An Empirical Study of Scaling Instruction-Tuned Large Multimodal Models

Yadong Lu^{*1}, **Chunyuan Li**^{*2}, **Haotian Liu**^{3}, **Jianwei Yang**^{2}, **Jianfeng Gao**^{2}, **Yelong Shen**^{1}
^{1}Microsoft Azure AI ^{2}Microsoft Research ^{3}University of Wisconsin–Madison

Abstract

Visual instruction tuning has recently shown encouraging progress with open-source large multimodal models (LMM) such as LLaVA and MiniGPT-4. However, most existing studies of open-source LMM are performed using models with 13B parameters or smaller. In this paper we present an empirical study of scaling LLaVA up to 33B and 65B/70B, and share our findings from our explorations in image resolution, data mixing and parameter-efficient training methods such as LoRA/QLoRA. These are evaluated by their impact on the multi-modal and language capabilities when completing real-world tasks in the wild. We find that scaling LMM consistently enhances model performance and improves language capabilities, and performance of LoRA/QLoRA tuning of LMM are comparable to the performance of full-model fine-tuning. Additionally, the study highlights the importance of higher image resolutions and mixing multimodal-language data to improve LMM performance, and visual instruction tuning can sometimes improve LMM's pure language capability. We hope this study makes state-of-the-art LMM research at a larger scale more accessible, thus helping establish stronger baselines for future research. Code and checkpoints will be made public.

1 Introduction

Recent studies on large multimodal models (LMM) [9, 10] have been focused on the methods of *visual instruction tuning* [12]. The results are promising: *e.g.,* the open-source project Large Language and Vision Assistant (LLaVA) shows that training a 7B large language model (LLM) with multimodal instruction-following data for 3 hours on 8 A-100 GPUs leads to a LMM with strong visual understanding and reasoning capabilities in the wild: reproducing some of the most appealing examples of the proprietary OpenAI multimodal GPT-4 model [14]. A similar idea is explored in its co-current work MiniGPT-4 [21]. It has rapidly become a prominent research topic, spurring the development of numerous new models, benchmarks, and applications [10]. However, the high compute cost has led most existing studies to utilize 7B and 13B LLMs. Thus, the impact of significantly scaling up the model size to *e.g.,* 33B and 65B remains unexplored.

This study aims to fill this gap by empirically investigating language models of larger sizes for LMM, sharing insights of our scaling experiments and establishing stronger baselines using larger-scale LLaVA for future research. Specifically, we explore the impact of larger model sizes, model tuning and data mixing methods on model performance, and present our findings and recommendations. The scaling recipe leads to new state-of-the-art (SoTA) performance on LLaVA-Bench [12] and MM-VET [20]. We hope that our findings and larger LLaVA checkpoints would provide a reference for future research on visual instruction tuning.

Workshop on Instruction Tuning and Instruction Following at NeurIPS 2023.

2 Experiment Setup

Model Checkpoints. To study the impact of scaling up LLM on multimmodal capabilities, we increase the language model size to 33B and 65B [15], in addition to the 7B and 13B models used for existing LMM.

- **LLaVA-33B** We employ the open source Vicuna-33B checkpoint [1] [17] to preform the two-stage training. The training data is around 125K conversations collected from ShareGPT.com.
- **LLaVA-65B** Due to a lack of public 65B Vicuna checkpoint, we conduct our own training of the Vicuna-65B model, utilizing ShareGPT data that we have independently processed. This data contains 159M tokens used during training. As a comparison, the reported number of tokens used in training Vicuna 33B is 370M [2].

Once the instruction-tuned LLM is given, we follow [12] to perform the two-stage LLaVA lightning training: (i) *Stage 1: Pre-training for Feature Alignment.* The linear projection layer is trained, which maps the visual feature (the features before the last layer of the pre-trained image encoder) to word embedding space of LLM. More specifcally, the projection dimension is 1024→6656 for the 33B model and 1024→8192 for the 65B model, respectively. In this stage, we use the concept-balanced subset of LAION-CC-SBU data with 558K samples. (ii) *Stage 2: Visual Instruction Tuning.* We use the LLaVA-80K multimodal instruct dataset for the fine-tuning stage. Various training schedules are explored to enable the model to follow the diverse instructions to complete tasks in the wild, as to be detailed below.

Tuning Methods. We explore both the trainable modules and training data mixing for efficient and effective visual instruct tuning of large models.

