# Multi-Agent VQA: Exploring Multi-Agent Foundation Models in Zero-Shot Visual Question Answering

Bowen Jiang, Zhijun Zhuang, Shreyas S. Shivakumar, Dan Roth, Camillo J. Taylor
GRASP Lab, University of Pennsylvania
Philadelphia, PA, 19104, USA
{bwjiang, zhijunz, sshreyas, danroth, cjtaylor}@seas.upenn.edu *

## Abstract

*This work explores the zero-shot capabilities of foundation models in Visual Question Answering (VQA) tasks. We propose an adaptive multi-agent system, named Multi-Agent VQA, to overcome the limitations of foundation models in object detection and counting by using specialized agents as tools. Unlike existing approaches, our study focuses on the system's performance without fine-tuning it on specific VQA datasets, making it more practical and robust in the open world. We present preliminary experimental results under zero-shot scenarios and highlight some failure cases, offering new directions for future research.*

## 1. Introduction

Recently we have witnessed a rapid emergence of multi-modal foundation models [2, 8, 9], that seamlessly bridge vision and language understanding tasks. This fusion allows vision systems to leverage the versatility of natural language, thereby extending their understanding and reasoning capabilities to an unprecedented level.

Visual Question Answering (VQA) [4] serves as a suitable problem in testing foundation models on complicated vision-language understanding. Despite their popularity, the zero-shot capabilities of foundation models in the domain of VQA remain largely unexplored. Almost all pretrained large vision-language models (LVLM) in the VQA literature require fine-tuning on specified VQA datasets with a very limited vocabulary to achieve state-of-the-art performance [3, 12, 13]. This practice, while effective, overlooks the actual potential of foundation models and limits their open-world usage beyond their fine-tuned datasets. The generalization ability of foundation models already demonstrated in many multi-modal tasks suggests that an exploration of their zero-shot VQA performance would unveil new dimensions of their abilities.

*Codes and detailed prompts are available at https://github.com/bowen-upenn/Multi-Agent-VQA

This work uncovers some of the challenges these models might face. While foundation models are typically pretrained with images and corresponding textual descriptions, they may not have been specifically pretrained to interpret underlying graphical structures in images. As a result, they often fall short in VQA when the question details specific object attributes and relationships that form local scene graphs [16], or counting the number of objects in images.

Tools [11] are specialized agents in their own fields. Instead of fine-tuning the foundation models to overcome their limitations, we harness specialized models for object detection and counting as tools within our multi-agent system [15]. Specifically, when an LVLM fails to detect an object in the scene, an object detector like Grounded Segment Everything [10] can help. Likewise, for questions that require counting objects, a model specifically trained for the counting task [6] can fulfill this role. This collaborative framework enables a more flexible response to the diverse challenges in VQA tasks, effectively exploiting the full capability of foundation models without additional training.

## 2. Methods

This section describes the pipeline of our adaptive Multi-Agent VQA system, shown in Figure 1. Different agents guide the system to analyze shortcomings, fill in missing information, and discover the final answer step-by-step. We use GPT-4V [8] as our LVLM and GPT-3.5 [9] as our LLM, but the system can be adapted to other foundation models.

### 2.1. Initial attempt

We introduce an adaptive pipeline that allows an LVLM to answer a given question directly, which can optimize the average inference time. We carefully craft prompts to introduce the problem and guide the LVLM to avoid overconfidence. If the LVLM thinks it cannot produce the answer because it has missed key objects in the image, it is instructed to say so explicitly using the following special tokens "[Answer Failed]". Otherwise, the system will bypass the multi-agent modules to avoid unecessary additional computation.

