On the Performance of Multimodal Language Models

Utsav Garg
Scale AI
utsav.garg@scale.com

Erhan Bas*
GE HealthCare
erhan.bas@gehealthcare.com

Abstract

Instruction-tuned large language models (LLMs) have demonstrated promising zero-shot generalization capabilities across various downstream tasks. Recent research has introduced multimodal capabilities to LLMs by integrating independently pretrained vision encoders through model grafting. These multimodal variants undergo instruction tuning, similar to LLMs, enabling effective zero-shot generalization for multimodal tasks. This study conducts a comparative analysis of different multimodal instruction tuning approaches and evaluates their performance across a range of tasks, including complex reasoning, conversation, image captioning, multiple-choice questions (MCQs), and binary classification. Through rigorous benchmarking and ablation experiments, we reveal key insights for guiding architectural choices when incorporating multimodal capabilities into LLMs. However, current approaches have limitations; they do not sufficiently address the need for a diverse multimodal instruction dataset, which is crucial for enhancing task generalization. Additionally, they overlook issues related to truthfulness and factuality when generating responses. These findings illuminate current methodological constraints in adapting language models for image comprehension and provide valuable guidance for researchers and practitioners seeking to harness multimodal versions of LLMs.

1 Introduction

Large instruction-tuned language models (LLMs) have emerged as powerful models showcasing remarkable zero-shot generalization capabilities across a diverse spectrum of downstream tasks. By learning to interpret natural language instructions, these models obviate the need for task-specific training. However, real-world applications, often involve multimodal data, such as images and text, necessitating the combination of visual and linguistic information for accurate and robust inference. To address this challenge, recent research Li et al. [2023], Ye et al. [2023], Zhu et al. [2023], Liu et al. [2023a] has introduced multimodal variants of LLMs, which integrate independently pretrained large vision encoders with LLMs. These models further undergo instruction tuning, aiming to leverage the combined power of visual and linguistic understanding.

In this paper, we conduct a comprehensive investigation that centers on comparing various approaches to multimodal instruction tuning and assessing their performance on a wide array of downstream tasks. Our study seeks to illuminate the efficacy, generalization capabilities, and limitations of publicly available models and their ablations across various domains. We evaluate the different approaches across a diverse range of tasks from complex reasoning to captioning and classification, to test their generalization capabilities. Moreover, we aim to identify whether specific design choices prove more suitable for specific tasks.

For our experiments, we consider five publicly available approaches for multimodal adaptation of LLMs: BLIP-2 Li et al. [2023], InstructBLIP Dai et al. [2023], LLaVA Liu et al. [2023a], MiniGPT4 Zhu et al. [2023] and mPLUG-Owl Ye et al. [2023]. These approaches encompass a wide

*work done while Erhan Bas was at Scale AI.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

gamut of architectural choices for injecting LLMs with multimodal capabilities. More details about these respective models is in the appendix A.1. By conducting thorough benchmarking and analysis, we aim to identify the strengths and weaknesses of different multimodal instruction tuning strategies. Furthermore, we conduct ablation experiments on previously untested combinations within these approaches to unearth the optimal method for integrating visual capabilities into a language model. This includes exploring the utility of the largest feasible frozen vision encoder, attaching a trained vision head to extract concise visual features, employing a linear mapping layer to project these into the linguistic domain, and fine-tuning the decoder on multimodal data samples.

2 Datasets and Evaluation

We systematically assess all approaches across two primary task categories to comprehensively gauge their capabilities across a broad spectrum of challenges:

1. **Complex Reasoning:** In this task category, the model confronts questions that necessitate external knowledge and cannot be resolved solely based on the image and question. This test predominantly assesses the language model's performance and illuminates the extent to which multimodal instruction tuning influences the base LLM's regression.
2. **Captioning, Classification, Conversation, MCQ:** These tasks become feasible for the LLM only through the incorporation of the visual encoder. The performance across these tasks serves as a robust indicator of the approach's generalization prowess, encompassing architectural choices and instruction tuning data.