- **Trainable modules.** In addition to tuning the linear projection layer, two schemes are considered to tune the LLM: (i) Full-model fine-tuning of LLM and (ii) Parameter-efficient training methods. For the latter, LoRA [7] and QLoRA [4] are employed to allow us to tune large models with limited compute resource. This aims to gain an in-depth understanding of the trade-off between the training cost and model performance.
- **Data mixing.** Typically only the multimodal instruction data is used in Stage-2. We further consider mixing the language-only instruct data ShareGPT with the LLaVA-80K multimodal instruction data to gain an in-depth understanding of the trade-off between models' language and multimodal capabilities.

Hyper-parameters. In the training process of both stages, we utilize the DeepSpeed library [3] and employ the ZeRO3 optimizer, except for QLoRA runs we use ZeRO2. We use a maximum sequence length of 2048. For Stage 1, we train both the 33B and 65B models with a learning rate of 1×10^{-4} with no weight decay, and a learning rate with linear decay and linear warmup for 3% of training steps in total. For Stage 2, we use a learning rate of 2×10^{-5} in full fine-tuning to train 1 epoch for all the models in full finetuning, and a learning rate of 1×10^{-4} for the LoRA/QLoRA runs. We conducted a set of hyperparameter search and for LoRA runs, and found larger LoRA alpha or equivalently larger learning rate was crucial to get the best performance. Specifically, we use LoRA alpha equals 2 times the LoRA rank, and a learning rate of 1×10^{-4}, which works the best for all the models. For full fine-tuning, we use a total batch size of 512 on 4 A100 nodes, where each of these nodes is equipped with 8 A100-80G GPUs. For LoRA/QLoRA runs, we use a total batchsize of 64 on 1 A100 node for 33B model and 2 nodes for 65B model.

3 Results

We first compare our large checkpoints on two recent benchmarks which are specifically designed for LMM, then report our findings in the course of scaling up LLaVA models.

[1] https://huggingface.co/lmsys/vicuna-33b-v1.3

[2] https://github.com/lm-sys/FastChat/blob/main/docs/vicuna_weights_version.md

[3] https://github.com/microsoft/DeepSpeed

Models	Reasoning	Conversation	Detail	Overall
Bard-0718	78.7	83.7	69.7	77.8
Bing-Chat-0629	90.1	59.6	52.2	71.5
LLaVA-13B (beam=1)	81.7	64.3	55.9	70.1
LLaVA-13B (beam=5)	84.3	68.4	59.9	73.5
LLaVA-33B (beam=1)	82.9	70.2	62.6	73.9
LLaVA-33B (beam=5)	83.5	72.6	61.9	74.8
LLaVA-65B (beam=1)	87.3	63.8	62.3	74.2
LLaVA-65B (beam=5)	88.7	59.4	65.7	74.4

Table 1: The performance comparison on LLaVA-Bench. Beam search sizes at 1 and 5 are reported.

Model	Rec	OCR	Knowledge	Generation	Spatial	Math	Total
Results of various open-source LMM on reported in the MM-VET paper [20]							
OpenFlamingo-9B [1, 2]	24.6	14.4	13.0	12.3	18.0	15.0	21.8±0.1
MiniGPT-4-8B [21]	27.4	15.0	12.8	13.9	20.3	7.7	22.1±0.1
BLIP-2-12B [11]	27.5	11.1	11.8	7.0	16.2	5.8	22.4±0.2
LLaVA-7B [12]	28.0	17.1	16.3	18.9	21.2	11.5	23.8±0.6
MiniGPT-4-14B [21]	29.9	16.1	20.4	22.1	22.2	3.8	24.4±0.4
Otter-9B [8]	28.4	16.4	19.4	20.7	19.3	15.0	24.6±0.2
InstructBLIP-14B [3]	30.8	16.0	9.8	9.0	21.1	10.5	25.6±0.3
InstructBLIP-8B [3]	32.4	14.6	16.5	18.2	18.6	7.7	26.2±0.2
LLaVA-13B [12]	30.9	20.1	23.5	26.4	24.3	7.7	26.4±0.1
MM-ReAct-GPT-3.5 [19]	24.2	31.5	21.5	20.7	32.3	26.2	27.9±0.1
LLaVA-7B (LLaMA-2) [12]	32.9	20.1	19.0	20.1	25.7	5.2	28.1±0.4
LLaMA-Adapter v2-7B [5]	32.9	20.1	19.0	20.1	22.9	3.9	31.4±0.1
LLaVA-13B (V1.3, 336px) [12]	38.1	22.3	25.2	25.8	31.3	11.2	32.5±0.1
LLaVA-13B (LLaMA-2) [12]	39.2	22.7	26.5	29.3	29.6	7.7	32.9±0.1
MM-ReAct-GPT-4 [19]	33.1	65.7	29.0	35.0	56.8	69.2	44.6±0.2
Results with our own experiment runs							
LLaVA-13B (LLaMA-2)	38.4	21.0	26.3	28.8	28.0	7.7	32.6±0.1
LLaVA-33B	38.5	25.0	26.2	28.2	29.2	7.7	32.9±0.3
LLaVA-33B (Data Mixing)	37.7	27.1	26.2	28.6	28.1	11.5	34.1±0.3
LLaVA-65B	39.2	28.2	26.2	28.3	33.0	15.0	35.5±0.3
LLaVA-65B (Data Mixing)	41.8	27.9	30.4	32.3	30.5	7.3	**36.4±0.2**

