Low-Rank Approximation for Sparse Attention in Multi-Modal LLMs

Lin Song[1] Yukang Chen[3] Shuai Yang[2] Xiaohan Ding[1]
Yixiao Ge[1] Ying-Cong Chen[2] Ying Shan[1]

[1]Tencent AILab [2]HKUST(GZ) [3]CUHK

Abstract

This paper focuses on the high computational complexity in Large Language Models (LLMs), a significant challenge in both natural language processing (NLP) and multi-modal tasks. We propose Low-Rank Approximation for Sparse Attention (LoRA-Sparse), an innovative approach that strategically reduces this complexity. LoRA-Sparse introduces low-rank linear projection layers for sparse attention approximation. It utilizes an order-mimic training methodology, which is crucial for efficiently approximating the self-attention mechanism in LLMs. We empirically show that sparse attention not only reduces computational demands, but also enhances model performance in both NLP and multi-modal tasks. This surprisingly shows that redundant attention in LLMs might be non-beneficial. We extensively validate LoRA-Sparse through rigorous empirical studies in both (NLP) and multi-modal tasks, demonstrating its effectiveness and general applicability. Based on LLaMA and LLaVA models, our methods can reduce more than half of the self-attention computation with even better performance than full-attention baselines.

1. Introduction

Large Language Models (LLMs) such as LLaMA [42], T5 [38], PaLM [10], and OPT [50] have gained increasing interest in the research community, showcasing notable abilities in complex reasoning and multi-round conversation. These models, through extensive pre-training on large datasets [4], represent significant progress in natural language processing (NLP). Additionally, in the multi-modality domain, frameworks like BLIP-2 [26], LLaVA [30, 31], KOSMOS-1 [18], and PaLM-E [10], have advanced multi-modal comprehension by effectively fine-tuning foundational models such as LLaMA [42] and Vicuna [7]. However, LLMs face limitations due to the quadratic complexity of their self-attention mechanisms, which leads to increased computational and storage demands as input lengths grow. This issue is particularly pronounced in models dealing with long input sequences.

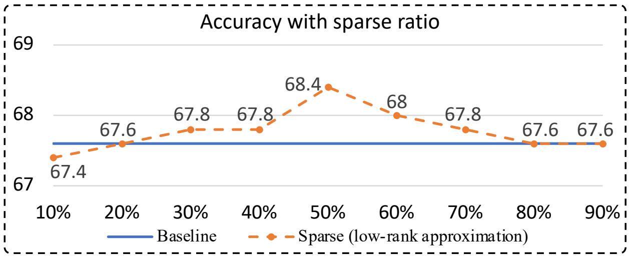


Figure 1. Image accuracy on ScienceQA [32] with the proposed sparse attention method. The baseline model is LLaVA-1.5 in 7B model size. We change the selection ratio in the self-attention computation in the LoRA-Sparse. We show that sparse attention can not only save computation, but also improve performance, with a proper selection ratio, *e.g.*, 30%. The performance gradually goes back the the baseline as the sparsity decreases. In other words, the heavy computation in self-attention layers in LLMs might be redundant and even harmful, while removing the useless attention is beneficial.

To address these concerns, we seek to integrate sparsity into LLMs [17, 28, 46], which offers significant advantages. 1) Computational Efficiency: LLMs typically suffer from high computational costs due to their large parameter count and the quadratic complexity of self-attention computations [4, 10, 38, 50]. Sparse mechanisms improve efficiency by adaptively selecting important tokens for attention computation, especially in longer input sequences [12, 47]. 2) Enhanced Interpretability: The interpretability of sparse mechanisms is particularly vital in LLMs given their large scale, complex training processes, and diverse application areas [42, 43]. Because sparsity compresses the attention map to be more concise and focus on more important tokens. A more interpretable LLM fosters trust and understanding in intricate scenarios. 3) Multi-Modal Correlation Modeling: Sparse attention mechanisms facilitate the explicit modeling of correlations between different modalities [10, 18, 26, 31]. We visualize the sparse attention pattern in multi-modal tasks, which show the correlation between image and text embeddings. This aspect is especially beneficial for multi-modal LLMs, enabling them to more effectively integrate and process information from varied sources.

Despite such benefits, incorporating sparse attention into LLMs faces two main challenges. The first challenge is to keep **efficiency**. Identifying the most relevant pairwise correlations between query (Q) and key (K) elements requires computing the full attention map. However, this would negate the desired acceleration effect. The second challenge arises from the **pre-trained weights**. There is a large gap between the sparse attention and the standard full attention used in the LLMs' pre-training stage. Existing literature typically fine-tunes LLMs based on pre-trained models, such as LLaMA, Llama2, and Vicuna. For most practitioners, starting from the pre-training stage is computationally prohibitive. When directly applying the typical sparse attention during fine-tuning, its performance is unsatisfactory due to the discrepancy with the standard full attention.

To address these challenges, we propose a simple but effective method, Low-Rank Approximation for Sparse Attention (LoRA-Sparse), which significantly enhances the efficiency of LLMs while maintaining their performance. Specifically, LoRA-Sparse approximates the attention map in a low-rank space, enabling the identification of the most relevant Q-K pairs without the need for a full attention map. It effectively reduces computational demands. Specifically, we project the d-dimensional Q and K into a r-dimensional low-rank space ($r \ll d$) and compute an approximate attention map in the low-rank space. Then, we select the most relevant Q-K pairs by sorting the attention scores on the approximate attention map and compute the sparse attention map in the original d-dimensional space using the selected Q-K pairs only. Compared to computing the full attention map, this approach significantly reduces computational complexity while preserving the most important relationships between the elements.