To evaluate the general Visual Question Answering (VQA) abilities, we use the test set curated by LLaVA Liu et al. [2023a]. The dataset consists of 30 images from the COCO dataset, each associated with three distinct question types: *conversation*, *description*, and *complex reasoning* types, resulting in a total of 90 instruction-answer pairs. The ground-truth answers for these questions are generated by GPT-4, which is given the captions and bounding boxes associated with the image for context. The evaluation involves ranking both the predicted and reference answers on a scale of 1 to 10 by GPT-4. The metric is the relative score of the predicted answer to GPT-4's answer. To address potential biases due to answer order, as demonstrated by Wang et al. [2023], we adopt the Balanced Position Calibration Wang et al. [2023] (BPC) strategy proposed by them, averaging scores from both possible orderings. Additionally, to account for the stochasticity of the generation, we average scores across 5 generations by each approach. Despite these considerations to make the evaluation as robust as possible, there are still limitations as GPT-4 is not actually seeing the image and is instead using a proxy for visual information. Moreover, we cannot expect GPT-4 evaluations to be perfect underlining the current limitation of open-ended answer evaluations.

For captioning tasks, we evaluate on the *val* set of the NoCaps Agrawal et al. [2019] dataset and report the CIDEr Vedantam et al. [2015] metric. For visual Multiple Choice Questions (MCQ), we evaluate on the *test* split of the ScienceQA Lu et al. [2022] dataset. Here, we exclusively consider samples with available image context. For binary classification, we evaluate on the Visual Spatial Reasoning Liu et al. [2023b] (VSR) dataset. We use the *zero-shot test* split provided in the official GitHub repository.

For tasks that require generation, we use a beam size of 5. For MCQ, we measure the log-likelihood of generating each of the answer options.

3 Experiments

3.1 Comparing Existing Approaches

Table 1 highlights the main differences between existing approaches, include architectural choices and the data used for instruction tuning.

For fair comparison, we compare variants of all approaches with a similarly sized LLM. Two of the approaches (LLaVA and mPLUG-Owl) use a smaller vision encoder (ViT-L), while the others use ViT-g. BLIP-2 does not do any multimodal instruction tuning, while all other approaches are instruction tuned on varying amounts of multimodal data. The vision encoder is frozen in all approaches except

Approach	Instruction Data Size	Vision Encoder		Vision Head		LLM	
		Backbone	Frozen	Type	Frozen	Type	Frozen
BLIP-2	N/A	ViT-g	Yes	Q-Former	No	Vicuna-7B	Yes
MiniGPT-4	3.5K	ViT-g	Yes	Q-Former	Yes	Vicuna-7B	Yes
LLaVA	150K	ViT-L	Yes	N/A	N/A	Vicuna-7B	No
mPLUG-Owl	290K	ViT-L	No	Similar to Perceiver Resampler	No	LLaMA-7B	No
InstructBLIP	15M	ViT-g	Yes	Q-Former	No	Vicuna-7B	Yes

Table 1: Details of the major components of the compared approaches. We compare variants of each approach with a similar decoder size. The instruction data sizes are approximate.

mPLUG-Owl, while the vision head is always trained when used, except for MiniGPT-4. Additionally, both of the approaches using the smaller vision encoder fine-tune their LLMs, while the others keep the language model frozen.

3.2 Performance on Benchmarks

Table 2 presents the results on the various benchmarks discussed in Section 2.

Approach	LLaVA VQA				VSR	ScienceQA (Image)	NoCaps
	Conversation	Detail	Complex Reasoning	Overall			
BLIP-2	61.5	66.7	53.3	60.5	50	53.8	107.5
MiniGPT-4	72.5	68.2	76.4	72.4	57.2	36.04	86.91
LLaVA	75.1	70.6	88.2	78	52.45	34.8	67.75
mPLUG-Owl	74	69.8	86.8	76.9	54.99	34.01	70.82
InstructBLIP	**85.3**	**75.8**	**88.6**	**83.3**	**58.51**	**59.49**	**123.65**

Table 2: Comparison of all publicly available multimodal variants of LLMs on the four datasets discussed above. The scores for LLaVA VQA are relative scores compared to the reference answer as determined by GPT-4. For NoCaps, we use CIDEr and report accuracy for VSR and ScienceQA. The best results for each dataset are in **bold**, and the second-best results are underlined.

