Retrieving Visual Facts For Few-Shot Visual Question Answering

Anonymous ACL submission

Abstract

We introduce the Retrieving Visual Facts (RVF) framework for few-shot visual question answering (VQA). The RVF framework represents an image as a set of natural language facts; for example, in practice these could be tags from an object detector. Critically, the question is used to retrieve *relevant* facts: an image may contain numerous details, and one should attend to the few which may be useful for the question. Finally, one predicts the answer from the retrieved facts and the question, e.g., by prompting a language model as we do here. Compared to PICA (Yang et al., 2021), the previous state-of-the-art in few-shot VQA, a proof-of-concept RVF implementation improves absolute performance by 2.6% and 1.5% respectively on the VQAv2 (Goyal et al., 2017) and OK-VQA (Marino et al., 2019) datasets. We also analyze our implementation's strengths and weaknesses on various question types, highlighting directions for further study.

1 Introduction

Fully supervised performance on VQA datasets has risen sharply due to recent advances in neural architectures and feature representations (Anderson et al., 2018; Wu et al., 2019; Zhang et al., 2021). However, as labeled VQA data can be expensive to annotate, there has been increasing interest in *few-shot* VQA (Tsimpoukelli et al., 2021; Yang et al., 2021), for which only a handful (e.g., 16) of labeled training samples are provided. For example, the previous state-of-the-art PICA method (Yang et al., 2021) takes advantage of large pretrained models for both vision (Zhang et al., 2021) and text (Brown et al., 2020) to answer questions given only a few labeled examples in the form of a prompt.

In this work, we propose the Retrieving Visual Facts (RVF) paradigm for few-shot VQA, inspired by text-based question answering (QA) methods such as Clark and Gardner (2017). Text QA systems do not try to answer questions using a question-independent summary of a document. Rather, systems first retrieve passage(s) that are *most relevant* to the question. Here, we apply this lens to VQA. Whereas systems like PICA reduce an image to "facts" (e.g., captions or tags) without seeing the query, RVF instead views the image as a collection of facts, and retrieves the ones most relevant to the query before using them to predict the answer. Figure 1 illustrates RVF's main intuition: since an image has myriad details, one should use the question to extract the most relevant ones.

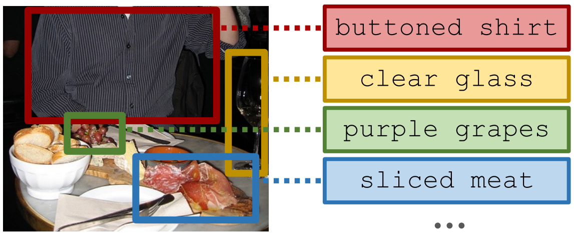

Figure 1: A VQAv2 (Goyal et al., 2017) example for intuition. We can view the image as a collection of facts. Looking first at the image and only then at the question (see footnote[1]), it may be difficult to recall the most relevant facts; this is analogous to PICA's operation. But the task is easier if one attends to relevant facts *with* question in mind, akin to RVF.

We evaluate the RVF framework using a proof-of-concept implementation RVF-P (Sec. 3), structured similarly to PICA to facilitate comparison.[2] Concretely, RVF-P generates a caption and a list of tags, with tags selected based on the question. RVF-P improves over PICA by an absolute 2.6% on the VQAv2 dataset (Goyal et al., 2017) and 1.5% on OK-VQA (Marino et al., 2019), as shown in Sec. 4. However, if we limit the evaluation to questions which are not already correctly answered by a trivial text-only baseline, RVF-P outperforms PICA by a relative 16% and 13% on the two datasets respectively. Finally, our analysis in Sec. 4.1 highlights several avenues for further improvement.

[1] Question for Fig. 1: What fruit can be seen on the table?

[2] All code will be open-sourced upon publication.

