Episode-Level Multimodal KV Caching for Embodied Question Answering

Hyobin Ong[1,2], **Minsu Jang**[1,2†]
[1]University of Science and Technology (UST), South Korea
[2]Electronics and Telecommunications Research Institute (ETRI), South Korea

Abstract

Embodied Question Answering (EQA) requires agents to sustain a representation of the world while answering multi-turn queries in real time. A key challenge is how to maintain and update this world model efficiently under resource constraints. Existing approaches repeatedly re-encode visual inputs or apply retrieval-augmented generation, both of which introduce latency that limits interactive use. We propose an episode-level multimodal KV cache that is constructed once from uniformly sampled frames and reused across all queries in the same episode. This cache serves as a lightweight multimodal memory that reduces redundant computation while preserving relevant context. On the openEQA benchmark, our method achieves up to an 82% reduction in total question-answering time compared to naïve multi-image inference, with only a modest drop in accuracy. These findings demonstrate that reusing an episode-level cache provides an effective mechanism for maintaining and updating world models to achieve efficient reasoning in EQA.

1 Introduction

Embodied Question Answering (EQA) requires an agent to explore and understand its environment and then respond immediately to multi-turn queries [1, 2]. From the perspective of human–robot interaction, reducing response latency is therefore a critical challenge [3]. In particular, within a single indoor episode, EQA often presents a sequence of contextually related questions. Re-encoding the same or similar frames at every turn leads to an accumulation of computation and memory costs, which in turn causes latency to increase significantly. As a result, optimizing only the sequence length at the scene level, as is common in VideoQA, is insufficient. For real-time EQA, it is essential to maintain and update a multimodal KV cache at the episode level in order to secure inference efficiency.

Research on multimodal KV caching for large multimodal language models (MLLMs) is advancing rapidly. ReKV [4] enables real-time answers by reloading past visual context through in-context KV-cache retrieval. LOOK-M [5] proposes *look-once* cache compression based on the observation that text dominates during the prefill stage. MEDA [6] uses cross-modal attention entropy to allocate caches dynamically by layer, cutting memory and speeding decoding. VL-Cache [7] exploits modality- and layer-level sparsity to keep accuracy with smaller caches while greatly accelerating inference. SCBench [8] analyzes the full cache pipeline—creation, compression, retrieval, loading—and pinpoints limits in multi-turn, multi-request settings. In robotics and EQA, ReMEmbR [9] introduces long-term spatiotemporal memory, and MemoryEQA [10] uses a hierarchical memory pipeline across exploration, planning, and answering. While many prior works have focused on optimizing streaming video or generic long-context multimodal inference, this work is distinguished by constructing and reusing an episode-level prefix multimodal kv cache for EQA with multi-turn interactions, thereby eliminating redundant visual re-encoding for each question.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: The 1st Workshop on Efficient Reasoning.

Retrieval-Augmented Generation (RAG) [11] which performs separate retrieval and encoding for each question has the advantage of improving accuracy, but it also increases inference latency. However, in tasks such as EQA, where real time interaction is required, this latency becomes a critical bottleneck. Naïve multi-image inference repeatedly processes all frames and RAG based methods repeatedly perform retrieval and encoding for each query which is inefficient. Therefore, this study explores whether maintaining and reusing a multimodal cache at the episode level can improve the balance between latency and accuracy.

2 Method

We propose a method for efficient question answering in a setting where questions are posed sequentially within a single episode.

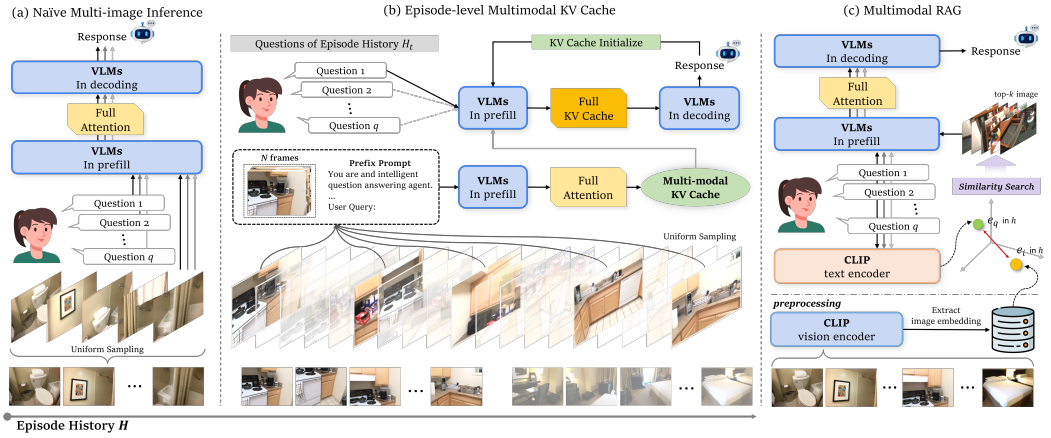

Figure 1: Comparison of three inference methods for Embodied Question Answering: (a) naïve multi-image inference that re-encodes all frames for every question, (b) the proposed episode-level multimodal KV cache that initializes once and is reused across all queries, and (c) multimodal RAG that retrieves relevant frames per question using CLIP encoders.