For fine-tuning the low-rank layers, we introduce a novel soft-margin loss function that ensures the low-rank attention map closely mimics the order of attention scores in the full attention map. This loss function, referred to as the Order Mimic Loss, enables a more accurate selection of the most important Q-K pairs hence better performance. Specifically, Order Mimic Loss employs 1) a soft-margin loss to optimize the ordering boundary, enhancing both the efficiency and robustness of the model, and 2) an auxiliary loss to further ensure that the dynamic range of the low-rank attention map closely mirrors the original.

We would like to note that LoRA-Sparse effectively addresses the challenges by allowing easy adaptation of existing pre-trained LLMs to incorporate sparse attention. Since our method operates in a low-rank space and does not modify the model's structure or weights, it can be readily applied to pre-trained LLMs without the need for costly from-scratch training. This makes LoRA-Sparse a practical and efficient solution for leveraging sparse attention in both LLMs and multi-modal LLMs.

In experiments, LoRA-Sparse consistently outperforms standard attention mechanisms by achieving superior performance with significantly reduced computational requirements, as shown in Figure 1. This effectiveness stems from its ability to filter out noise, focusing on the most relevant key-value pairs. LoRA-Sparse excels in processing long input texts, offering better convergence and overall performance than standard attention models. This aspect is particularly crucial for LLMs that frequently handle extensive sequences. Additionally, LoRA-Sparse is demonstrated to be highly adaptable in multi-modal scenarios, effectively managing diverse data types without compromising performance. Its versatility and efficiency make it a robust solution for enhancing LLMs, especially in contexts involving lengthy input sequences and the integration of multiple data modalities.

2. Related Work

2.1. Efficient Attention in Transformer Models

Efficient attention mechanisms in Transformer models, pivotal for addressing computational and memory intensity inherent in their design, have seen significant advancements. Key approaches include fixed patterns [8], combination of patterns [48], learnable patterns [21], neural memory [3], low-rank methods [45], and kernels [9]. Models like the memory compressed transformer and image transformer leverage localized attention spans for handling longer sequences efficiently [44]. The Set Transformer introduces inducing points to manage set-input problems [45], while the sparse transformer employs sparse attention patterns to reduce complexity [8]. The Axial Transformer uniquely applies attention along single axes of input tensors, saving computational resources [3]. These innovations collectively enhance the scalability of Transformer models, making them more feasible for applications dealing with large inputs or long sequences [9, 48] However, as a discrete module, these methods have not been explored in application to pre-trained large language models.

2.2. Efficient Large Language Models

LLMs have received great attention recently in both research and industry areas. Many famous LLMs, including GPT-3 [4], T5 [38], PaLM [10], and OPT [50], LLaMA [42], and Llama2 [43], present notable ability in many NLP tasks, as they are feasible in complex reasoning, multi-round conversation, and having world-level knowledge. One limitation in LLMs would be its computation cost, due to the large amounts of parameters. To overcome

Table 1. Evaluation on ScienceQA [32] for image accuracy with sparse inference. The baseline model is a LLaVA-v1.5 [30] 7B model, where we conduct sparse attention upon it. For the result without low-rank approximation, we use the standard attention to compute the sparse formulation introduced in Section 3.2. For the result with low-rank approximation, we fine-tune the model with the method introduced in Section 3.3. No matter with or without low-rank approximation, the sparse method achieves better performance than the full attention baseline. It shows that sparsity is beneficial for the performance and redundancy indeed exists in the heavy LLMs. With low-rank approximation, the model achieves the best performance, which is 0.8% better than the baseline, at the selection ratio of 0.5. The accuracy goes back to the baseline level as sparsity decreases.

Selection ratio		10%	20%	30%	40%	50%	60%	70%	80%	90%
LLaVA-v1.5 7B (baseline)		67.6								
Low-rank approximation	✗	67.2	67.2	67.5	67.7	67.8	68.3	**68.5**	67.6	67.2
	✓	67.4	67.6	67.8	67.8	**68.4**	68.0	67.8	67.2	67.0

this obstacle, some literature investigates various techniques to make LLMs efficient, including pruning [46], quantization [28], and distillation [17]. In this work, we study the efficiency in attention patterns of LLMs, which is also a large proportion of the overall computation, especially for the long-context inputs.

2.3. Multi-modal Large Language Models

Multi-modal LLMs have also been popular in these days, based upon the development of LLMs, including BLIP-2 [26], FROMAGe [22], KOSMOS-1 [18], and PaLM-E [10]. They are typically trained upon LLMs with image-text pairs. For example, OpenFlamingo [1], LLaMA-Adapter [49], LLaVA [30, 31], and Mini-GPT4 [51] are fine-tuned upon LLaMA [42], Llama2 [43], and Vicuna [7] models. The training approach involves text-image alignment and visual instruction tuning. Similar to LLMs, the multi-modal LLMs also have the computational limitation.

3. Method

In this section, we first review the formulation of self-attention in Section 3.1. Then, we introduce the aspects of our method. We first show that sparse attention can benefit performance in LLMs in Section 3.2. After that, we present a low-rank approximation manner in Section 3.3. Finally, we introduce the loss function in Section 3.4.

