HAMMR 🔨: HierArchical MultiModal React agents for generic VQA

Lluis Castrejon [1] **Thomas Mensink** [1] **Howard Zhou** [1] **Vittorio Ferrari** [2]
Andre Araujo [1] **Jasper Uijlings** [1]

[1] Google DeepMind [2] Synthesia

Abstract

The next generation of Visual Question Answering (VQA) systems should handle a broad range of questions over many VQA benchmarks. Therefore we aim to develop a single system for a varied suite of VQA tasks including counting, spatial reasoning, OCR-based reasoning, visual pointing, external knowledge, and more. In this setting, we demonstrate that naively applying a LLM+tools approach using the combined set of all tools leads to poor results. This motivates us to introduce HAMMR: HierArchical MultiModal React. We start from a multimodal ReAct-based [1] system and make it hierarchical by enabling our HAMMR agents to call upon other specialized agents. This enhances the compositionality, which we show to be critical for obtaining high accuracy. On our generic VQA suite, HAMMR outperforms a naive LLM+tools approach by 16.3% and outperforms the generic standalone PaLI-X VQA model [2] by 5.0%.

1 Introduction

Visual question answering (VQA) (*e.g.* [2, 3, 4, 5, 6, 7]) is a key multimodal and reasoning problem in artificial intelligence. The standard approach for VQA uses Vision+Language Models (VLMs) [8, 2, 9, 10] to generate a textual answer given a question and image. However, an alternative paradigm has recently emerged: combine Large Language Models (LLMs) and computer vision tools to create flexible programs tailored to a given question [5, 6, 11]. The LLM+tools approach can tackle new problems through in-context instructions instead of expensive model finetuning.

VQA problems have been evaluated mostly on individual benchmarks, each with specialized methods for specific question types (*e.g.* [9, 3, 6, 11, 12, 13, 5, 14]). However, it is crucial for real-world systems to handle a broad range of multimodal questions in the wild. Therefore we pose the VQA problem from a unified perspective, where systems have to handle diverse question types. We explore the LLM+tools approach for this generic VQA setting for the first time – in contrast to [5, 6, 11], we want a single method to answer any VQA question. We show that naively applying the LLM+tools approach in this generic setting by using the combined set of all tools leads to poor results.

To address this, we propose **HAMMR** (HierArchical MultiModal React). HAMMR leverages a multimodal ReAct-based [1] system, where LLM agents can be prompted to select the most suitable tools to answer a given question. Departing from previous work [6, 11, 5], our agents may call upon other specialized agents effectively reusing specialized agents as tools. This enhances the compositionality of the LLM+tools approach by enabling reuse of tool combinations that are effective for complex tasks (Fig. 1). Careful experimentation on our generic VQA suite shows that this hierarchical setup is critical for achieving high performance in this generic setting, outperforming naive extensions of existing LLM+tools methods by 16.3%. Additionally, HAMMR achieves state-of-the-art results, outperforming the recent PaLI-X VLM [2] by 5.0%.

An extended version of the paper can be found at https://arxiv.org/abs/2404.05465.

NeurIPS 2024 Workshop on Compositional Learning: Perspectives, Methods, and Paths Forward.