For mPLUG-Owl, we use the checkpoint provided by the authors, where the full decoder is fine-tuned instead of the one that uses LORA Hu et al. [2021], as it performs better overall. We ran all benchmarks using the checkpoints provided by the official implementations but directly used the results provided for BLIP-2 in the InstructBLIP paper for VSR, ScienceQA, and NoCaps.

The results demonstrate that InstructBLIP, which performs multitask instruction tuning on a variety of datasets, performs the best overall on all tasks, highlighting the importance of data diversity during instruction tuning. In contrast, all other instruction-tuned approaches perform poorly on tasks they were not trained on, likely due to overfitting to a specific task type. This is evident from the results of LLaVA, mPLUG-Owl, and InstructBLIP, all of which were trained on the LLaVA-150K data and are thus the top three performers on LLaVA VQA. However, only InstructBLIP continues to perform well on out-of-distribution tasks due to its richer instruction tuning data. There are qualitative samples from each of the three LLaVA benchmark categories in the appendix A.2.

3.3 Ablations on Model Components

In this section, we conduct ablations on the LLaVA architecture to analyze the effects and importance of changes to different components, namely the vision encoder, vision head, and data. We chose the LLaVA architecture as the base because its training/evaluation code and data are available, and the LLaVA-150K training set has been used by other approaches as well.

3.3.1 Vision Head

The vision head operates over the patch embeddings of a frozen vision encoder (e.g., CLIP) and compresses the features to extract relevant details. In this section, we specifically compare the effects of using a Querying Transformer (Q-Former) as the vision head. The Q-Former is a small Transformer model that learns queries to extract relevant visual features from the vision encoder. For

all experiments in this section (unless otherwise mentioned), we use an 80K balanced subset of the LLaVA data provided by the authors in the official GitHub repository.

Vision Head		LLaVA VQA				VSR	ScienceQA (Image)	NoCaps
Type	Frozen	Conversation	Detail	Complex Reasoning	Overall			
N/A	N/A	74.8	72.5	88.2	78.5	51.88	34.95	61.25
Q-Former	Yes	75	70.6	87.4	77.6	51.47	**35.35**	71.99
Q-Former	No	**75.5**	**75.1**	**89.9**	**80.2**	**52.7**	**35.35**	**75.45**

Table 3: Comparison of the effect of having and training a vision head. All configurations use the same frozen vision backbone and trained LLM.

The results in Table 3 clearly show that training the vision head over a frozen encoder offers a significant improvement compared to not having a vision head. However, using a frozen vision head does not provide any additional benefit compared to not having one. The improvement in performance comes from training the vision head along with the decoder to align features based on the task requirements.

3.3.2 Variations of Vision Head

In the previous section, we compared the effect of having a vision head. However, it is also possible to have different types of vision heads. The Q-Former architecture supports attending to both text and image together. For instruction tuning, it can be beneficial to have the Q-Former attend to the instruction along with the image, learning image features that are relevant for answering the question.

Vision Head		LLaVA VQA				VSR	ScienceQA (Image)	NoCaps
Type	Frozen	Conversation	Detail	Complex Reasoning	Overall			
Image Q-Former	No	75.5	**75.1**	**89.9**	**80.2**	52.7	35.35	**75.45**
MultiModal Q-Former	No	72.7	74	87	77.8	50.98	35.6	47.74
MultiModal Q-Former*	No	**77.5**	73	88.3	79.6	**54.17**	**36.29**	48.2

Table 4: Comparison of the effect of having an image-only vs multimodal vision head. All configurations use the same frozen vision backbone and trained LLM.