3.1. Preliminary

Multi-head Attention in Transformers. Transformers [44] are network architectures that are based on self-attention mechanisms. Given a set of tokens x, for example, text tokens in a sequence or image patches in vision transformers [14], the multi-head attention module first projects them into queries, keys, and values with linear projection layers.

$$Q = xW_q,\ K = xW_k,\ V = xW_v. \tag{1}$$

After that, the multi-attention head computes an attention map M_{attn} between queries and keys via the softmax function.

$$M_{attn} = \textbf{softmax}(QK^T/\sqrt{d}), \tag{2}$$

where d is the dimension of each multi-attention head. We then multiply the attention map M_{attn} with the values V and further project with a linear projection layer W_o.

$$O = (M_{attn}V)W_o. \tag{3}$$

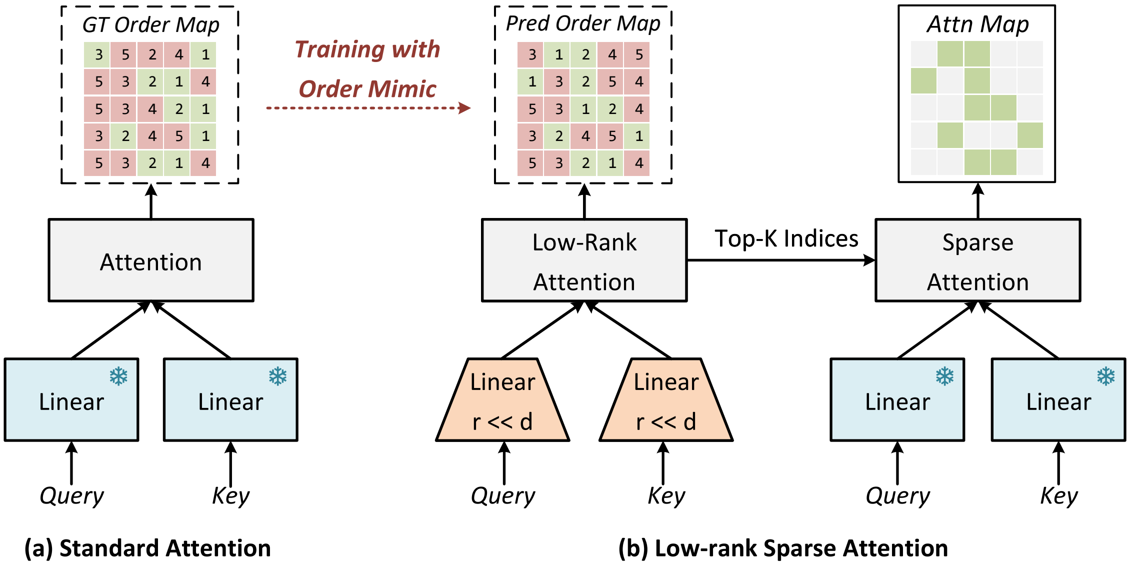
There are typically hundreds or thousands of input tokens x. Thus, the computational cost in the self-attention computation is usually expensive, especially in the case of LLMs or when the number of input tokens is large. A straight idea is that sparse attention would be a proper manner for efficiency. In the following, we study the effects of sparse attention on LLMs.

3.2. Empirical Study on the Sparsity in LLMs

In this section, we empirically show that sparse attention can be beneficial in LLMs. The computations in the self-attention layers in LLMs are somewhat redundant. In other words, in the self-attention computations, some keys and values could be irrelevant to the target, while these irrelevant keys and values still have some influence on the results. We conduct a series of plain experiments that directly sparsify the attention during inference, without fine-tuning.

During inference, we first compute the attention map M_{attn} via the softmax function, find the important entries (*i.e.*, the largest attention scores), and mask the others to be zeros. Specifically, given the sparse ratio $s\%$, for each query, we select the top $s\%$ keys in the attention map M_{attn}. Then we multiply the resultant sparse attention map with the values, following Eq. 3.

In Table 1, we evaluate the sparse attention inference on both LLMs and multi-modal LLMs. For LLMs, we evaluate LLaMA [42] 7B models on NLP benchmarks. For multi-modal LLMs, we evaluate LLaVA [30] 7B models on ScienceQA [32] benchmark. We find that with a proper selection ratio s, clear improvements with sparse attention can



GT Order Map

3	5	2	4	1
5	3	2	1	4
5	3	4	2	1
3	2	4	5	1
5	3	2	1	4

Pred Order Map

3	1	2	4	5
1	3	2	5	4
5	3	1	2	4
3	2	4	5	1
5	3	2	1	4

Figure 2. Overall architecture of the low-rank approximation sparse attention (LoRA-Sparse) for LLMs. Given an attention map, we sort its entries (*i.e.*, attention scores of query-key pairs) to obtain an order map, where the numbers reflect the importance orders for each individual row. For example, a number 1 at (2,3) indicates that the third key is the most important key to the second query. We take the full attention map computed by queries and keys as the Ground Truth (GT) so that the corresponding order map is referred to as the GT Order Map. We introduce two low-rank linear projections for queries and keys, and the order map associated with the attention map approximated with the low-rank queries and keys is referred to as the Predicted Order Map. During training, we supervise the low-rank attention with the standard attention in an order mimic manner, *i.e.*, we let the Predicted Order Map mimic the GT Order Map. During inference, we use the low-rank attention for top keys/values selection, sparsifying the attention by avoiding redundant computation. This example shows that we set the sparse ratio to be 40% so that only the two most important keys out of five keys are selected for each query.

be observed upon baseline results obtained with full attention. For example, on the LLaVA [30] 7B model, sparse attention inference introduces 0.9% accuracy increase.