Table 2: Performance of various open-source LMM on MM-VET. Note that MM-ReAct is not an single multimodal model, it is a system built on chaining visual tools via GPT-3.5 or GPT-4, which we append as a reference. Our experiment run on LLaVA-13B (LLaMA-2) yields very similar score with the same checkpoint reported in MM-VET paper, indicating that our evaluation pipelines are consistent.

3.1 Comparisons on Benchmarks

LLaVA-Bench. LLaVA-Bench (In-the-Wild)[4] [12] is a diverse evaluation dataset consisting of 24 images with 60 questions in total, including indoor and outdoor scenes, memes, paintings, sketches. Each image is paired with a manually-curated, detailed description and a set of properly-selected questions related to open-ended visual chat scenarios. Each questions belongs to one of three types of tasks: conversations that contain simple visual recognition & QA questions, detailed descriptions that characterize the image with a long paragraph, and a complex reasoning task that focuses on deducing implications from an image. Language GPT-4 (`gpt4-0314`) is used to score to the generated answers. The relative scores between the model output and gold response are reported. We compare LLaVA against the commercial visual chat systems including Microsoft BingChat[5] and Google Bard[6] on LLaVA-Bench [12].

[4] https://github.com/haotian-liu/LLaVA/blob/main/docs/LLaVA_Bench.md

[5] https://www.bing.com/chat

[6] https://bard.google.com/

The results are presented in Table 1. The 33B and 65B checkpoints outperform the 13B LLaVA model and Bing Chat. Despite the fact that LLaVA-Bench is small (thus the comparison might not be statistically significant), the results are encouraging: compared to large LMM, small open-sourced LMM are far more cost-effective to be deployed in real-world applications. With negligible increase of inference latency, we can significantly improve the performance for all model sizes by increasing the beam search size from 1 to 5. Our results show that larger LLaVA models generally exhibit better performance in tasks involving complex reasoning and generating detailed descriptions, which requires strong language competencies from larger LLM. In addition, larger LLaVA models obtain comparable results to BingChat in multi-turn, multi-modal conversation tasks that require strong image understanding capability.

MM-VET. MM-VET [20] is designed based on the assumption that the intriguing capability of solving complicated tasks is often achieved by a generalist LMM which is able to integrate a varity of vision-language (VL) capabilities. MM-Vet contains 200 images and 218 questions (samples), aiming to evaluate6 core VL capabilities (recognition, OCR, knowledge, language generation, spatial awareness, and math) and their combinations. For evaluation, an LLM-based evaluator (`gpt4-0613`) is used to score open-ended outputs of different forms. In Table 2, we report the results on MM-VET. The performance is consistently improved from 13B to 33B and 65B. The largest LLaVA model improves SoTA performance among the end-to-end open-source LMM. The most significant improvements are observed when evaluating the capabilities of knowledge and generation, followed by recognition and OCR. The performance on spatial and math remains comparable. The result reveals that the improved LLM capability is instrumental in storing more knowledge in the weights and leading to a stronger language responding capability.

3.2 Scaling up LLaVA

The experiments are conducted to answer three research questions.

① **Which scaling factor matters?** We study the relative contribution of three scaling-up factors to the performance improvement of LLaVA. The results are summarized in Table 3 (a).

- **Model size.** Increasing the model size consistently improves the overall performance. We conjecture that larger data size is essential to train a larger model. For example, if we only train on LLaVA-80K data, we see smaller gain when model size becomes larger.
- **Image resolution.** By fixing the CLIP ViT image encoder, we compare the variants that are pre-trained to take image resolution 224×224 and 336×336, and find that the higher resolution consistently yields 2-3 points improvement across all four LLM sizes.
- **Data mixing.** Larger models tend to have higher capability of fitting the instruction data. By mixing the language-only instruction data (ShareGPT) with LLaVA-80K, we can improve model performance by 2 points, compared to training on multimodal instruction data only.