Table 4 presents the results of comparing different types of vision heads. In the "Image Q-Former" row, the Q-Former attends only to the image, while in the "MultiModal Q-Former" row, the Q-Former attends to both the instruction and the image. However, as the instruction tuning data contains multi-turn conversations, the instructions for all turns are concatenated together, which leads to suboptimal performance. The "MultiModal Q-Former*" row modifies the instruction data by breaking all multi-turn conversation samples into separate training points. This version performs mostly on par with the image-only version. The drop in performance is observed in the detail category and captioning (NoCaps), as we do not want to restrict the features to the instruction in these cases, and it is best for the decoder to receive all possible information about the image. These experiments indicate that using instruction-aware visual features does not offer any advantage with open-ended question answering. However, where the final objective is classification, it offers some advantage as also observed by Dai et al. [2023].

3.3.3 Size of Vision Encoder

In the previous experiments, we used ViT-L as the frozen vision encoder. Here, we compare the effect of using a larger vision encoder, specifically ViT-g, for the different configurations mentioned above.

Vision Encoder			LLaVA VQA				VSR	ScienceQA (Image)	NoCaps
Backbone	# Params	Frozen	Conversation	Detail	Complex Reasoning	Overall			
ViT-L	304M	Yes	75.5	75.1	89.9	80.2	**52.7**	35.35	75.45
ViT-g	1.0B	Yes	**81.5**	**79.1**	**90.9**	**83.9**	50.16	**35.6**	**90.89**

Table 5: Comparison of the effect of using a larger frozen vision encoder. All configurations use a trained image-only Q-Former and trained LLM.

Table 5 presents the results comparing the ViT-L and ViT-g vision encoders, with all other configurations remaining the same. The results clearly demonstrate that having a larger vision encoder

(ViT-g) helps improve performance across the board, even on tasks for which the model was not specifically trained (NoCaps). The larger encoder produces a richer image representation, leading to better overall performance.

3.3.4 Effect of Data Size in Alignment Phase

In all the discussed approaches, an alignment stage precedes instruction tuning, where a projection layer is learned to map the outputs of the vision heads to the input space of the language model. In this section, we study the effect of data size during the alignment phase. We use BLIP-2 as the starting point, which has been aligned on 129M image-text pairs. For comparison, we perform alignment using 595K image-text pairs, as done in LLaVA, for the same architecture.

Amount of Alignment Data	LLaVA VQA				VSR	ScienceQA (Image)	NoCaps
	Conversation	Detail	Complex Reasoning	Overall			
129M	**82.6**	76.5	90	83.1	**52.21**	**35.65**	**91.07**
595K	81.5	**79.1**	**90.9**	**83.9**	50.16	35.6	90.89

Table 6: Comparison of the effect of data size during the alignment phase. All configurations use a trained image-only Q-Former and trained LLM.

Table 6 presents the results comparing the alignment with different amounts of data. We observe that aligning the model with a larger amount of data does not significantly improve in-distribution downstream performance when the model is further instruction tuned. This is reasonable as the projection layer, which is trained during alignment, has a small number of parameters. Therefore, increasing the alignment data size eventually leads to diminishing returns. However, for out-of-distribution tasks (VSR, ScienceQA, NoCaps), the larger amount of alignment data does lead to small improvement in performance, most likely due to more robust features.

Amount of LLaVA VQA Data	LLaVA VQA				VSR	ScienceQA (Image)	NoCaps
	Conversation	Detail	Complex Reasoning	Overall			
150K	**75.1**	70.6	**88.2**	78	**52.45**	34.8	**67.75**
80K	74.8	**72.5**	**88.2**	**78.5**	51.88	**34.95**	61.25

Table 7: Comparison of the effect of data size during the instruction tuning phase. All configurations use a ViT-L vision encoder, no vision head, and a trained LLM.

Table 7 shows a similar trend for the quantity of data used in instruction tuning. In all the ablation experiments, we used an 80K balanced subset of the LLaVA instruction data. However, if we compare the performance of our recreated versions of LLaVA in the table above, on the original 150K and the 80K data subset, they perform at par. Beyond a certain point, increasing the amount of data does not significantly improve performance. However, data diversity does play a role, as demonstrated by the superior performance of InstructBLIP in Table 2.