It is noted that these improvements require no additional fine-tuning and are free-lunch during inference. In other words, due to the trait of softmax and the redundant computation in self-attention layers, it is beneficial to sparsify attention and remove irrelevant keys/values.

3.3. Low-rank Approximation

Although the sparse attention implementation in Section 3.2 is effective, it relies on the attention map M_{attn}. The computation cost of M_{attn} is still expensive. In this section, we introduce a low-rank manner to approximate the attention map. It costs much less than the standard attention map computation, but is still feasible for sparse approximation.

As shown in Figure 2, we introduce two additional linear layers that compress queries Q and keys K into $\hat{Q}$ and $\hat{K}$ in a low-rank space,

$$\hat{Q} = QW_{\hat{q}},\ \hat{K} = KW_{\hat{k}}, \tag{4}$$

where $W_{\hat{q}} \in \mathbb{R}^{d\times r}$, $W_{\hat{k}} \in \mathbb{R}^{d\times r}$, and $r \ll d$.

After that, we use the compressed queries $\hat{Q}$ and keys $\hat{K}$ to compute an attention map $\hat{M}_{attn}$, which is an approximation of the standard attention map M_{attn}. Because $\hat{M}_{attn}$ is much more efficient to compute than M_{attn}, we use $\hat{M}_{attn}$ to estimate the importance in the actual attention map M_{attn}. Similar to Section 3.2, we sparsify and skip unnecessary multiplication among queries, keys, and values.

3.4. Training with Order Mimic

Based on the analysis above, it is evident that sparse attention, which focuses only on the most relevant key-value pairs, can enhance the accuracy of pre-trained language models. To achieve an accurate approximation of the order of query-key relevance, we propose an efficient order-mimic training strategy that aligns the order of sorted attention scores in the approximated attention map $\hat{M}_{attn}$ with that of the original attention map M_{attn}. Given the sparse ratio of $s\%$, for an arbitrary $i-\mathrm{th}$ query, we regard the most relevant $s\%$ keys (*i.e.*, the keys with attention scores higher than the $s\%$ percentile) as positive samples according to the original attention map. The remaining keys are treated as negative samples. The indices of these positive and negative samples are denoted as ϕ_+^i and ϕ_-^i, respectively.

We note that an ideal ordering method needs to rank the positive samples above the negative ones by a large margin. Motivated by this, the corresponding training objective using hard-hinge loss [15] can be formulated as

$$\min_{\theta} \sum_{i} \sum_{\forall \phi_+^i} \sum_{\forall \phi_-^i} \max(\hat{Q}_i \hat{K}^T_{\phi_-^i} - \hat{Q}_i \hat{K}^T_{\phi_+^i} + \lambda, 0), \tag{5}$$

where θ denotes the trainable parameters and λ indicates a pre-defined positive margin. However, this training objective is applied to uniformly sampled pairs, resulting in a greater focus on intra-class distances rather than inter-class distances hence trivial solutions. Moreover, pair-wise relations demand a substantial computational burden. Therefore, we propose to *only optimize the ordering boundary* to avoid trivial solutions and expensive computations:

$$\min_{\theta} \sum_{i} \max(\max_{\phi_-^i}(\hat{Q}_i \hat{K}^T_{\phi_-^i}) - \min_{\phi_+^i}(\hat{Q}_i \hat{K}^T_{\phi_+^i}) + \lambda, 0). \tag{6}$$

To make the objective function smooth and easy to optimize, we further improve it by introducing the soft-margin loss [29] to replace the hard-hinge loss. Thus, the eventually proposed Order Mimic Loss can be formulated as:

$$\mathcal{L}_{\text{order}} = \frac{1}{N} \sum_{i} \log(1 + \exp^{p_i}),$$
$$\text{where } p_i = \max_{\phi_-^i}(\hat{Q}_i \hat{K}^T_{\phi_-^i}) - \min_{\phi_+^i}(\hat{Q}_i \hat{K}^T_{\phi_+^i}).$$

Intuitively, Order Mimic Loss encourages the lowest-scoring positive samples to attain a higher score than the highest-scoring negative sample, thereby fulfilling the ordering requirements of the entire set. Additionally, by focusing only on the most challenging cases, the network pays more attention to the difficult inter-class distances, and the complexity can be reduced from $O(N^3)$ to $O(N^2)$. Nevertheless, we note that the proposed training objective focuses solely on ordering, overlooking the specific magnitude, rendering it sensitive to the initial state of the network and the sample noise, which in turn affects the selection of boundary pairs.

Therefore, to further improve the robustness, we introduced an auxiliary loss to directly constrain the magnitude of the predicted low-rank attention map, ensuring its dynamic range is similar to that of the original attention map. Similar to [36], we adopt a magnitude loss to achieve it:

$$\mathcal{L}_{\text{mag}} = \frac{1}{N^2} \sum -\delta(QK^T) \log \delta(\hat{Q}\hat{K}^T), \tag{7}$$

where $\delta(\cdot)$ indicates the sigmoid function. This approach could provide a better initial state for network training and facilitate more efficient convergence. Furthermore, we fuse the proposed two training objectives to form the training loss for the network:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{order}} + \beta \mathcal{L}_{\text{mag}}. \tag{8}$$

Table 2. The effects of the rank in the attention map approximation. We evaluate our method on the LLaVA-1.5 [30] 7B models and ScienceQA [32] for image accuracy. We find that rank 8 is enough for approximation. There is no further improvement on larger ranks.

Rank	1	2	4	8	16	32
Ours	64.6	66.6	67.6	68.4	68.5	68.5

Through two predefined hyper-parameters, α and β, we can control the significance of rank and magnitude during the training process.

