

Supplementary Materials: Few-shot Multimodal Explanation for Visual Question Answering

Anonymous Authors

Table 5: 12 simplified atomic operations of reasoning steps.

Operation	Semantic
Select	Select a specific class of objects
Relate	Relate objects by their relations
Query	Query a specific attribution of an object
Exist	Check the existence of an object
Filter	Filter objects by a specific attribution
Verify	Check if the object has a specific attribution
Common	Search the common attribution of multiple objects
Same	Check if multiple objects have the same specific attribution
Different	Check if multiple objects have different specific attributions
Compare	Compare a specific attribution of multiple objects
And	Logical and operation of specific previous results
Or	Logical or operation of specific previous results

A MORE INFORMATION OF DATASET

More Details of Explanation Construction We follow [4] to categorize 127 operations in the GQA dataset into 12 atomic operations that cover the essential semantics. For example, “same color”, “same material”, and “same shape” are merged into the “same” operation while “color”, “material”, and “shape” become their arguments. The semantics of these simplified operations are shown in Table 5.

Data Splits Since the GQA dataset does not release the scene graph annotations of their test sets, we can only leverage their training and validation set. Specifically, we adopt *balanced training set* with 943,000 samples and *balanced validation set* with 132,062 samples of GQA. After constructing explanations and removing low-quality samples, we obtain 901,203 training samples and 127,027 validation samples. Then, we further randomly select 30,000 validation samples to form the test set and the rest to form the validation set. Considering the *balanced test-dev set* of GQA only contains 12,578 samples, our test samples are already sufficient. Furthermore, to keep costs reasonable for researchers utilizing OpenAI APIs to conduct experiments on our test set, we believe that expanding the test set significantly would not be suitable. For example, our method costs around \$200 to call OpenAI APIs for a single complete run, not to mention the entire set of experiments. However, researchers can also utilize our validation set of the larger size to evaluate their methods, if the costs are not a problem for them.

Word Distribution Figure 9 shows the distribution of the first four words in our explanations. The arc lengths of the words represent their occurrence frequencies. Specially, all words with less than 1.5% frequencies are merged into gray regions (since they are hard to display). While the majority of our explanations begin with “The” or “There”, the gray region significantly expands at the following positions. This highlights the diversity of our constructed explanations in the SME dataset.

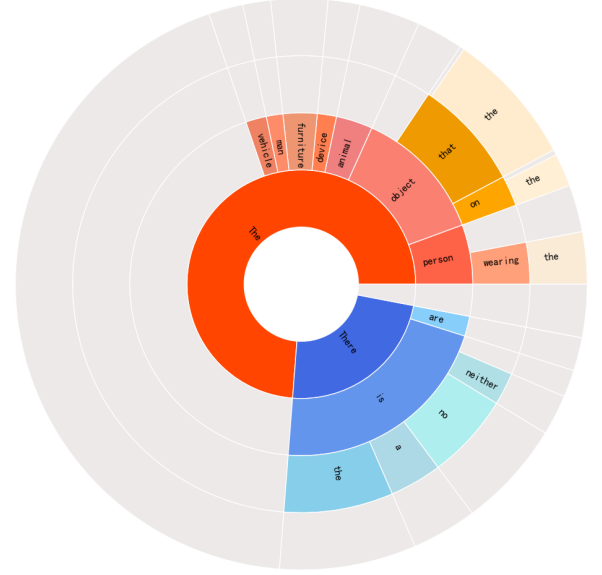


Figure 9: Distribution of first four words in the SME dataset. Gray regions denote all other words of less than 1.5% frequencies.

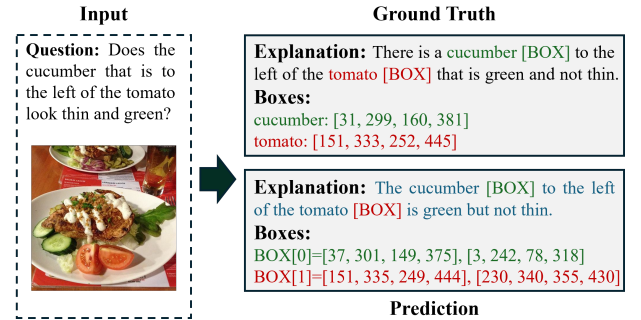


Figure 10: A formatted example of ground truth explanation and predicted explanation. While our ground truth annotates the names of all boxes, the prediction simply needs to link values to boxes by their order.