3.3.5 Training the Language Model

The ablations discussed so far all fine-tune the language model following LLaVA. This section compares the effect of training or keeping the decoder frozen across various configurations.

Vision Encoder	Vision Head		LLM Frozen	LLaVA VQA				VSR	ScienceQA (Image)	NoCaps
	Type	Frozen		Conversation	Detail	Complex Reasoning	Overall			
ViT-L	N/A	N/A	Yes	58.1	66.2	75.8	66.7	50.65	34.9	54.31
ViT-L	N/A	N/A	No	**74.8**	**72.5**	**88.2**	**78.5**	**51.88**	**34.95**	**61.25**
ViT-L	Q-Former	No	Yes	68.4	69.1	85	74.1	49.75	**35.45**	67.18
ViT-L	Q-Former	No	No	**75.5**	**75.1**	**89.9**	**80.2**	**52.7**	35.35	**75.45**
ViT-g	Q-Former	No	Yes	72.3	77.6	88	79.3	**50.49**	**36.49**	89.98
Vit-g	Q-Former	No	No	**81.5**	**79.1**	**90.9**	**83.9**	50.16	35.6	**90.89**

Table 8: Comparing the effect of training the decoder. The configurations mention the vision encoder and head used. All experiments use the same LLM (Vicuna-7B).

The experiments in Table 8 show that when instruction tuning on the LLaVA dataset, training the decoder helps in almost all scenarios. The performance improvements are reduced to an extent

when using a larger vision backbone (ViT-g), but there is still some gain. InstructBLIP uses a frozen decoder, but these experiments suggest that we can achieve an additional boost in performance if the decoder is trained as well. Moreover, as the gradients are always propagated through the decoder in this architecture setup, training the decoder does not incur any significant overhead.

4 Discussion and Conclusion

The benchmarks and ablation experiments offer various insights into multimodal LLMs. The main takeaways from this study are as follows:

- **Vision Encoder:** Using a larger vision encoder (ViT-L vs. ViT-g) consistently improves performance across all tasks, as it captures a richer image representation. However, fine-tuning the vision encoder does not improve performance on downstream tasks as observed from the relative performance of mPLUG-Owl in Table 2, which does not keep it frozen.
- **Vision Head:** Fine-tuning a vision head (e.g., Q-Former) is powerful as it enables the model to extract a better representation for the downstream task while also speeding up training and inference due to the smaller output representation. However, using a multimodal vision head that encodes both the image and instruction together does not show any apparent advantage over using an image-only head in open-ended question answering. This suggests that it is more important to condense all visual information and pass that to the language model through grafting and letting the LLM use the full context to answer questions.
- **LLM:** Training the Large Language Model (LLM) during instruction tuning can lead to additional performance gains without significant overhead in training cost, as gradients are propagated through the LLM in all cases. If there are concerns of LLM regression on text only tasks, we can use adapters (LoRA Hu et al. [2021]/IA3 Liu et al. [2022] etc.) to leave the base model unaltered and use the adapter version when inputs are multimodal.
- **Data:** The size of the training dataset becomes less important beyond a certain point in both the alignment and instruction tuning stages. However, data diversity, both within a task and across tasks, remains crucial for achieving superior performance as shown by results of InstructBLIP.

These results provide valuable insights for future research in this direction, guiding architecture choices and emphasizing the importance of data diversity both within and across tasks. The findings suggest that focusing on larger frozen vision encoders, training vision heads, and optimizing the LLM can yield improvements in performance on multimodal tasks. Moreover, exploring diverse and representative datasets can contribute to achieving state-of-the-art performance.

This work also highlights that differences in architectural methodologies for grafting multimodal capabilities into LLMs. A simple strategy of extracting relevant condensed visual features and transforming them via linear projections to the language space performs as well as any other. The major areas to focus on are data curation and task diversity. Another challenge the community should focus on are hallucinations of these multimodal variants. Language models are unaware of visual concepts during pretraining and tend to hallucinate objects/concepts in images that might not exist when probed about them instead of answering no. To make these models useful, a focus on alleviating these hallucinations will be essential.