4. Experiments

In this section, we first introduce the experimental setting in Section 4.1. In Section 4.2, we conduct several ablation studies on the factors, including the rank in the low-rank approximation, and the effects of loss weights. After that, we present the main results in both LLMs and multi-modal LLMs in Section 4.3.1 and Section 4.3.2.

4.1. Experimental Settings

4.1.1 Language Models

Training For long-context language modeling, our proposed LoRA-Sparse architecture is fine-tuned on the RedPajama-V2 dataset [11], an expansive collection featuring, for 1000 iterations, with global 64 batch size in the target context length. This dataset is particularly notable for its breadth and quality. For the LLM benchmarks, we fine-tune our models with the Alpaca [41] data. It contains 52k instruction tuning data for supervised fine-tuning.

Evaluation For long-context language modeling, we assess our model using the validation split of PG19 [37] for perplexity. Each selected document contained a minimum of 32,768 SentencePiece tokens, a format optimized for machine learning models dealing with languages. We limited our evaluation to the initial target context length tokens per document to maintain consistency. To effectively measure perplexity across different context lengths, we used a sliding window approach with a stride of 256 tokens, offering a balance between contextual breadth and computational feasibility. For the LLM benchmarks, we use the benchmarks provided in the open-sourced lm-evaluation-harness project[1]. It contains many benchmarks for language models. We use 13 popular benchmarks for evaluation.

[1] https://github.com/EleutherAI/lm-evaluation-harness

Table 3. The effects of loss weights. In this experiment, we use LLaVA-1.5 [30] as the baseline model and evaluate our models on ScienceQA [32] for image accuracy. When the loss weight is 0, it means that the loss is disabled. We find that both order and magnitude losses are helpful, while only one of them presents sub-optimal accuracy. In addition, larger loss weights have no effect.

α	1.0	0	1.0	1.0	2.0	2.0
β	1.0	1.0	0	2.0	1.0	2.0
Acc	**68.4**	68.1	67.4	68.0	67.8	68.2

Table 4. Evaluation on long-context language modeling. We evaluate perplexity on PG19 [37] validation set. Lower perplexity is better. Due to the 2048 context length limitation of LLaMA [42], it presents unsatisfied results in a longer context. Our sparse attention method has effects on long-context language modeling, as it reduces the number of keys and queries that are included in the attention computation. We show that our method maintains good perplexity from 4096 to 16384 context length.

Model Size	Context Length		
	4096	8192	16384
LLaMA 7B + Ours	10.80	10.82	11.01
LLaMA 13B + Ours	9.91	9.98	10.04

Table 5. Comparison to the deformable mechanism that is based on linear interpolation, which is another approach to efficient attention operations. It is observed that directly using the original deformable mechanism degrades the performance. Compared to deformable attention, sparse attention is a better option.

Tasks	arc-easy	arc-challenge	openbookqa	piqa
Baseline	**73.4**	45.6	32.2	**79.4**
Deformable	24.4	21.4	23.0	53.2
Ours	73.2	**47.3**	**32.6**	**79.4**

4.1.2 Multimodality Models

Training In the multimodality experiments of LoRA-Sparse, we meticulously adhere to the dataset configuration as delineated in LLAVA1.5 [30]. For instruction tuning, we engage with a comprehensive mixture of datasets, encompassing the VQA datasets (VQAv2 [16], GQA [19], OKVQA [34]), OCR datasets (OCRVQA [35], TextCaps [39]), region-level VQA (Visual Genome [23], RefCOCO [20, 33]), along with visual and language conversation data from LLaVA [31] and ShareGPT, summing up to a total of 665K instances.

Evaluation To benchmark the efficacy, we leverage a diverse suite of benchmarks that span a spectrum of multi-modal tasks. These benchmarks include GQA [19], TextVQA [40], POPE [27], ScienceQA [32], SEED-Bench [25], and LLaVA-Bench [31]. This comprehensive evaluation strategy ensures that our findings are robust, providing a nuanced understanding of our model's capabilities in interpreting and integrating complex multi-modal data.

4.2. Ablation studies

4.2.1 Rank in the approximation

In Table 2, we ablate the rank of the low-rank projections for attention map approximation. We fine-tune the LLavA-1.5 [30] with 7B parameters without the proposed method and evaluate the image accuracy on ScienceQA [32]. We show that the rank indeed affects the performance. The performance improves as the rank increases to 8. The accuracies with rank 16 and 32 are comparable to that with rank 8. Thus, we set the rank as 8 by default in experiments.

4.2.2 Loss weights

In Table 3, we evaluate the effects of loss weights during fine-tuning. We conduct experiments on LLaVA-1.5 [30] 7B model and evaluate the image accuracy on ScienceQA [32]. With a loss weight of 0, the loss is disabled during fine-tuning. We show that both the magnitude loss and the order loss are effective. We also observe that larger loss weights (*e.g.*, 2) are not helpful. Thus, we set both loss weights to be 1 as a default setting in experiments.

4.3. Main Results

4.3.1 Large language models

LLM benchmarks In Table 6, we evaluate our method on 13 LLM benchmarks. We fine-tune LLaMA 7B on the Alpaca [41] instruction tuning dataset for supervised fine-tuning. The baseline model uses the standard full attention in both fine-tuning and evaluation. Our method uses the low-rank approximation sparse attention that is introduced in Section 3.2 during fine-tuning and inference. We set the sparse ratio as 0.5 in this experiment. We show that our method achieves comparable or even better performance than the full attention baseline. For example, on the arc-challenge benchmark, our method improves the baseline from 45.6% to 47.3% by 1.7%, while saving 50% computation in the self-attention modules. There are only two tasks, winogrande and record, where the performance degrades with sparse attention.