B DISCUSSION OF EVALUATION AND METRICS

While REX [4] makes the very first attempt at multimodal explanations for VQA, its adopted metrics have several problems. Accordingly, we improve the evaluation metrics in this work. We show a formatted example of the ground truth explanation and the predicted explanation in Figure 10 to facilitate comprehension.

B.1 Improving Textual Evaluation

In explanations constructed in REX [4], the visual objects are represented by $\#i$ that is the i -th object predicted by a Faster RCNN trained on MS-COCO. When using language metrics (i.e., BLEU-4 [10], METEOR [3], ROUGE-L [7], CIDEr [12], and SPICE [2]), the evaluation is sensitive to the number of visual objects. For example, for the token “ $\#i$ ” in the ground truth explanation, the token “ $\#j$ ” ($i \neq j$) in the generated explanation is considered as wrong, even though “ $\#j$ ” may represent almost the same grounding box as “ $\#i$ ”. Moreover, when the Faster R-CNN does not predict the needed visual objects, their annotated “ $\#i$ ” is thereby inaccurate, which also reduces the reliability of textual metrics.

To overcome these problems, we separate the evaluation of text generation and visual grounding. We use the [BOX] token in explanations to represent grounding boxes. Therefore, while adopting the same language metrics (i.e., BLEU-4 [10], METEOR [3], ROUGE-L [7], CIDEr [12], and SPICE [2]), our textual evaluation only requires the model to generate [BOX] tokens in the correct positions, leaving the evaluation of grounded boxes in visual metrics.

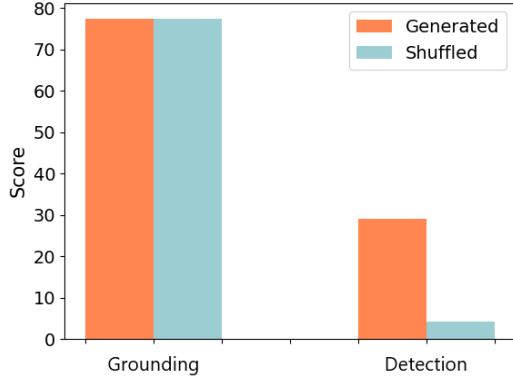


Figure 11: Comparison of the grounding metric in REX [4] and the detection metric in our SME after shuffling tokens in the generated explanations.

B.2 Improving Visual Evaluation

For a ground truth explanation E_{gt} and a reference explanation E_{re} , REX [4] directly compute the IoU score between all boxes in E_{gt} and all boxes in E_{re} . Their grounding score can be written as follows:

$$Grounding = IoU(\{boxes\ in\ E_{gt}\}, \{boxes\ in\ E_{re}\}), \quad (1)$$

which is insensitive to the position of boxes and their corresponding names in the explanation. For example, for a question “What’s the color of the apple to the left of the pear?”, two explanations “ $\#i$ to the left of $\#j$ is green” and “ $\#j$ to the left of $\#i$ is green” have the same grounding score, though $\#i$ is apple and $\#j$ is pear.

To address this problem, in our explanations, [BOX] follows the name of the corresponding object and we annotate the grounding boxes with their names mentioned in the explanation, based on scene graphs annotated by humans. Then, for every object name s annotated in a ground truth explanation (e.g., “cucumber” and

“tomato”), we match the [BOX] token following s in the reference explanation. Then, we compute the IoU (intersection of union) score of the ground truth boxes B_{gt}^s of s and the reference boxes B_{re}^s related to this [BOX] token, evaluating the detection precision of this object. The final detection score of one explanation is averaged over all object names, as follows:

$$Detection = \frac{1}{N} \sum_s IoU(B_{gt}^s, B_{re}^s), \quad (2)$$

where N is the number of object names that occur in the ground truth and reference explanations. Therefore, the redundant boxes in the reference explanation and the missing boxes can punish the final detection score. Our visual metric relates object boxes and their names for a more precise evaluation. We conduct a simple random shuffle experiment to compare the effectiveness of two visual metrics. We randomly shuffle the generated explanation tokens and compute the visual scores, as shown in Figure 11. After shuffling the tokens, the grounding score adopted in REX remained the same, while our proposed detection score significantly drops. These results verify that our metric is sensitive to the position of grounding tokens in the generated explanations.