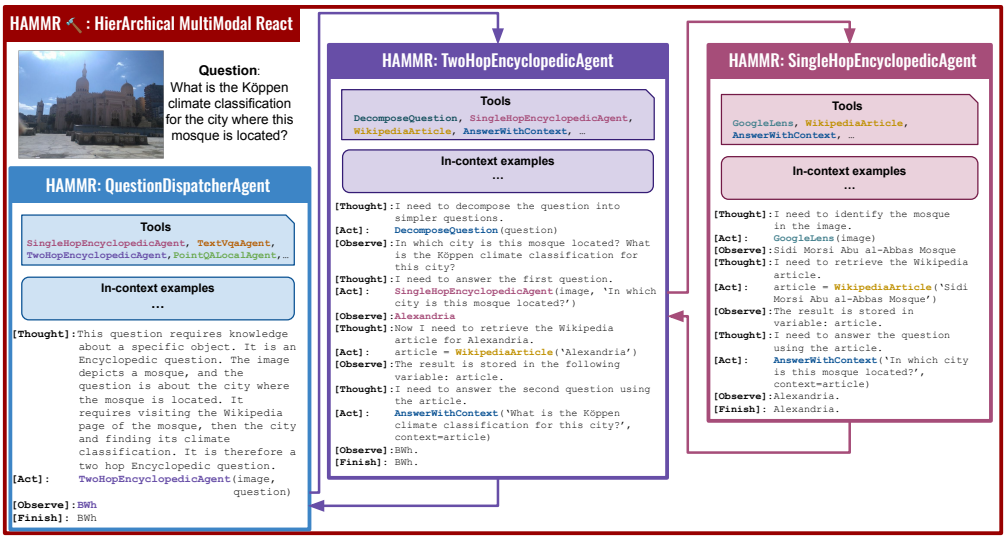

Figure 1: **Example of HAMMR.** We propose HAMMR, an evolution of the LLM+tools approach [6, 11, 5], where a single system can handle a large variety of VQA tasks. Concretely, HAMMR is a multimodal extension of ReAct [1] where agents themselves can act as tools. This results in a hierarchical and highly compositional approach where high-level HAMMR agents call lower-level agents dedicated to more specific tasks.

2 Method

Related work. Several works in VQA have proposed retrieval-augmented models [3, 4, 15, 16, 17, 18]. Complementary, other approaches show that LLMs can solve VQA tasks by translating the image into text which is given as context to the question asked [19, 20, 21, 17]. Building upon these, recent works [6, 5, 11] leverage iterative planning in LLMs [1, 22, 23, 24, 25, 26, 27] in combination with tool-use [28, 29, 30, 31, 32] to solve complex visual tasks.

Multimodal ReAct for generic VQA. To make ReAct multimodal, we give it access to variables, which could contain images, text, or other data types. In the prompt we provide in-context examples of function calls that input and return variables in an `[Act]` step. The returned variable is stored, and its name is mentioned in the `[Observe]` step, so that it is available for further reasoning and actions. The input image is immediately made available as a variable called `image`. For example:

```
[Thought]: I need to crop the top left corner of the image to detect ...
[Act]:     crop = CropImage(image, [0,0,50,50])
[Observe]: Output of 'CropImage' is stored in the variable: 'crop'
```

Now subsequent tools can access the data stored in the variable `crop`. This principle allows tools to return and consume various data types, including bounding-boxes or a whole Wikipedia page.

Tools for VQA. For solving VQA tasks we provide the following tools: fine-grained entity recognition with Google Lens [33], Object Detection using OwlViTv2 [34, 35], OCR, Captioning, and ObjectInImage using either BLIP-2 [36] or PaLI-X [2], image cropping and bounding box tools, retrieving a Wikipedia page, answer a question from context, and decomposing a question (both using LLMs). Finally, analog to [6] we also include VQA as a tool through the provided VLMs.

Naive generic ReAct agent. A ReAct agent for a task with a certain question type is constructed by specifying a prompt with a list of tool descriptions and in-context examples showing how to solve a few questions by using those tools. Hence the straightforward way to make a generic ReAct agent working on a broad set of question types is to create a long prompt listing all tools and all in-context examples across all question types. Such generic ReAct agent is shown in Fig. 2 and acts as our main baseline, representing the most direct extension of current LLM + Tools approaches to generic VQA.