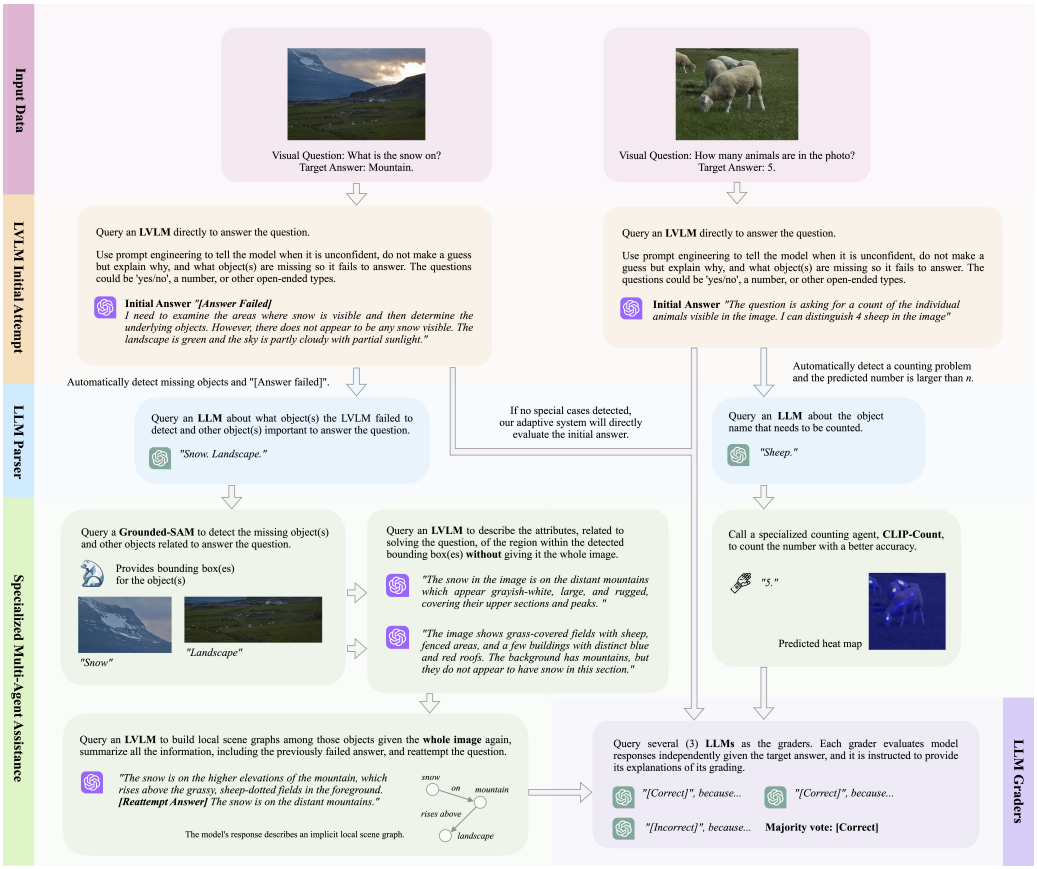


Figure 1. Overview of the adaptive Multi-Agent VQA System. The process begins with an LVLM attempting to answer a visual question directly. An LLM parsing agent automatically detects challenging cases and calls specialized agents, including object-detection and counting models as tools. The LVLM would reattempt the question, with the response assessed by LLM-based graders for a majority vote.

## 2.2. Reattempt by adaptively calling multi-agents

### 2.2.1 LLM parsing agent

An LLM-based parsing agent aims to automatically detect special cases in the initial response and select appropriate agents for assistance. Specifically, whenever it detects the special tokens "[Answer Failed]", it will extract the object names that the LVLM thought were missing. We also realize that LVLMs may struggle with counting objects, especially when the number $n$ is large. The LLM parsing agent will detect whether the question refers to a counting problem and extract the named objects that need to be counted.

### 2.2.2 Object detection agent

Following the identification of missing objects, our system employs a pretrained object-detection agent to localize them in the visual scene. We rely on the Grounded Segment Everything model [10] as the tool, which takes object names as inputs and outputs masks and bounding boxes. Unlike general LVLMs, it is specifically trained in object detection tasks is better at recognizing and localizing small or non-obvious objects that the initial LVLM might overlook.

### 2.2.3 LVLM object description agent

This step bridges the object-detection agent and the original LVLM, focusing on the detailed analysis of important objects detected in the previous step. We crop the original image based on the predicted bounding boxes and feed each region to a separate instance of the LVLM in parallel. These LVLMs have identical architecture and weights, tasked with describing attributes related to answering the question for each object. For example, one LVLM object-description agent describes the snow in Figure 1 as covering a *"grayish-white distant mountain"*, pointing out the key information that can finally solve the question *"what is the snow on"* successfully. By narrowing each LVLM's field of view to just the cropped region within the bounding box, we eliminate the complexity of scanning the entire visual scene to localize relevant objects and their visual details.