Long-context language modeling In Table 4, we evaluate our method on long-context language modeling. We fine-tune LLaMA [42] 7B model on Redpajama [11] dataset with our low-rank sparse attention approximation method and the position interpolation [6] extension. We evaluate our method on PG19 validation set for perplexity in various

Table 6. Evaluation on LLM benchmarks in the lm-evaluation-harness project. We train the LLaMA [42] 7B model using the Alpaca [41] data for supervised fine-tuning (SFT). The baseline uses full attention in both fine-tuning and evaluation. Our method uses sparse inference and the low-rank approximation method introduced in Section 3.3 during fine-tuning. It shows that on most benchmarks, sparse inference introduces comparable or even better performance than the baseline. On only two benchmarks, arc-easy and record, the performance degrades but the gap is marginal.

Model	wsc	sst	wic	race	hellasway	mrpc	piqa	openbookqa	mnli	record	Rte	arc-easy	arc-challengh
Baseline	**36.5**	70.9	50.2	42.1	60.0	**68.4**	**79.4**	32.2	42.8	**92.2**	63.2	**73.4**	45.6
Ours	**36.5**	**75.8**	**50.3**	**42.5**	**60.1**	**68.4**	**79.4**	**32.6**	**44.0**	91.8	**63.9**	73.2	**47.3**

Table 7. Comparison with multi-modal LLMs on mutli-modal benchmarks including GQA [19], ScienceQA [32], TextVQA [40], POPE [27], SEED-Bench [25], and LLaVA-Bench [31]. Our method achieves comparable or even better performance than the state-of-the-art LLaVA-1.5 [30] model with lower compute. We show that sparse attention is not only efficient but also beneficial to performance.

Method	LLM	GQA	ScienceQA-Image	TextVQA	POPE	SEED-Bench	LLaVA-Bench
BLIP-2 [26]	Vicuna-13B	41.0	61.0	42.5	85.3	**46.4**	38.1
InstructBLIP [13]	Vicuna-7B	49.2	60.5	50.1	-	53.4	60.9
InstructBLIP [13]	Vicuna-13B	49.5	63.1	50.7	78.9	-	58.2
Shikra [5]	Vicuna-13B	-	-	-	-	-	-
IDEFICS-9B [24]	LLaMA-7B	38.4	-	25.9	-	-	-
IDEFICS-80B [24]	LLaMA-65B	45.2	-	30.9	-	-	-
Qwen-VL [2]	Qwen-7B	59.3	67.1	63.8	-	56.3	-
Qwen-VL-Chat [2]	Qwen-7B	57.5	68.2	61.5	-	58.2	-
LLaVA-1.5 [30]	Vicuna-7B	62.0	67.6	**58.2**	85.9	58.6	**63.4**
Ours	Vicuna-7B	**62.4**	**68.4**	**58.2**	**86.8**	**58.8**	**63.4**

context lengths, from 4096 to 16384. We show that with our sparse attention, models maintain much lower perplexity than LLaMA baselines, which have a limited 2048 context length. This shows that sparse attention is beneficial in the long sequence inputs for LLMs.