HAMMR: HierArchical MultiModal React. The naive generic ReAct agent has all the necessary information to solve generic VQA. However, as the number of question types increase, it leads to an increasingly long prompt with many reasoning patterns. This makes it difficult for the orchestrator to attend to the relevant parts of the prompt given a new input question. Furthermore, if the orchestrator makes planning mistakes, their cause is hard to identify which makes the system difficult to debug. Therefore we introduce HAMMR: To answer a broad range of question types, we enable agents to

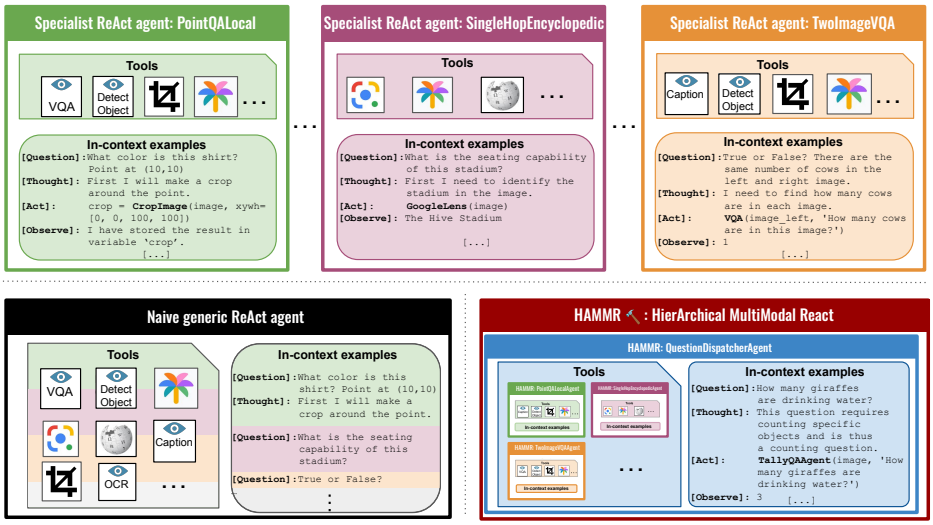

Figure 2: **Our approach**. *Top*: the common non-generalist approach [11, 6, 5] with a specialist agent for each task. *Bottom left*: The naive approach to create a generalist agent is to collect together all tool descriptions and in-context examples of each individual specialist. *Bottom right*: Our HAMMR approach consists of a high-level orchestrator agent capable of calling specialized agents for each required task.

call upon other specialized agents focused on a specific question type – each specialized agent can be reused as a tool. This leads to a compositional approach which enables solving increasingly complex tasks, while limiting the complexity that each individual agent needs to handle. HAMMR modularizes generic reasoning by relying on specialized agents, each requiring a much smaller prompt involving a single reasoning pattern and a small number of tools. This makes the task of each agent in HAMMR simpler, since the solution to the problem is distributed across agents at different reasoning levels.

We start by creating a specialized agent for each VQA question type. Each specialized agent is created via prompts with few-shot examples taken from the training set, which we refine via multiple iterations on a validation set (disjoint from the final test set). Specialized agents may also reuse other specialized agents as part of their reasoning chain, as visualized in Fig. 1.

To tackle generic VQA, we create a high-level dispatcher agent which determines the question type and calls the appropriate specialist HAMMR agent. We create our question dispatcher via prompts, which we refine on a small validation set consisting of a few dozen examples per question type. To quickly iterate we only verify whether the correct specialist agent is identified without actually invoking it, highlighting how the compositionality of HAMMR speeds up development. Fig 2 (bottom-right) visualizes our approach. In Fig. 1 HAMMR solves a question using 3 reasoning levels.

Our design with a high-level dispatcher and specialized agents has multiple benefits over naive generic ReAct: (1) Agents are task-specific, enabling researchers to focus on a single problem at a time; (2) Improving a task-specific agent will improve the overall system, since it will more often solve that task successfully when invoked. In contrast, for generic ReAct progress on one task may mean a regression on another, since the reasoning process for all tasks is entangled. (3) The compositionality of HAMMR means it is easier to debug since failure modes can be attributed to specific agents.