In conclusion, this study sheds light on the key components and strategies for building effective multimodal Large Language Models. It focuses on the limitations of current research, and highlights focus areas that will bring the largest impact to their capabilities and usefulness.

References

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023a.

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. *The 37th Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*, 2023.

Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. nocaps: novel object captioning at scale. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8948–8957, 2019.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

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.

Fangyu Liu, Guy Edward Toh Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 2023b.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568, 2021.

Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24, 2011.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.

Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 742–758. Springer, 2020.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. *arXiv*, 2022.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*, 2019.

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.

OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. `https://github.com/kakaobrain/coyo-dataset`, 2022.

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.

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, March 2023. URL `https://lmsys.org/blog/2023-03-30-vicuna/`.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *arXiv preprint arXiv:2304.01196*, 2023.

A Appendix

A.1 Compared Approaches

For our experiments, we consider the five publicly available approaches listed below.

- **BLIP-2 Li et al. [2023]:** Aligns a frozen vision encoder and frozen language model through a Querying Transformer (Q-Former) and a linear projection layer. The Q-Former is a small Transformer model that learns queries to extract relevant visual features from the vision encoder. The model is trained in two stages: first, to learn the image representation, and then to translate the learned representation through the LLM. It is pretrained with 129M image-text pairs from COCO Lin et al. [2014], Visual Genome Krishna et al. [2017], CC3M Sharma et al. [2018], CC12M Changpinyo et al. [2021], SBU Ordonez et al. [2011], and LAION400M Schuhmann et al. [2021].
- **InstructBLIP Dai et al. [2023]:** Uses the same architecture as BLIP-2 and performs multimodal instruction tuning. During instruction tuning, the instruction is also passed along with the image to the Q-Former to extract instruction-relevant visual features. It translates multiple captioning and VQA datasets into an instruction-answer format and performs instruction tuning in a multi-task setup. It is trained on around 15M image-instruction pairs from COCO Caption Lin et al. [2014], Web CapFilt Li et al. [2022], TextCaps Sidorov et al. [2020], VQAv2 Goyal et al. [2017], OKVQA Marino et al. [2019], A-OKVQA Schwenk et al. [2022], OCR-VQA Mishra et al. [2019], and LLaVA-150K Liu et al. [2023a].
- **LLaVA Liu et al. [2023a]:** Has a CLIP Radford et al. [2021] ViT-L vision encoder mapped to a Vicuna decoder through a linear layer. It follows a two-stage training process: in stage-1 (alignment), only the linear mapper is trained to align the visual output with the language input. In stage-2, both the mapper and LLM are trained to output answers to image-instruction pairs. For stage-1, 595K image-text pairs from CC3M Sharma et al. [2018] are used. For stage-2, they create a dataset of 158K image-instruction pairs across conversation, description, and complex reasoning tasks by prompting the language-only GPT-4 OpenAI [2023].
- **MiniGPT4 Zhu et al. [2023]:** Follows the same architecture as BLIP-2 but uses only the pretrained Q-Former and learns the linear mapping layer through the same alignment process as LLaVA. It also follows a two-stage training paradigm, but in both stages, only the linear mapping layer is trained. For stage-1, they use 5M image-text pairs from LAION, SBU, and Conceptual Captions. For stage-2, they self-curate a high-quality instruction dataset with 3.5K instruction-image pairs to improve the model's conversational ability.
- **mPLUG-Owl Ye et al. [2023]:** This approach uses a CLIP ViT-L vision encoder, a visual abstractor which follows a similar architecture to Perceiver Resampler Alayrac et al. [2022], a linear mapping layer, and a LLaMA Touvron et al. [2023] decoder. They also follow a two-stage training setup, but in this work, all parts of the model are trained during one of the two stages. The visual abstractor serves a similar purpose to the Q-Former and is trained in stage-1 along with the visual encoder to extract relevant features. In stage-2, the visual encoder and abstractor are frozen, and the LLM is trained. For stage-1, they use a massive dataset of about 1.1B image-text pairs from LAION-400M, COYO-700M Byeon et al. [2022], Conceptual Captions, and MSCOCO. For stage-2, they use around 290K instruction samples from Alpaca Taori et al. [2023], Vicuna Chiang et al. [2023], Baize Xu et al. [2023], and LLaVA-150K.

