. 1, 2, 3
- [48] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhua Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*, 2024. 5
- [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. 5
- [50] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, 2024. 5