3 Results

Experimental setup. We evaluate on multiple VQA datasets which focus on different question types: PointQA [37] local and look twice questions, Encyclopedic-VQA [17] single hop and two hop questions, NLVR2 [38], TallyQA [39], TextVQA [40] and GQA [41]. We follow [6, 5] and select 1000 random samples from each test set. We use the suggested metric per dataset; exact match (EM) accuracy, VQA accuracy [42], or BERT Matching accuracy [43, 17]. The final metric is the average of the per-dataset accuracies. As the orchestrator LLM we use PaLM 2 [44], publicly available as `text-bison@001` [45]. Most of our experiments use BLIP-2 T5-XXL [36] for asking image questions. When comparing to state-of-the-art VLMs in Sec. 3.2 we use PaLI-X 55B multitask VQA finetuning [2]. We use the publicly available object detector OWL-ViTv2 CLIP L/14 ST+FT [46].

Table 1: **Comparison to generic VLMs.** We compare HAMMR to BLIP-2 [36], Gemini Pro 1.0 [47], and PaLI-X [2]. HAMMR outperforms PaLI-X by 5.0%, showing that it is a strong approach for generic VQA.

Dataset	Specialist React	Generic React	HAMMR-BLIP-2	BLIP-2 [36]	Gemini Pro 1.0 [47]	PaLI-X [2]	HAMMR-PaLI-X
PointQA local	48.1	8.7	47.8	25.3	16.2	48.2	**69.1**
PointQA look twice	55.0	46.1	55.0	54.0	59.3	54.8	**59.5**
EncVQA single hop	51.8	9.8	45.0	15.3	22.1	18.5	**47.8**
EncVQA two hop	25.9	13.9	22.8	14.3	16.6	10.2	**22.8**
NLVR2	61.0	37.2	55.4	52.2	**70.5**	64.8	63.3
GQA	50.7	41.5	50.5	52.9	61.3	**76.4**	72.7
TallyQA	28.6	25.4	29.2	25.1	50.5	**73.2**	72.0
TextVQA	22.0	9.7	16.3	26.3	72.0	**70.1**	49.4
Average performance	42.9	24.0	40.3	33.2	46.1	52.0	**57.0**

3.1 LLM+Tools for generic VQA

Specialist ReAct agents. First we develop and evaluate specialist ReAct agents for each question type and dataset separately, in line with previous LLM+Tools works [11, 6, 5]. These specialist agents serve as an approximate upper-bound for the much harder generic VQA problem, because each specialist is developed and evaluated only on its own specific VQA question type. Tab. 1, left column, shows results. For context, on NLVR2 VisProg [6] reports 62.4%, vs 61.0% for our specialist agents. On GQA we report 50.7% whereas VisProg [6] reports 50.5% and ViperGPT [5] reports 48.1%. While the numbers are not directly comparable due to using different yet approximately equally powerful tools, it shows we have a solid implementation of the LLM+Tools approach.

Naive generic ReAct agent. We build a generic ReAct agent by collecting all tool descriptions and in-context examples from all specialists agents into a single long prompt (Fig. 2 bottom-left). Results over the whole benchmark, Tab. 1 column 2, are poor (24.0%) compared to the approximate upper-bound of specialist agents 42.9%. Manual inspection showed that generic ReAct often makes many reasoning errors by confusing question types while frequently hallucinating non-existent tools.

HAMMR. To address generic VQA using HAMMR, we implement a 'question dispatcher' agent which determines the type of question and dispatches it to the appropriate specialist (Fig 2 bottom-right). Furthermore, HAMMR enables specialist agents to call other specialist agents as in Fig. 1. Results in Tab. 1 show that HAMMR outperforms naive generic ReAct by 16.3% (40.3% vs 24.0%). Hence our hierarchical and compositional approach is superior for generic VQA. Furthermore, HAMMR performs close to the approximate upper-bound of the specialist agents (40.3% vs 42.9%).

3.2 Comparison to SOTA VLMs

We compare HAMMR against several modern generic Vision+Languages models [2, 47, 36] which were trained on a large variety of tasks (including VQA) and report emerging capabilities on tasks not present in their training mix. In particular, we compare to BLIP-2 [36] (FlanT5-XXL), Gemini Pro 1.0 Multimodal [47], and PaLI-X [2] (55B parameter version, finetuned specifically for VQA). For this comparison, we replace BLIP-2 with the more powerful PaLI-X [2] in our tool calls.