A.2 Qualitative comparison of benchmaked approaches

Image and Question	Model	Generations
What is the main focus of the image? (Coversation)	BLIP-2	Is it the cat or the laptop?
	InstructBLIP	The main focus of the image is a cat sitting on a desk in front of a laptop computer.
	LLaVA	The main focus of the image is a cat sitting on a desk and looking at a laptop.
	MiniGPT-4	The main focus of the image appears to be a cat sitting on top of a laptop computer that is open and connected to a keyboard. The cat is sitting on the right side of the laptop, facing the camera. The background of the image appears to be a cluttered desk with papers, pens, and other items scattered around.
	mPLUG-Owl	The main focus of the image is on a cat sitting on top of a laptop and looking at the screen.
Why might the man be holding the clothes on hangers? (Complex Reasoning)	BLIP-2	He might be getting ready to pack for a trip.
	InstructBLIP	The man might be holding the clothes on hangers because he is preparing to pack them into a suitcase. In the image, he is standing next to a window with a suitcase open, indicating that he is in the process of packing his belongings for a trip. By holding the clothes on hangers, he can ensure that they maintain their shape and don't get wrinkled or creased during the packing process. Additionally, hanging the clothes on hangers helps to conserve space in the suitcase and makes it easier to access and organize the items when unpacking at the destination.
	LLaVA	The man could be holding the clothes on hangers for a variety of reasons. He might be preparing to hang the clothes in a closet or dressing area in his home, or he could be taking them to a dry cleaning store or laundromat to have them cleaned. The image does not provide enough context to determine the exact reason, but it is clear that the man is in the process of handling the clothes on hangers.
	MiniGPT-4	It is not clear why the man is holding the clothes on hangers. It could be that he is trying to decide what to wear, or he could be holding the clothes to take them to a dry cleaner. Without more information, it is difficult to say for certain why the man is holding the clothes on hangers.

Image and Question	Model	Generations
	mPLUG-Owl	The man could be holding the clothes on hangers as he is either organizing his closet, packing for a trip, or getting ready to hang them up in his room or a hotel room. Hanging the clothes in a well-organized manner can help keep them clean and prevent wrinkles, especially if he is packing for a trip. Additionally, he may be preparing for a photo shoot or attending an event where the clothing needs to be presented nicely. Holding the clothes on hangers helps him maintain their appearance and ensure they are ready for display, presentation, or use.
What do you see happening in this image? (Detail Description)	BLIP-2	There are two airplanes in the picture. One is landing and the other is taking off. There are also a lot of birds in the water. The mountains are in the background.
	InstructBLIP	The image depicts two large airplanes flying over a body of water, possibly an ocean or a lake. One of the planes is in the process of taking off, while the other is already airborne. There are several birds scattered throughout the scene, with some perched on the water's surface and others flying in the sky. In total, there are at least 10 birds visible in the image. The airplanes are positioned at different heights, with one closer to the water and the other higher up in the sky. Overall, the scene captures the beauty of aviation and nature coexisting in harmony.
	LLaVA	The image features a large airplane taking off from an airport runway, with the airplane's wings fully extended. In the background, there is another airplane on the runway, possibly preparing for takeoff or landing. The scene is set on a body of water, with several birds in the vicinity of the airplanes. A flock of birds is resting near the first airplane, while another flock is situated in the background of the image. The combination of the airplanes, birds, and water contribute to a picturesque aerial view.
	MiniGPT-4	I see a large passenger jet airplane flying over a body of water with some birds in the foreground.
	mPLUG-Owl	The image captures a large airplane taking off from a runway, surrounded by a group of birds. The birds appear to be flying in a nearby area, likely observing the airplane as it ascends. There is also a flock of birds flying in the air.

Table 9: Qualitative comparison of the publicly available multimodal LLMs on a selected sample of image-question pairs from the LLaVA test set.