### 2.2.4 LVLM reattempts the question

Finally, we revisit the original LVLM by giving it a comprehensive set of inputs: the original image and question, the initially unsuccessful attempt and why it failed, and the de-

scription of the detected objects that were previously missing. The model now is expected to have enough context to construct an implicit local scene graph expressed in natural language, figuring out relationships among these objects that are relevant to answering the question. Take the example in Figure 1 again; the model would now understand that the snow is only located on the mountain but not on the broader landscape. Such an implicit graphical structure allows the LVLM to weave together the various pieces of information into a whole and reattempt the visual question to provide a more accurate and comprehensive response.

#### 2.2.5 Object counting agent

In scenarios where the LLM parsing agent recognizes a counting problem and the initially predicted number $n$ is greater than 3, the system will call a specialized counting agent, CLIP-Count [6], for help. Guided by text prompts, CLIP-Count generates a density map for open-vocabulary objects in a zero-shot manner and achieves better counting accuracy. Its answer will serve as the final answer.

### 2.3. LLM answer grading agent

Unlike existing approaches that fine-tune models on given datasets [3, 12, 13], we no longer expect the model to replicate exact dataset annotations [4]. By embracing zero-shot learning, we avoid training foundation models to fit a limited vocabulary and the inevitable human annotation bias. To this end, we introduce LLM-based graders as a novel component of our framework for open-ended evaluation, mimicking humans that allow diverse phrasing and additional information not mentioned in the annotation. We collect assessments from three individual grading agents and take the majority vote to provide a more robust evaluation.

## 3. Experiments

### 3.1. Datasets

We evaluate our method on the widely adopted VQA-v2 [4] and GQA [5] datasets. Due to the costs and time requirements of GPT-4V API [8], we have to use a subset of the data to evaluate the performance - a typical drawback in large foundation models. For GQA, we take the same 1000 validation samples used in [16] for testing. VQA-v2 comprises "yes/no", "number", and other question types. However, its test set is not publicly available and requires exact matches of the answers, making our LLM-based graders inapplicable. We instead adopt the VQA-v2 rest-val dataset, the validation dataset in [1, 13] that was never used for training. It contains 5228 unique image-question pairs.

### 3.2. Results and Zero-Shot VQA benchmark

We split Table 4 into fine-tuned and zero-shot sections for fair comparisons. We run BEiT-3 [13] and VLMo [1], representing the current state of the art. When the BEiT-3 [13] or VLMo model [1] fine-tuned on VQA-v2 training dataset is evaluated on VQA-v2 rest-val, namely *BEiT3-large-vqa-v2* or *VLMo-large-vqa-v2* in Table 4, it keeps its advantage.

However, versions of BEiT-3 or VLMo that have *not* been fine-tuned on VQA-v2, namely *BEiT3-large-indomain* or *VLMo-large-coco* in Table 4, achieve almost zero accuracies on VQA-v2. Table 2 also shows significant drops in their performance on GQA in zero-shot, a dataset different from the one, namely VQA-v2, they are fine-tuned on. **These declines highlight a significant limitation of existing VQA models:** Despite their advancements, models like BEiT-3 and VLMo depend heavily on dataset-specific fine-tuning with a low zero-shot generalization ability. Most existing works share similar designs [3, 12, 13], underscoring the unique value of our zero-shot solution.

Table 1. Results on VQA-v2 rest-val [4] dataset.

| Weights | Methods | Accuracy |
|---|---|---|
| Fine-tuned | BEiT3-large-vqa-v2 [13] | **88.33** |
| | VLMo-large-vqa-v2 [1] | 83.36 |
| Zero-shot | BEiT3-large-indomain [13] | 0.01 |
| | VLMo-large-coco [1] | 0.00 |
| | Multi-Agent VQA (ours) | **78.02** |

Table 2. Results on GQA-val [5, 16] subset.

| Weights | Methods | Accuracy |
|---|---|---|
| Zero-shot | BEiT3-large-vqa-v2 [13] | 64.67 |
| | VLMo-large-vqa-v2 [1] | 0.00 |
| | BLIP2-flan-t5-xl [7] | 50.40 |
| | LessIsMore-local [16] | 58.30 |
| | Multi-Agent VQA (ours) | **79.70** |

### 3.3. Ablation study

W assess the impact of removing detailed chain-of-thought (CoT) reasoning [14], the CLIP-Count object-counting agent [6], and the multi-agent pipeline in Table 3. Keeping only basic prompt instructions leads to reduced performance in all question types. The removal of CLIP-Count forces the LVLM to answer all counting questions itself, resulting in an accuracy drop on this type of question, labeled as "num" in Table 3. The most profound impact happens by removing the proposed multi-agent pipeline, where the system has to rely solely on a single LVLM for VQA tasks, here the accuracy decreases by nearly 10 percent.