Results (Tab. 1, right) reveal that HAMMR outperforms PaLI-X by 5.0% on average across all datasets, demonstrating its strength for generic VQA. PaLI-X only outperforms HAMMR on datasets it was trained on, whereas HAMMR is superior on the others (on NLVR2 they perform comparably). HAMMR performs especially well on encyclopedic questions because it can leverage Google Lens and Wikipedia to access specific information which is hard to memorize for generic VLMs. PaLI-X outperforms Gemini Pro 1.0 likely because we use the VQA-specific PaLI-X [2]. Overall, this experiment shows (1) HAMMR enables easy replacement of tools (i.e. PaLI-X for BLIP-2). (2) HAMMR leverages the best of both worlds: the wealth of implicit knowledge stored in a VLM, and the complementary knowledge that can be accessed by explicitly calling tools.

4 Conclusions

We introduced HAMMR, an evolution of the LLM+tools approach capable of tackling generic VQA. We start from a multimodal ReAct-based system and make it hierarchical by enabling our HAMMR agents to call upon other specialized agents, enhancing the compositionality of the LLM+tools approach. We demonstrate that the hierarchical agent setup is critical for obtaining high accuracy on generic VQA: Using BLIP-2, HAMMR outperforms naive generic ReAct by 16.3%. The improved HAMMR-PaLI-X version outperforms the strong generic PaLI-X VQA model by 5.0%.

References

[1] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing reasoning and acting in language models," in *ICLR*, 2023. [Online]. Available: https://arxiv.org/abs/2210.03629

[2] X. Chen, J. Djolonga, P. Padlewski, B. Mustafa, S. Changpinyo, J. Wu, C. R. Ruiz, S. Goodman, X. Wang, Y. Tay, S. Shakeri, M. Dehghani, D. M. Salz, M. Lucic, M. Tschannen, A. Nagrani, H. Hu, M. Joshi, B. Pang, C. Montgomery, P. Pietrzyk, M. Ritter, A. J. Piergiovanni, M. Minderer, F. Pavetic, A. Waters, G. Li, I. M. Alabdulmohsin, L. Beyer, J. Amelot, K. Lee, A. Steiner, Y. Li, D. Keysers, A. Arnab, Y. Xu, K. Rong, A. Kolesnikov, M. Seyedhosseini, A. Angelova, X. Zhai, N. Houlsby, and R. Soricut, "Pali-x: On scaling up a multilingual vision and language model," *arXiv*, 2023.

[3] L. Gui, B. Wang, Q. Huang, A. Hauptmann, Y. Bisk, and J. Gao, "KAT: A Knowledge Augmented Transformer for Vision-and-Language," in *NAACL*, 2022.

[4] Z. Hu, A. Iscen, C. Sun, Z. Wang, K.-W. Chang, Y. Sun, C. Schmid, D. A. Ross, and A. Fathi, "REVEAL: Retrieval-Augmented Visual-Language Pre-Training with Multi-Source Multimodal Knowledge Memory," in *CVPR*, 2023.

[5] D. Surís, S. Menon, and C. Vondrick, "Vipergpt: Visual inference via python execution for reasoning," in *ICCV*, 2023.

[6] T. Gupta and A. Kembhavi, "Visual programming: Compositional visual reasoning without training," in *CVPR*, June 2023, pp. 14 953–14 962.

[7] J. Wang, Z. Yang, X. Hu, L. Li, K. Lin, Z. Gan, Z. Liu, C. Liu, and L. Wang, "GIT: A generative image-to-text transformer for vision and language," *Transactions on Machine Learning Research*, 2022.

[8] OpenAI, "GPT-4 Technical Report," *arXiv*, vol. 2303.08774, 2023.