4.3.2 Multi-modal large language models

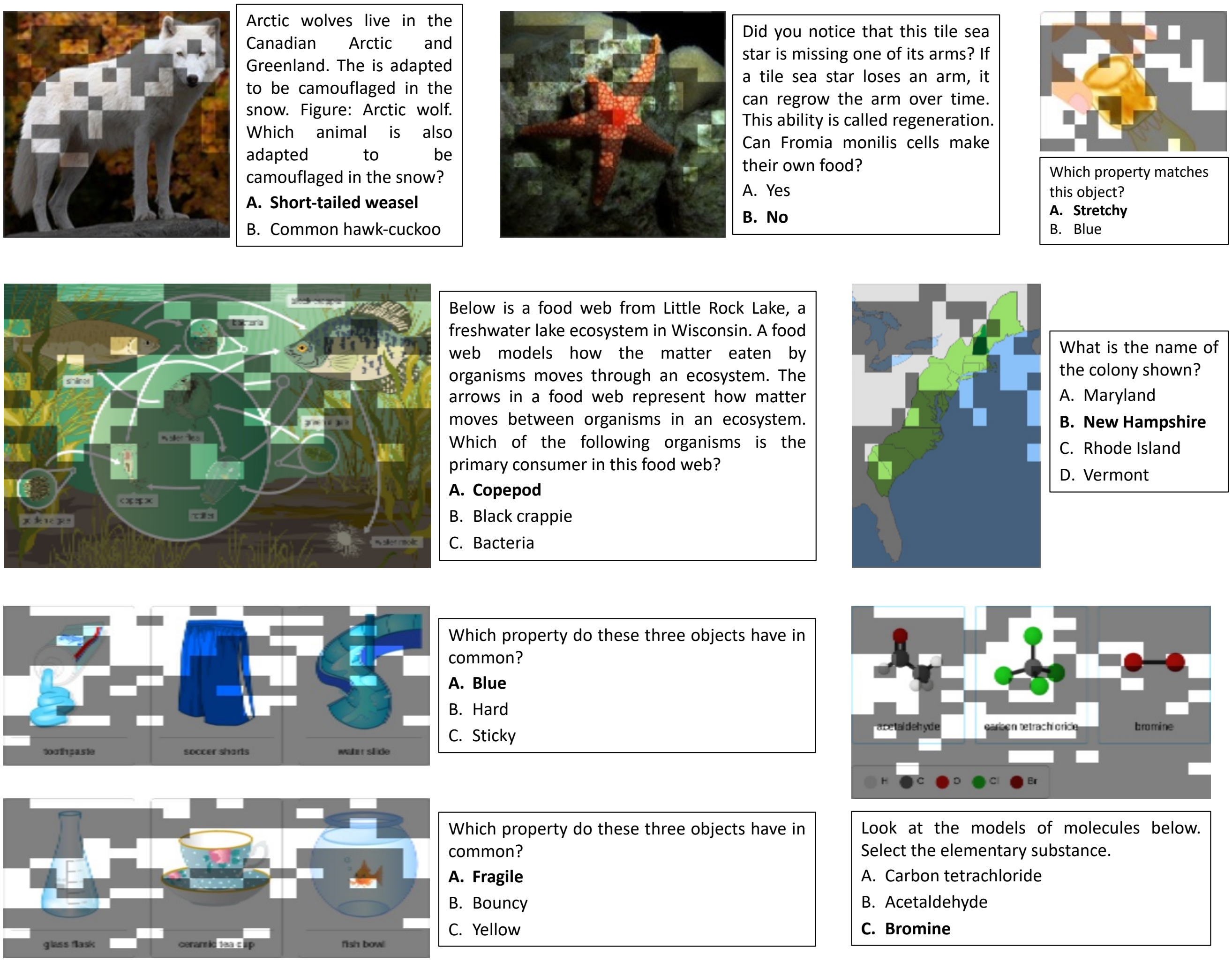
In Table 7, we compare our method with SOTA multi-modal LLMs in benchmarks, including GQA [19], ScienceQA [32] in image accuracy, TextVQA [40], POPE [27], SEED-Bench [25], and LLaVA-Bench [31]. Our method follows LLaVA-1.5 for the settings in fine-tuning and evaluation, except that we include the low-rank approximation for sparse inference. During inference, we set the sparse ratio as 0.5. In this experiment, we show that our method can bring improvements upon the SOTA LLaVA-1.5 baseline, while saving computational cost in the sparse attention, *i.e.*, about 50% attention computation.

4.4. Comparison to Deformable Mechanism

Deformable mechanism [12, 47, 52] is another common approach to efficient models. In this section, we show that the deformable mechanism in attention operations, which learns the attention offsets via (bi-)linear interpolation, can not work well in fine-tuning LLMs. We fine-tune the LLaMA [42] 7B model with the Alpaca [41] data in a supervised fine-tuning or instruction-tuning manner. The baseline model uses full attention in both fine-tuning and inference. Ours is fine-tuned and tested with the proposed sparse attention. We evaluate the models on 4 LLM benchmarks. Table 5 shows that directly using the original deformable mechanism degrades the performance. This phenomenon can be explained that the interpolation mechanism that is commonly adopted by deformable operators is designed for image features and is improper for attention in LLMs, because tokens in LLMs are discrete and not smooth among neighbors. In contrast, our method retains the original attention with additional low-rank layers, which are not only efficient but also compatible with the pre-trained LLMs.

4.5. Visualization

In Figure 3, we show the visualization of attention weights in our low-rank sparse attention upon the LLaVA-1.5 7B model. We evaluate our model on ScienceQA [32] test split. We use the attention weights in the last self-attention layer in the LLM for visualization. We average the attention weights among multi-attention heads and select the top 30% visual keys/values for highlighting. In the visualization, we show that the highlighted areas are highly related to the keywords in questions or the predicted answers (the bold font). This proves the effects of ours.



Arctic wolves live in the Canadian Arctic and Greenland. The is adapted to be camouflaged in the snow. Figure: Arctic wolf. Which animal is also adapted to be camouflaged in the snow?

A. Short-tailed weasel

B. Common hawk-cuckoo

Did you notice that this tile sea star is missing one of its arms? If a tile sea star loses an arm, it can regrow the arm over time. This ability is called regeneration. Can Fromia monilis cells make their own food?

A. Yes

B. No

Which property matches this object?

A. Stretchy

B. Blue

Below is a food web from Little Rock Lake, a freshwater lake ecosystem in Wisconsin. A food web models how the matter eaten by organisms moves through an ecosystem. The arrows in a food web represent how matter moves between organisms in an ecosystem. Which of the following organisms is the primary consumer in this food web?

A. Copepod

B. Black crappie

C. Bacteria

What is the name of the colony shown?

A. Maryland

B. New Hampshire

C. Rhode Island

D. Vermont

Which property do these three objects have in common?

A. Blue

B. Hard

C. Sticky

Which property do these three objects have in common?

A. Fragile

B. Bouncy

C. Yellow

Look at the models of molecules below. Select the elementary substance.

A. Carbon tetrachloride

B. Acetaldehyde

C. Bromine

Figure 3. Visualization of sparse attention weights in multi-modal LLMs. We combine our low-rank sparse approximation method into the LLaVA-1.5 [30] 7B model. We collect the predicted attention weights during the evaluation on ScienceQA [32] test split. We use the attention weights predicted in the last self-attention layer in the model for visualization. In the figure, we highlight the top 30% attentions on the average of attention heads. We show that the top attention is related to the questions or the predicted answer. We highlight the predicted answer in bold font.