2.1 Problem Definition

We denote by $\mathcal{H} = \{1, ..., N\}$ the episode history composed of several distinct indoor environments. Each episode memory $h \in \mathcal{H}$ is an RGB–D frame sequence ordered in time, $F_h = \{I_{h,t}\}_{t=1}^{T_h}$, where T_h is the total number of frames in episode memory h. For each episode memory we have multi-turn QA pairs, $Q_h = \{q_{h,j}\}_{j=1}^{M_h}$ and $A_h^= \{a^*_{h,j}\}_{j=1}^{M_h}$, with M_h the number of QA pairs. Given the image and text context $X_{h,j}$ for question $q_{h,j}$, a Vision–Language Model (VLM) generates $\hat{a}_{h,j} \sim p_\theta(\cdot \mid X_{h,j})$.

2.2 Proposed method

VLM prefill Given an episode memory h with the time-ordered frame sequence F_h, we uniformly sample n frames $\widetilde{F}_h = \{I_{h,t}\}_{t \in S_h}$ from F_h, where $S_h \subseteq \{1, \ldots, T_h\}$ and $|S_h| = n$. We input to the VLM the sequence of m tokens $\{x_1, .., x_m\}$ that includes all visual and language tokens composed of the selected frames $\widetilde{F}_h$ and the fixed prefix text p_0. The VLM performs a forward pass with parallel self-attention over the tokens and, at the same time, computes the key vectors $\{k_1^{(l)}, ..., k_m^{(l)}\}$ and the value vectors $\{v_1^{(l)}, ..., v_m^{(l)}\}$ at each transformer layer l, carrying out the KV cache C_h in GPU memory:

$$C_h \sim p_{\text{prefill},\theta}\big(\cdot \,\big|\, \widetilde{F}_h, p_0\big). \tag{1}$$

VLM decode When a question $q_{h,j}$ within episode memory h is given, the model inputs $q_{h,j}$ and the cached C_h from the prefill stage as past key/values. The model computes the keys/values of the question tokens at each transformer layer l, appends them to the cache C_h, and performs auto-regressive decoding conditioned on that cache to generate one token at a time:

$$\hat{a}_{h,j} \sim p_{\text{decode},\theta}\big(\cdot \,\big|\, C_h, q_{h,j}\big). \tag{2}$$

Since the keys/values of the generated answer tokens are also accumulated in the cache, after the turn ends the cache is reset to the initially fixed prefix length so that it does not increase unnecessarily.

3 Experiments

Baselines To compare with the proposed multimodal cache in Fig. 1b, we establish two baselines corresponding to Fig. 1a and Fig. 1c. **(i) Naïve** (Fig. 1a): For each question the prefix the question text and n frames are input to the VLM and the answer is generated with full attention. The same or similar visual context is re-encoded at every turn so no cache is reused and each question is inferred independently. **(ii) Multimodal RAG** (Fig. 1c): Implemented as a two stage pipeline. In the offline stage frame embeddings are extracted and stored for all episode history using the CLIP vision encoder. In the online stage when a question $q_{h,j}$ of episode h is given a question embedding is computed using the CLIP text encoder and similarity search is performed over the frame embeddings of episode h to retrieve the top-k frames. The selected frames are input to the VLM together with the question to generate an answer. Retrieval and encoding are repeated for each question and no cache reuse is performed.

Evaluation To evaluate accuracy on the openEQA benchmark [2], we use the **LLM-Match**, whose robustness has been validated in prior work. Given the question q, the ground-truth answer a*, and the model's prediction $\hat{a}$, an LLM assigns a correctness score $\sigma_i \in \{1, 2, 3, 4, 5\}$. The aggregated LLM-Match is defined as:

$$LLM\text{-}Match = \frac{1}{N} \sum_{i}^{N} \frac{\sigma_i - 1}{4} \times 100\%. \tag{3}$$

Additionally, on the openEQA benchmark we evaluate latency from cache use with three metrics. **(i) Episode-level KV cache construction time (KV cache time)**: the time to generate and cache the multimodal prefix during the prefill stage. **(ii) Answer generation time (Decoding time)**: the elapsed time for image processing and generating the answer to the question, excluding RAG retrieval and the construction and initialization of the KV cache. **(iii) Total question answering time (Total inference time)**: the end-to-end time to process all 1,636 questions, including every runtime operation such as retrieval, cache construction, and decoding.