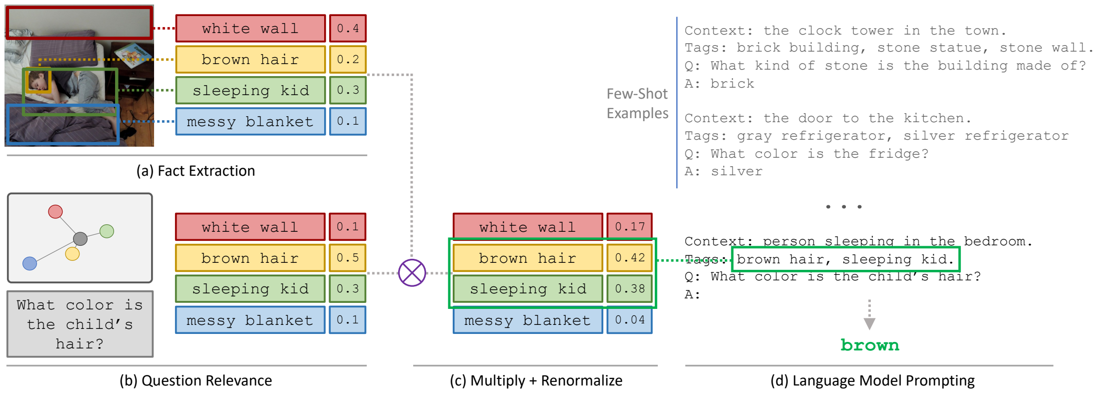
white wall
0.4
brown hair
0.2
sleeping kid
0.3
messy blanket
0.1
(a) Fact Extraction
What color is the child's hair?
white wall
0.1
brown hair
0.5
sleeping kid
0.3
messy blanket
0.1
(b) Question Relevance
white wall
0.17
brown hair
0.42
sleeping kid
0.38
messy blanket
0.04
(c) Multiply + Renormalize
Few-Shot Examples
Context: the clock tower in the town.
Tags: brick building, stone statue, stone wall.
Q: What kind of stone is the building made of?
A: brick
Context: the door to the kitchen.
Tags: gray refrigerator, silver refrigerator
Q: What color is the fridge?
A: silver
. . .
Context: person sleeping in the bedroom.
Tags: brown hair, sleeping kid.
Q: What color is the child's hair?
A:
brown
(d) Language Model Prompting

Figure 2: Overview of RVF-P, our implementation of the RVF framework. **(a)** An object detector extracts facts (here, tags), scoring by confidence/salience. **(b)** The same tags are scored by question relevance; note PICA does not do this. **(c)** Probabilities from (a) and (b) are multiplied and renormalized. **(d)** The top tags from (c) are used in the prompt; for few-shot evaluation, the process is repeated on some training set examples. The full prompt is finally fed to a language model.

2 Related Work

Recent work has substantially advanced VQA performance, whether by modifying the model architecture (Anderson et al., 2018; Lu et al., 2019; Wu et al., 2019, 2021), or improving underlying feature representations (Li et al., 2020; Zhang et al., 2021). In particular, Hu et al. (2019) explore question-conditioned image processing, though not in a few-shot setting. Additionally, the advent of large pretrained vision-language models such as CLIP (Radford et al., 2021) and GLIP (Li et al., 2021) suggests leveraging a pretrained backbone for training and/or fine-tuning (Shen et al., 2021).

Pretrained models have also enabled few-shot or even zero-shot VQA (Tsimpoukelli et al., 2021; Wang et al., 2021; Yang et al., 2021). For example, SIMVLM (Wang et al., 2021) pretrains a large vision-language model for use on downstream tasks. Most similar to our approach is PICA (Yang et al., 2021), which uses an off-the-shelf captioning model to write a text description of the image before querying a language model for the answer. RVF also constructs a text description of the image and then uses a language model to predict the answer, but the key difference is that we condition the description on the question.

Finally, RVF is inspired by question answering methods in text domains. Such methods typically use the question to select a relevant passage, whether from a document (Lei et al., 2016; Clark and Gardner, 2017) or from a large database (Chen et al., 2017; Karpukhin et al., 2020), before extracting the answer from the selected passage. RVF uses a similar idea in VQA, using the question to select facts about the image.

3 Retrieving Visual Facts

Below we describe our proposed Retrieving Visual Facts (RVF) framework for few-shot VQA.

1. Generate a set $\mathcal{F}$ of natural language facts for the given image.
2. Select the facts f from $\mathcal{F}$ most relevant to the given question.
3. Answer the question using the selected f's.

To analyze RVF empirically, we run a proof-of-concept implementation (denoted RVF-P) of the RVF framework. Concretely, RVF-P builds on PICA (Yang et al., 2021), which first generates a caption and a list of tags for the image using off-the-shelf models, then concatenates that description with the question, and finally uses the result to query GPT3 (Brown et al., 2020) for an answer.

Therefore, RVF-P works as follows. We can view a tagging model as generating a subset of possible facts about an image, i.e., facts about the presence of particular objects. Thus we apply a pretrained object detector $\mathcal{D}$ (Zhang et al., 2021) to the image to obtain the set $\mathcal{F}$, where each fact f is a detected adjective-noun pair (e.g., "white wall"; Fig. 2a).[3] From $\mathcal{D}$ we obtain a probability $\mathcal{D}(f)$, corresponding to the confidence of each tag f.