In Table 3 (b), we present our result on MM-Bench [13], which contains a set of 2,974 questions, which evaluate models' reasoning skills of six categories. The combination of the three factors improve the baseline LLaVA 7B model, reported in [13].

② **When should the parameter-efficient training method be considered?** As model size increases, it becomes necessary to consider using tuning methods that are more efficient than full-model fine-tuning. LoRA and QLoRA are well-known parameter-efficient tuning methods. As shown in Table 4, we report compute cost using *GPU hours per node*, because the unit can be equivalent to the price $13.63/hour (ND A100 v4 series) on Azure [7]. The total cost can be estimated by multiplying the #hours and #epochs.

In Table 4(a), we train both the 33B and 65B model with LoRA rank 8 and 64 for 1 epoch on the LLaVA-80K instruction-tuning dataset. For models with 33B parameters and above, as we increase the LoRA rank values, we notice an increase in both performance and cost until full-model tuning reaches its maximum performance for a specific model size. In the case of the 13B model, we find that a rank of 64 can deliver comparable performance to full-model tuning. The cost is more related to the total number of parameters than the number of trainable parameters. The cost increase due to raising

[7] https://azure.microsoft.com/en-us/pricing/details/machine-learning/

Image Size	Data Mixing	7B	13B	33B	65B
224×224	✗	63.6	67.1	69.3	70.3
336×336	✗	65.9	70.1	72.0	72.3
336×336	✓	–	–	73.9	74.2

(a) Performance scores on LLaVA-Bench.

Checkpoint	Image Size	Data Mixing	Overall	LR	AR	RR	FP-S	FP-C	CP
LLaVA-7B	224×224	✗	36.2	15.9	53.6	28.6	41.8	20.0	40.4
LLaVA-33B	336×336	✓	55.7	23.3	74.0	46.0	51.5	50.4	67.2
LLaVA-65B	336×336	✓	56.0	24.4	72.3	49.3	50.5	51.2	68.1

(b) Performance scores on MM-Bench. The skills to evaluate include logic reasoning (LR), attribute reasoning (AR), relation reasoning (RR), fine-grained single-instance perception (FP-S), fine-grained cross-instance perception (FP-C), and coarse perception (CP).

Table 3: The performance to scale up model size, image resolution and data mixing.

	7B	13B		33B				65B	
LoRA Rank	Full	64	Full	8	64-QLoRA	64	Full	64	Full
Performance ↑	65.9	70.1	70.1	70.3	71.6	71.8	72.0	72.2	72.3
Time (GPU Hours per node) ↓	1.3	2.1	2.3	4.62	4.68	4.79	5.80	9.17	13.50
# Trainable Parameters (B) ↓	7	0.26	13	0.06	0.49	0.49	33	0.81	65

Table 4: The trade-off between performance and compute cost among different model sizes and traing methods on LLaVA-80K data. "Full" indicates the full-model fine-tuning. "Time" is reported as the total GPU hours to finish 1 epoch training (running time × #GPUs) divided by 8 (#GPUs per node). All models are trained on LLaVA-80K data, results are obtained through averaging 3 repeated evaluation runs with same set up on LLaVA-Bench.

the LoRA rank for a given model size is significantly smaller than the cost increase by enlarging model sizes. For example, increasing the LoRA rank from 8 to 64 nearly matches the performance as LoRA fine-tuning a 65B model with same rank, but only requires 50% of 65B model's training cost. In practice we find that LoRA fine-tuning 33B model provide a good trade-off between cost and performance.

Different LoRA variations have similar performance, and QLoRA requires slightly lower GPU memory cost and running-time cost than LoRA. In the experiments, we find that the hyperparameters of LoRA have a large impact of performance:(i) Large learning rate and alpha value of LoRA improves the results significantly. For example, With the same rank=64, we reduce the learning rate=2×10^{-5} and alpha=16, the performance decrease from 71.8 to 65.5 on LLaVA-Bench. (ii) Under the same setting, large ranks leads to little improvement. *e.g.,* we increase the rank from 64 to 128 and 512, it improves from 65.5 to 66.1 and 68.1, respectively.

③ A LMM with strong capabilities in both language and multimodal? We expand our evaluation in two aspects: (i) MM-VET is added to measure the integrated multimodal capabilities of LMM; (ii) The pure language ability of LMM is measured using Vicuna-80 [17] and MMLU [6], where the former evaluates the instruct-following ability in real-world language tasks, the latter evaluates the multi-task language ability. The results are shown in Table 5, where all models are full-model fine-tuned.