[9] X. Chen, X. Wang, S. Changpinyo, A. Piergiovanni, P. Padlewski, D. Salz, S. Goodman, A. Grycner, B. Mustafa, L. Beyer, A. Kolesnikov, J. Puigcerver, N. Ding, K. Rong, H. Akbari, G. Mishra, L. Xue, A. Thapliyal, J. Bradbury, W. Kuo, M. Seyedhosseini, C. Jia, B. K. Ayan, C. Riquelme, A. Steiner, A. Angelova, X. Zhai, N. Houlsby, and R. Soricut, "PaLI: A Jointly-Scaled Multilingual Language-Image Model," *ICLR*, 2023.

[10] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan, "Flamingo: a Visual Language Model for Few-Shot Learning," *NeurIPS*, 2022.

[11] Z. Hu, A. Iscen, C. Sun, K.-W. Chang, Y. Sun, D. A. Ross, C. Schmid, and A. Fathi, "AVIS: Autonomous visual information seeking with large language model agent," in *NeurIPS*, 2023.

[12] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, Y. Choi, and J. Gao, "Oscar: Object-semantics aligned pre-training for vision-language tasks," in *ECCV*, 2020.

[13] J. Lu, D. Batra, D. Parikh, and S. Lee, "ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *NeurIPS*, 2019.

[14] H. Tan and M. Bansal, "LXMERT: Learning cross-modality encoder representations from transformers," in *EMNLP*, 2019.

[15] Y. Lin, Y. Xie, D. Chen, Y. Xu, C. Zhu, and L. Yuan, "REVIVE: Regional visual representation matters in knowledge-based visual question answering," *NeurIPS*, 2022.

[16] K. Marino, X. Chen, D. Parikh, A. Gupta, and M. Rohrbach, "KRISP: Integrating Implicit and Symbolic Knowledge for Open-Domain Knowledge-Based VQA," in *Proc. CVPR*, 2021.

[17] T. Mensink, J. Uijlings, L. Castrejon, A. Goel, F. Cadar, H. Zhou, F. Sha, A. Araujo, and V. Ferrari, "Encyclopedic vqa: Visual questions about detailed properties of fine-grained categories," in *ICCV*, 2023. [Online]. Available: https://arxiv.org/abs/2306.09224

[18] P. Vickers, N. Aletras, E. Monti, and L. Barrault, "In Factuality: Efficient Integration of Relevant Facts for Visual Question Answering," in *ACL*, 2021.

[19] Z. Yang, Z. Gan, J. Wang, X. Hu, Y. Lu, Z. Liu, and L. Wang, "An empirical study of gpt-3 for few-shot knowledge-based vqa," in *AAAI*, 2022.

[20] P. Zhang, X. Li, X. Hu, J. Yang, L. Zhang, L. Wang, Y. Choi, and J. Gao, "Vinvl: Revisiting visual representations in vision-language models," in *CVPR*, 2021.

[21] Y. Hu, H. Hua, Z. Yang, W. Shi, N. A. Smith, and J. Luo, "PromptCap: Prompt-guided task-aware image captioning," *ICCV*, 2023.

[22] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," in *NeurIPS*, 2022.

[23] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le, and E. Chi, "Least-to-most prompting enables complex reasoning in large language models," in *ICLR*, 2023.

[24] M. Besta, N. Blach, A. Kubicek, R. Gerstenberger, L. Gianinazzi, J. Gajda, T. Lehmann, M. Podstawski, H. Niewiadomski, P. Nyczyk, and T. Hoefler, "Graph of thoughts: Solving elaborate problems with large language models," in *arXiv*, 2023.

[25] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan, "Tree of thoughts: Deliberate problem solving with large language models," in *arXiv*, 2023.

[26] S. Hao, Y. Gu, H. Ma, J. J. Hong, Z. Wang, D. Z. Wang, and Z. Hu, "Reasoning with language model is planning with world model," in *EMNLP*, 2023.

[27] Y. Xie, K. Kawaguchi, Y. Zhao, X. Zhao, M.-Y. Kan, J. He, and Q. Xie, "Self-evaluation guided beam search for reasoning," in *NeurIPS*, 2023.