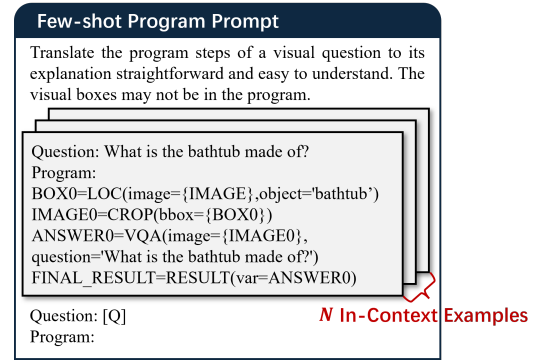


Figure 12: Few-shot program prompt in our method. [Q] denotes the input question.

C MORE DETAILS OF METHOD

Multimodal Programming. In our Multimodal Programming (MulProg), we implement 16 program modules, as shown in Table 6. Specially, *LOC* and *VQA* are implemented based on neural models. These models do not use MEVQA or GQA as training data, avoiding data leaks. Moreover, though we adopt a *VQA* module, it is not an end-to-end module in our method and we construct complex reasoning steps for reasoning. For example, for the question “What is the bathtub made of?”, instead of directly calling the *VQA* module to solve the question, our method first calls the *LOC* module to find the bathtub and then calls the *CROP* module to crop the bathtub in the image. Finally, we input the cropped image and the question into the *VQA* module for answering. These complex steps facilitate the multimodal explanation for VQA in our method. These program modules are combined to form the programs for solving visual questions.

After defining the program modules, we construct a few-shot program prompt based on $N (= 16)$ training samples, as shown in

Table 6: Program modules in our Multimodal Programming.

Definition	Smantic	Backbone
LOC(image, object)	Detect all boxes of the visual object in a image	OWL-ViT [9]
COUNT(box)	Count the number of boxes	Python
CROP(image, box)	Crop the box in the image	Python
CROP_RIGHTOF(image, box)	Crop the image right of the box	Python
CROP_LEFTOF(image, box)	Crop the image left of the box	Python
CROP_RIGHTOF(image, box)	Crop the image right of the box	Python
CROP_FRONTOF(image, box)	Crop the image in front of the box	Python
CROP_INFRONTOF(image, box)	Crop the image in front of the box	Python
CROP_BEHIND(image, box)	Crop the image behind of the box	Python
CROP_AHEAD(image, box)	Crop the image ahead of the box	Python
CROP_BELOW(image, box)	Crop the image below the box	Python
CROP_ABOVE(image, box)	Crop the image above the box	Python
VQA(image, question)	Answer the question about the image	BLIP [6]
EVAL(expression)	Evaluate the expression	Python
SIZE(box)	Evaluate the size of the box	Python
RESULT(variable)	Return the value of the variable	Python

Figure 12. By exemplifying the correspondence between questions and programs, the prompted GPT-3.5 can generate the program for the input question Q . Then, our MEAgent utilizes multimodal open-world tools to execute the multimodal program and infer the answer to the question.

D MORE DETAILS OF EXPERIMENTS

D.1 More Implementation Details

We adopt *gpt-3.5-turbo-instruct* as our backbone LLM, which is an instruct LLM needed for following the prompt instructions in our method. Since it is currently the only running OpenAI API of instruct LLMs, we do not experiment with other LLMs. Another important reason for not adopting GPT-4 is that we do not know if GPT-4 uses GQA data in its training, which can cause data leaks in our few-shot learning experiments.

For the GPT-4V [1] baseline, we adopt the *gpt-4-1106-vision-preview* version, since it is currently the most powerful version of GPT-4V, claimed by OpenAI.

In experiments of REX [4] and VCIN [13], we use the official code and carefully tune the hyper-parameters. However, we have found that the training processes of these traditional MEVQA methods may collapse with very few training samples. To address this, we simply discard these collapsed results and only adopt the converged results. We have to acknowledge that this can cause an overvaluation of these baselines.