Compared to Vicuna which initializes the LLM weights of LLaVA, it is surprising to observe that LLaVA, after being trained solely on multimodal instruction data, exhibits a comparable language capability. Mixing language instruction data can boost LLaVA's multimodal ability, but not the language ability. This is partially attributed to the inclusion of complex reasoning questions, and long-form answers in LLaVA-Instruct-158K, which helps maintain the language capabilities of LLaVA. We also train LLaVA-70B based on the LLaMA-2-70B-Chat checkpoint [15], and find that mixed results on multimodal and language abilities. Interestingly, we improve LLaMA-2-70B-Chat

Model	Data Mix	Multimodal		Language	
		LLaVA-Bench	MM-VET	Vicuna-80	MMLU
Vicuna-13B	-	-	-	79.9	55.8
LLaVA-13B	✗	70.1	32.5	79.6	55.0
Vicuna-33B	-	-	-	85.6	59.0
LLaVA-33B	✗	72.0	32.9	85.3	56.1
LLaVA-33B	✓	73.9	34.1	80.3	58.6
Vicuna-65B	-	-	-	83.2	62.5
LLaVA-65B	✗	72.3	35.5	84.5	62.6
LLaVA-65B	✓	74.2	36.4	82.6	62.2
LLaMA-2-70B-Chat	-	-	-	84.7	63.1
LLaVA-70B	✓	69.8	35.4	81.3	**65.1**

Table 5: Performance on both multimodal and language capabilities.

by 2.4 points on MMLU, yielding an overall MMLU score of 65.1, which is the best performance for the 70B chat model size, according to [18] and the Chatbot Arena Leaderboard [8]. The original LLaMa-2-70B yields 68.9 [15]. To the best of our knowledge, this is the first reported result which shows visual instructing tuning improves language ability of large-scale LMM, among the co-current work [16] showing the improvement in language truthfulness and ethics.

4 Conclusions and Limitations

We present an empirical study of scaling the language model size for LMM. Our main findings are: (i) Scaling LMM consistently enhances model performance, resulting in significant improvements in language capabilities, primarily due to the increased LLM model size. We leave it to future work how to scale the vision encoder to enhance the visual capabilities and improve model performance on vision recognition and understanding tasks. (ii) Parameter-efficient methods such as LoRA/QLoRA are viable solutions to finetune large-scale LLMs for a good performance-cost trade-off in some real-world settings with limited GPU memory. We observe that LoRA/QLoRA's performance are comparable to that of fine-tuning the full model, establishing their effectiveness through significant cost reduction in both model training and serving. (iii) Our study of training data curation reveals that properly selecting image resolutions and mixing multimodal-language data for model training can significantly improve the performance of the resultant LMM. We also show for the first time that visual instruction tuning can improve LMM's language capability. Note that the training datasets used in this study is small. So, our findings are still preliminary. In future work, we will experiment using much larger datasets to investigate in detail whether and how different methods of training data selection and mixing can improve the quality of much larger LMM.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 3

[2] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023. 3

[3] 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, 2023. 3

[8] https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard

[4] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023. 2

[5] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023. 3

[6] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020. 5

[7] 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. 2

[8] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023. 3

[9] Chunyuan Li. Large multimodal models: Notes on CVPR 2023 tutorial. *arXiv preprint arXiv:2306.14895*, 2023. 1

[10] Chunyuan Li, Zhe Gan, Zhengyuan Yang, Jianwei Yang, Linjie Li, Lijuan Wang, and Jianfeng Gao. Multimodal foundation models: From specialists to general-purpose assistants. *arXiv preprint*, 2023. 1

[11] 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. 3

[12] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 1, 2, 3

[13] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023. 4

[14] OpenAI. Gpt-4 technical report, 2023. 1

[15] 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. 2, 5, 6

[16] Haoqin Tu, Bingchen Zhao, Chen Wei, and Cihang Xie. Sight beyond text: Multi-modal training enhances llms in truthfulness and ethics. *arXiv preprint arXiv:2309.07120*, 2023. 6

[17] Vicuna. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. `https://vicuna.lmsys.org/`, 2023. 2, 5

[18] Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go? exploring the state of instruction tuning on open resources. *arXiv preprint arXiv:2306.04751*, 2023. 6

[19] Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action, 2023. 3

[20] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023. 1, 3, 4

[21] 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. 1, 3