5. Conclusion

In this paper, we analyze the sparse attention mechanism in multi-modal LLMs. Surprisingly, sparsity in LLMs is not only efficient but also beneficial to performance. This implies that some of the heavy computations in LLMs are redundant or even harmful. To realize both efficiency and higher accuracy, we introduce a low-rank approximation method, which predicts an attention map in a low-rank manner instead of the standard multiplication between queries and keys. Then, we utilize the order of attention scores in the approximated attention map to select queries and keys for the actual attention computation. In experiments, we show that our method is effective in both LLMs and multi-modal extensions. Results upon LLaMA [42] on language model benchmarks and multi-modal LLaVA [31] models present strong performance and efficiency.

Limitations and Broader Impacts. In this study on Large Language Models (LLMs), we find that sparse attention not only increases efficiency but also boosts performance. This suggests some computational processes in LLMs are redundant. A novel low-rank approximation method is introduced for efficiency and accuracy. However, a key limitation is that sparse attention offers limited GPU acceleration at lower sparsity levels, indicating a need for further optimization in such scenarios.

References

[1] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Yitzhak Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. Openflamingo: An open-source framework for training large autoregressive vision-language models. *CoRR*, abs/2308.01390, 2023. 3

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *CoRR*, abs/2308.12966, 2023. 7

[3] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *CoRR*, abs/2004.05150, 2020. 2

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020. 1, 2

[5] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *CoRR*, abs/2306.15195, 2023. 7

[6] Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *CoRR*, abs/2306.15595, 2023. 6

[7] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 1, 3

[8] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *CoRR*, abs/1904.10509, 2019. 2

[9] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J. Colwell, and Adrian Weller. Rethinking attention with performers. In *ICLR*, 2021. 2

[10] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113, 2023. 1, 2, 3

[11] Together Computer. Redpajama: An open source recipe to reproduce llama training dataset, 2023. 5, 6

[12] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017. 1, 7

[13] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *CoRR*, 2023. 7

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3

[15] Claudio Gentile and Manfred KK Warmuth. Linear hinge loss and average margin. *Advances in neural information processing systems*, 11, 1998. 5

[16] Yash Goyal, Tejas Khot, Aishwarya Agrawal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. *Int. J. Comput. Vis.*, 127(4): 398–414, 2019. 6

[17] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Knowledge distillation of large language models. *CoRR*, abs/2306.08543, 2023. 1, 3

[18] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, Qiang Liu, Kriti Aggarwal, Zewen Chi, Johan Bjorck, Vishrav Chaudhary, Subhojit Som, Xia Song, and Furu Wei. Language is not all you need: Aligning perception with language models. *CoRR*, abs/2302.14045, 2023. 1, 3

[19] Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pages 6700–6709, 2019. 6, 7

[20] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara L. Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, pages 787–798. ACL, 2014. 6

[21] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *ICLR*, 2020. 2

[22] Jing Yu Koh, Ruslan Salakhutdinov, and Daniel Fried. Grounding language models to images for multimodal generation. *CoRR*, abs/2301.13823, 2023. 3

[23] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.*, 123(1):32–73, 2017. 6

[24] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M. Rush, Douwe Kiela, Matthieu Cord, and Victor Sanh. Obelics: An open web-scale filtered dataset of interleaved image-text documents, 2023. 7

[25] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *CoRR*, abs/2307.16125, 2023. 6, 7

[26] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, pages 19730–19742. PMLR, 2023. 1, 3, 7

[27] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *CoRR*, abs/2305.10355, 2023. 6, 7

[28] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. AWQ: activation-aware weight quantization for LLM compression and acceleration. *CoRR*, abs/2306.00978, 2023. 1, 3

[29] Yi Lin. A note on margin-based loss functions in classification. *Statistics & probability letters*, 68(1):73–82, 2004. 5

[30] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *CoRR*, abs/2310.03744, 2023. 1, 3, 4, 5, 6, 7, 8

[31] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *CoRR*, abs/2304.08485, 2023. 1, 3, 6, 7, 8

[32] 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 *NeurIPS*, 2022. 1, 3, 5, 6, 7, 8

[33] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, pages 11–20. IEEE Computer Society, 2016. 6

[34] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. OK-VQA: A visual question answering benchmark requiring external knowledge. In *CVPR*, pages 3195–3204, 2019. 6

[35] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. OCR-VQA: visual question answering by reading text in images. In *ICDAR*, pages 947–952, 2019. 6

[36] 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

[37] Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *ICLR*, 2020. 5, 6

[38] 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. *J. Mach. Learn. Res.*, 21: 140:1–140:67, 2020. 1, 2

[39] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: A dataset for image captioning with reading comprehension. In *ECCV*, pages 742–758. Springer, 2020. 6

[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 *CVPR*, pages 8317–8326. Computer Vision Foundation / IEEE, 2019. 6, 7

[41] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023. 5, 6, 7

[42] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. 1, 2, 3, 6, 7, 8

[43] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023. 1, 2, 3

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 2, 3

[45] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *CoRR*, abs/2006.04768, 2020. 2

[46] Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared llama: Accelerating language model pre-training via structured pruning. *CoRR*, abs/2310.06694, 2023. 1, 3

[47] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *CVPR*, pages 4784–4793. IEEE, 2022. 1, 7

[48] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In *NeurIPS*, 2020. 2

[49] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *CoRR*, abs/2303.16199, 2023. 3

[50] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068, 2022. 1, 2

[51] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *CoRR*, abs/2304.10592, 2023. 3

[52] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets V2: more deformable, better results. In *CVPR*, pages 9308–9316, 2019. 7