Experiment Setup We evaluate on the openEQA benchmark using the indoor datasets HM3D [12] and ScanNet [13]. The frame resolutions are 1920×1080 and 1296×968. To balance the cost of multi-image inference, VLM inputs are resized to between 256 and 384 pixels. We use *Qwen/Qwen2.5-VL-7B-Instruct* [14], which supports multi-image inference, and the multimodal reasoning model *Fancy-MLLM/R1-Onevision-7B* [15]. All model parameters are frozen. We conducted experiments with two maximum generation lengths, 128 tokens and 4096 tokens for the reasoning model. All experiments run on a single NVIDIA A6000 GPU with SDPA attention. For Multimodal RAG, we use *openai/clip-vit-base-patch32* to extract image and text embeddings of the same dimensionality and perform cosine similarity search. For evaluation, we used GPT-4 [16] for LLM-Match.

Results Table 1 summarizes accuracy and latency by the number of input frames, the model, and the inference method. The key finding is that the KV cache greatly shortens decoding time. We encode the images and prefix text once at the start of each episode and store them in the KV cache. After that, for each question we encode only the newly generated tokens while reusing the cache. In contrast, the naïve and RAG baselines must rerun full attention over the images, the prefix text, and the question at every turn, which leads to similar decoding times for both. As a result, the KV cache removes redundant computation and significantly reduces episode-level multi-turn latency, which is especially helpful for real-time EQA.

We conducted additional experiments to test whether the method is effective for multimodal reasoning models. The results show that the multimodal KV cache consistently reduces decoding latency not only for standard VLMs but also for multimodal reasoning models. However, despite using a reasoning model, LLM-Match scores were lower than those of a standard VLM. Prior work [17] suggests that visual reasoning can help at very large scales but may hurt performance in smaller models. Overall, for real-time EQA, balancing inference speed and accuracy currently favors using a standard VLM.

In our experiments, the episode-level multimodal KV cache greatly reduced decoding latency, but its LLM-Match accuracy was generally lower than the naïve baseline and the multimodal RAG approach.

Model	# Frames	Method	Total infer. time ↓ (hh:mm:ss)	KV cache time (s)	Decoding time (s) ↓	LLM-Match ↑	
						HM3D	ScanNet
max_new_tokens=128							
Qwen2.5-VL [14]	5	Naïve	00:33:16	-	1.2	40.2	**52.3**
		Multimodal KV Cache	**00:10:41**	0.9	**0.3**	37.6	49.8
		Multimodal RAG	00:33:40	-	1.2	**40.8**	**52.3**
	10	Naïve	01:03:20	-	2.3	43.0	**57.7**
		Multimodal KV Cache	**00:13:22**	1.7	**0.3**	38.5	55.7
		Multimodal RAG	01:03:35	-	2.3	**43.3**	53.6
	20	Naïve	02:01:59	-	4.5	44.5	**59.7**
		Multimodal KV Cache	**00:21:22**	3.4	**0.4**	43.4	57.3
		Multimodal RAG	02:02:47	-	4.5	**47.1**	56.5
max_new_tokens=4096							
Qwen2.5-VL [14]	5	Naïve	01:36:09	-	3.5	**40.0**	**53.2**
		Multimodal KV Cache	**00:17:19**	3.1	**0.3**	37.7	49.9
		Multimodal RAG	01:36:21	-	3.5	39.8	**53.2**
R1-Onevision [15]	5	Naïve	06:53:00	-	15.1	27.9	40.8
		Multimodal KV Cache	**05:37:31**	3.2	**12.1**	28.4	40.4
		Multimodal RAG	06:37:37	-	14.5	**36.3**	**42.6**

Table 1: Comparison of *Qwen2.5-VL-7B* and *R1-OneVision-7B* on the openEQA benchmark under different numbers of frames. We report accuracy (LLM-Match) on HM3D and ScanNet, KV cache initialization time, decoding time per question, and total inference time. Both models are evaluated with three inference strategies: naïve multi-image inference, multimodal KV cache, and multimodal RAG.

The main reason is that the current cache uses a uniform design that ignores differences in attention density and importance across layers and modalities. When dense layers receive too little cache budget, critical cues are truncated [6]. When sparse layers receive too much, resources are wasted without accuracy gains. This issue is especially pronounced in long multimodal contexts. Because our method builds a fixed prefix at the start of the episode and reuses it at every turn, the lack of layer-wise allocation can more easily lead to accuracy drops.