D.2 Implementation Details of GPT-4V

Different from GPT-3.5 adopted in our method, GPT-4V [1] is a multimodal LLM that can input text, images, and videos. Since GPT-4V can generate text and detect visual objects, we construct a prompt based on the same $N (= 16)$ examples in our method, to facilitate question answering and multimodal explanation generation via GPT-4V. As shown in Figure 13, we construct a few-shot prompt based on the same $N (= 16)$ in-context examples in our method for GPT-4V. In the prompt, we exemplify the correspondence between

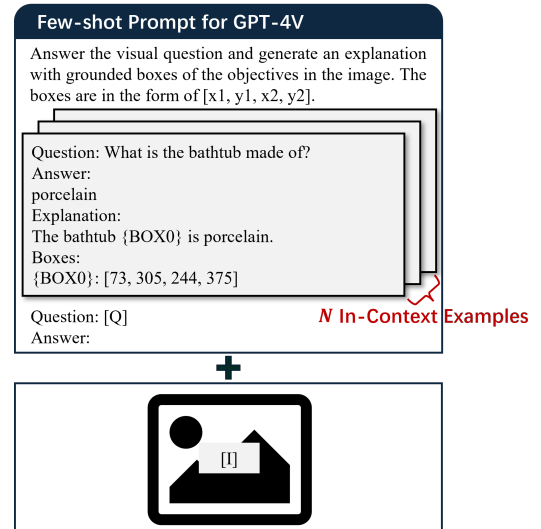


Figure 13: Few-shot prompt for GPT-4V. $[Q]$ denotes the input question and $[I]$ denotes the input image.

a question to its answer, explanation, and key object boxes. Then, by inputting the test question and its related image, GPT-4V can output the answer, explanation, and key object boxes. We further replace the box variables (i.e. $\{BOX_i\}$) in the generated explanation with $[BOX]$ linked to the corresponding variable values following “Boxes:”, to form the multimodal explanation. Notably, GPT-4V sometimes cannot find the key visual objects in the image and refuses to answer the question, outputting text such as “I’m sorry, but I cannot provide an answer to your question as there is no visible trashcan in the provided image.”.

E POTENTIAL RISK

A key potential risk in few-shot learning with trained models is data leak. Since our dataset is based on the public GQA dataset (a VQA dataset) [5], some models that share the training data in GQA should not be considered in few-shot learning. As we have checked, program modules in our MEAgent do not share the training data in our dataset. The GPT-3.5 used in our method, which is a language-only model, also does not use our multimodal data for training. For our baselines REX and VCIN, we utilize the backbone VisualBERT pretrained on MS-COCO [8] to avoid data leaks. Another backbone LXMERT [11] is pre-trained on GQA and we only adopt it in non-few-shot experiments. For GPT-4V, since there is no public information about its training data. We are not sure if GQA is used in its training. Therefore, GPT-4V may be overestimated in few-shot experiments. This is also an important reason for not using GPT-4 as the backbone of our method.

REFERENCES

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Floren-
cia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal
Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*
(2023).
- [2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016.
Spice: Semantic propositional image caption evaluation. In *Computer Vision–
ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14,
2016, Proceedings, Part V 14*. Springer, 382–398.
- [3] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for
MT evaluation with improved correlation with human judgments. In *Proceedings
of the acl workshop on intrinsic and extrinsic evaluation measures for machine
translation and/or summarization*. 65–72.
- [4] Shi Chen and Qi Zhao. 2022. Rex: Reasoning-aware and grounded explanation. In
Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
15586–15595.
- [5] Drew A Hudson and Christopher D Manning. 2019. 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*. 6700–6709.
- [6] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping
language-image pre-training for unified vision-language understanding and
generation. In *International conference on machine learning*. PMLR, 12888–12900.
- [7] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries.
In *Text summarization branches out*. 74–81.
- [8] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva
Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common
objects in context. In *Computer Vision–ECCV 2014: 13th European Conference,
Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 740–
755.
- [9] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk
Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa
Dehghani, Zhuoran Shen, et al. 2022. Simple open-vocabulary object detection.
In *European Conference on Computer Vision*. Springer, 728–755.
- [10] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a
method for automatic evaluation of machine translation. In *Proceedings of the
40th annual meeting of the Association for Computational Linguistics*. 311–318.
- [11] Hao Tan and Mohit Bansal. 2019. LXMERT: Learning Cross-Modality Encoder
Representations from Transformers. In *Proceedings of the 2019 Conference on
Empirical Methods in Natural Language Processing and the 9th International Joint
Conference on Natural Language Processing (EMNLP-IJCNLP)*. 5100–5111.
- [12] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider:
Consensus-based image description evaluation. In *Proceedings of the IEEE confer-
ence on computer vision and pattern recognition*. 4566–4575.
- [13] Dizhan Xue, Shengsheng Qian, and Changsheng Xu. 2023. Variational Causal
Inference Network for Explanatory Visual Question Answering. In *Proceedings
of the IEEE/CVF International Conference on Computer Vision*. 2515–2525.