### 3.4. Limitations and failure examples

The performance in object counting tasks is bounded by the counting agent which does not always work well, and there are currently few methods that support zero-shot object counting [6]. The reliance on API calls for LLM and LVLM agents constrains the inference speed of our model and the outputs are not entirely deterministic.

Table 3. Ablation study

| Dataset | Ablation | Acc (yes/no, num, other) |
|---|---|---|
| VQA-v2 | w/o detailed CoT [14] | 74.81 (81.20, 57.01, 74.54) |
| | w/o CLIP-Count [6] | 76.47 (84.26, 52.81, 76.50) |
| | w/o multi-agent | 69.20 (79.05, 53.43, 65.21) |
| | final | **78.02** (**84.82**, **60.63**, **77.83**) |
| GQA | w/o multi-agent | 68.50 |
| | final | **79.70** |

Failure cases often happen when the foundation model overthinks the questions and provides more exhaustive answers than necessary. Take Figure 2 as an example; the model accurately deduces the presence of a shore on the opposite side but unnecessarily tries to figure out if the shore is sandy and, therefore, a beach. It also calls the object-detection agent to spot umbrellas, a typical sign of a beach, but it wastes time in analyzing the umbrella on the current side of the shore.

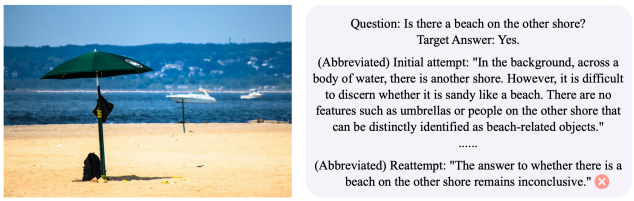


Figure 2. An example of the failure case.

Table 4. Results on VQA-v2 rest-val [4] dataset.

| Weights | Methods | Accuracy |
|---|---|---|
| Fine-tuned | BEiT3 fine-tuned on VQA-v2 | **88.33** |
| | VLMo fine-tuned on VQA-v2 | 83.36 |
| Zero-shot | BEiT3 pretrained only | 0.01 |
| | VLMo pretrained only | 0.00 |
| | BEiT3 fine-tuned on GQA instead | 64.67 |
| | VLMo fine-tuned on GQA instead | 0.00 |
| | Ours w/o multi-agents | 69.20 |
| | Ours | **78.02** |

## 4. Future work

Our findings represent preliminary work on the zero-shot VQA capabilities of foundation models. In the near future, we will employ different foundation models and specialized tools, discuss detailed prompt engineering and chain-of-thought reasoning, and present a more comprehensive zero-shot VQA benchmark in the open world.

## References


[1] Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *Advances in Neural Information Processing Systems*, 35:32897–32912, 2022. 3

[2] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 1

[3] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022. 1, 3

[4] 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. 1, 3, 4

[5] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019. 3

[6] Ruixiang Jiang, Lingbo Liu, and Changwen Chen. Clip-count: Towards text-guided zero-shot object counting. *arXiv preprint arXiv:2305.07304*, 2023. 1, 3, 4

[7] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 3

[8] OpenAI. Gpt-4v(ision) system card. *cdn.openai.com*, 2023. 1, 3

[9] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1

[10] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024. 1, 2

[11] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[12] Peng Wang, Shijie Wang, Junyang Lin, Shuai Bai, Xiaohuan Zhou, Jingren Zhou, Xinggang Wang, and Chang Zhou. One-peace: Exploring one general representation model toward unlimited modalities. *arXiv preprint arXiv:2305.11172*, 2023. 1, 3

[13] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022. 1, 3

[14] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022. 3, 4

[15] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 2023. 1

[16] Shu Zhao and Huijuan Xu. Less is more: Toward zero-shot local scene graph generation via foundation models. *arXiv preprint arXiv:2310.01356*, 2023. 1, 3