4 Limitation & Discussion

The proposed episode-level multimodal KV cache markedly reduces inference latency, but its accuracy is lower than that of the naïve and RAG baselines. This drop stems from a uniform cache design that ignores heterogeneity in attention density across layers and modalities. Dense layers suffer information loss under a tight cache budget, whereas sparse layers waste resources without benefit. Reusing a fixed prefix for all queries within an episode also fails to capture question-specific context. Our study advances a new perspective on real-time EQA by maintaining an episode-level cache. Experiments show that naïve and RAG methods still lead in accuracy, while the KV cache delivers a decisive advantage in latency, which is critical in Human–Robot Interaction scenarios. The findings illustrate a clear latency–accuracy trade-off. Naïve inference avoids information loss but is compute-heavy. RAG gains efficiency through selective retrieval, yet retrieval costs accumulate. The KV cache achieves sharp latency reductions with a simple design, but its information retention can be improved. Future work will explore dynamic, modality-aware cache allocation and query-conditioned refresh strategies, drawing on ideas from MEDA [6] and VL-Cache [7], to push the trade-off toward better accuracy without sacrificing efficiency.

5 Conclusion

We propose an episode-level multimodal KV cache to address inference latency in multi-turn EQA. The cache is built once at the start of each episode by uniform sampling and reused for questions, which greatly speeds inference compared with naïve multi-image inference and RAG-based frame retrieval. On the openEQA benchmark, this approach reduces latency by up to 82%. Although accuracy declines slightly, the results show that maintaining and updating an episode-level multimodal KV cache is essential for efficient EQA, which is especially important for real-time HRI. Future work will apply layer-wise dynamic cache allocation and question-conditioned cache updates to better balance efficiency and accuracy.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2022-II220951, Development of Uncertainty-Aware Agents Learning by Asking Questions, 50%, No. RS-2024-00336738, Development of Complex Task Planning Technologies for Autonomous Agents, 25%, No.GTL25041-000, by the National Research Council of Science & Technology (NST), 25%).

References

[1] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–10, 2018.

[2] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit Sharma, Vincent Berges, Shiqi Zhang, Pulkit Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Sasha Sax, and Aravind Rajeswaran. Openeqa: Embodied question answering in the era of foundation models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[3] Zhonghao Shi, Enyu Zhao, Nathaniel Dennler, Jingzhen Wang, Xinyang Xu, Kaleen Shrestha, Mengxue Fu, Daniel Seita, and Maja Matarić. Hribench: Benchmarking vision-language models for real-time human perception in human-robot interaction. *arXiv preprint arXiv:2506.20566*, 2025.

[4] Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Tao Zhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video kv-cache retrieval, 2025.

[5] Zhongwei Wan, Ziang Wu, Che Liu, Jinfa Huang, Zhihong Zhu, Peng Jin, Longyue Wang, and Li Yuan. Look-m: Look-once optimization in kv cache for efficient multimodal long-context inference, 2024.

[6] Zhongwei Wan, Hui Shen, Xin Wang, Che Liu, Zheda Mai, and Mi Zhang. Meda: Dynamic kv cache allocation for efficient multimodal long-context inference. *arXiv preprint arXiv:2502.17599*, 2025.

[7] Dezhan Tu, Danylo Vashchilenko, Yuzhe Lu, and Panpan Xu. Vl-cache: Sparsity and modality-aware kv cache compression for vision-language model inference acceleration, 2024.

[8] Yucheng Li, Huiqiang Jiang, Qianhui Wu, Xufang Luo, Surin Ahn, Chengruidong Zhang, Amir H. Abdi, Dongsheng Li, Jianfeng Gao, Yuqing Yang, and Lili Qiu. Scbench: A kv cache-centric analysis of long-context methods, 2025.

[9] Abrar Anwar, John Welsh, Joydeep Biswas, Soha Pouya, and Yan Chang. Remembr: Building and reasoning over long-horizon spatio-temporal memory for robot navigation, 2024.

[10] Mingliang Zhai, Zhi Gao, Yuwei Wu, and Yunde Jia. Memory-centric embodied question answer, 2025.

[11] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[12] Santhosh K. Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai, 2021.

[13] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[14] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025.

[15] Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, Bo Zhang, and Wei Chen. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization, 2025.

[16] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.

[17] Aishik Nagar, Shantanu Jaiswal, and Cheston Tan. Zero-shot visual reasoning by vision-language models: Benchmarking and analysis, 2024.