[28] S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Millican, G. van den Driessche, J.-B. Lespiau, B. Damoc, A. Clark, D. de Las Casas, A. Guy, J. Menick, R. Ring, T. Hennigan, S. Huang, L. Maggiore, C. Jones, A. Cassirer, A. Brock, M. Paganini, G. Irving, O. Vinyals, S. Osindero, K. Simonyan, J. W. Rae, E. Elsen, and L. Sifre, "Improving language models by retrieving from trillions of tokens," *ICML*, 2022.

[29] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: Retrieval-Augmented Language Model Pre-Training," in *Proc. ICML*, 2020.

[30] O. Khattab, K. Santhanam, X. L. Li, D. Hall, P. Liang, C. Potts, and M. Zaharia, "Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp," in *arXiv*, 2023.

[31] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language models can teach themselves to use tools," in *arXiv*, 2023.

[32] A. Parisi, Y. Zhao, and N. Fiedel, "Talm: Tool augmented language models," in *arXiv*, 2023.

[33] "Google Lens," https://lens.google.com - Web interface available at https://images.google.com.

[34] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, X. Wang, X. Zhai, T. Kipf, and N. Houlsby, "Simple open-vocabulary object detection with vision transformers," *ECCV*, 2022.

[35] M. Minderer, A. Gritsenko, and N. Houlsby, "Scaling open-vocabulary object detection," *arXiv preprint arXiv:2306.09683*, 2023.

[36] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models," in *ICML*, 2023.

[37] A. Mani, N. Yoo, W. Hinthorn, and O. Russakovsky, "Point and ask: Incorporating pointing into visual question answering," arXiv, Tech. Rep., 2020.

[38] A. Suhr, S. Zhou, I. Zhang, H. Bai, and Y. Artzi, "A corpus for reasoning about natural language grounded in photographs," in *ACL*, 2019.

[39] M. Acharya, K. Kafle, and C. Kanan, "Tallyqa: Answering complex counting questions," in *AAAI*, 2019.

[40] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach, "Towards vqa models that can read," in *CVPR*, June 2019.

[41] D. A. Hudson and C. D. Manning, "GQA: A new dataset for real-world visual reasoning and compositional question answering," in *CVPR*, 2019.

[42] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: visual question answering," in *ICCV*, 2015.

[43] J. Bulian, C. Buck, W. Gajewski, B. Boerschinger, and T. Schuster, "Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation," *arXiv preprint arXiv:2202.07654*, 2022.

[44] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel, "PaLM: Scaling Language Modeling with Pathways," *arXiv*, 2022.

[45] "PaLM API," https://cloud.google.com/vertex-ai/docs/generative-ai/learn/models.

[46] N. H. Matthias Minderer, Alexey Gritsenko, "Scaling open-vocabulary object detection," *NeurIPS*, 2023.

[47] Gemini Team Google, "Gemini: A family of highly capable multimodal models," in *arXiv*, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: Yes

   Justification: All the claims made in these sections are taken from the Results section.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: No

   Justification: We only briefly discuss limitations of our method due to space.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: NA.

   Justification: The paper does not include theoretical results.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: Yes.

   Justification: All reproducibility details (except those only found in code) are described in the paper.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: No.

   Justification: We do not provide code for the experimental results at the moment.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: Yes.

   Justification: We describe the exact models and setup used in our experiments.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: No.

   Justification: Due to the expensive cost of running experiments, we did not include confidence intervals for the experiments.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: No.

   Justification: We describe the models used, but we do not go into the details of the required resources.

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: Yes.

   Justification: We adhere to the code of ethics.

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: No.

    Justification: Our work has no direct negative societal impact, it only provides an improvement over previous LLM + Tools VQA approaches.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: NA.

    Justification: We do not release data or models with high risk of misuse.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: Yes.

    Justification: We mention and cite all the datasets used.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: NA.

    Justification: We do not release new assets.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: NA.

    Justification: No crowdsourcing or research with human subjects.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: NA.

    Justification: No crowdsourcing or research with human subjects.