Next, different from PICA, we apply a text encoder $\mathcal{S}$ designed for semantic relevance (Reimers and Gurevych, 2019) to both the question as well as each tag. The dot product of encoded represen-

[3] Zhang et al. (2021) actually output noun tags with a list of adjectives for each, and we consider all adjective-noun pairs. The resulting tags differ slightly from those used in PICA, which uses only nouns, but we observe that the question conditioning aspect is the main driver of performance improvement in our ablations (Sec. 4.1, Table 2).

tations yields a relevance score. The result is a distribution over tags based on question relevance, with probability $\mathcal{S}(f)$ for each f (Fig. 2b).

We combine the two distributions by assigning tag probabilities proportional to $\mathcal{D}(f) \times \mathcal{S}(f)$ (Fig. 2c). The final tag list is created by greedily selecting the top remaining tag until the cumulative probability reaches 0.8, inspired by nucleus sampling (Holtzman et al., 2019). In practice, this procedure usually selects 2 to 4 tags. Following PICA, these tags f are concatenated to a generated image caption c, before being fed together with the question q to a language model (Fig. 2d). For few-shot evaluation, the entire process is repeated to yield text descriptions for several training set examples; their respective c, f's, q, and answers a are prepended to the prompt for the language model.

We emphasize that RVF-P is a proof of concept. For instance, a more sophisticated implementation of RVF might not limit the fact set $\mathcal{F}$ to tags, and we leave such extensions to future work.

4 Experiments

Datasets. We evaluate on the VQAv2 (Goyal et al., 2017) and OK-VQA (Marino et al., 2019) datasets (both English). VQAv2 questions typically ask about lower-level visual details of an image, while OK-VQA questions generally require more commonsense knowledge. For each dataset, we evaluate on a random size-3000 subset of the validation set due to GPT3 API costs.

Methods Evaluated. We run the methods below.

1. NOIMAGE, a text-only baseline which predicts the answer from a language model using just the question. This weak method can be viewed as a per-question "majority baseline" for the language model.
2. PICA (Yang et al., 2021), the state-of-the-art for few-shot VQA, which generates a caption and tags for an image before predicting the answer with a language model.
3. RVF-P, our implementation of RVF. It also predicts the answer given a caption and tags from the same models as PICA, but selects tags conditioned on the question.

For fair comparison, all methods use GPT3-13B as the language model (not GPT3-175B, due to cost limitations). For both PICA and RVF-P, we use Clipcap (Mokady et al., 2021) for captioning, and the VinVL object detector (Zhang et al., 2021) for tagging. We use models trained on Conceptual Captions (Sharma et al., 2018) for both captioning and tagging in keeping with a strict few-shot setting for VQAv2 and OK-VQA, which are based on COCO images (Lin et al., 2014). Note that these latter models are smaller than those used in the original PICA work (some of which are non-public), so our numbers are systematically lower; indeed RVF-P's performance also varies based on choices such as GPT3 size (Appendix A).

We evaluate each method in a 16-shot scenario, prompting using random training set examples.

Results. RVF-P indeed significantly outperforms PICA by 2.6% and 1.5% respectively on VQAv2 and OK-VQA (Table 1). We additionally observe that most questions that both methods get right are also answered correctly by the trivial NOIMAGE "majority baseline." Excluding these easy questions where GPT3-13B predicts the correct answer without even using the image, RVF-P gets 16% more questions correct than PICA on VQAv2 and 13% more on OK-VQA.

Method	VQAv2	OK-VQA
NOIMAGE	40.4	23.6
PICA	48.9*	34.0*
RVF-P	**51.5**	**35.5**
SUPERVISED SOTA	*77.5*	*54.4*

Table 1: Main 16-shot results on VQAv2 and OK-VQA on size-3000 samples of validation set, with supervised state-of-the-art (Zhang et al., 2021; Gui et al., 2021) included below for reference. RVF-P outperforms the previous few-shot state-of-the-art, PICA, on both datasets ($p < 10^{-4}$ and $p < 0.02$ respectively on a paired t-test). *Lower than originally reported in Yang et al. (2021) due to smaller captioning/tagging/language models.

4.1 Analysis

We conduct additional analyses on VQAv2 to shed light on where RVF-P improves over PICA and where room for further improvement remains. See Appendix A for further analyses on the contents of the image description fed to the language model, and the size of the language model itself.

Tag Selection. To confirm that it is the question conditioning rather than our more detailed tags (compared to PICA) which make the difference in performance, we run a version of PICA (PICA-MATCHTAGS) which selects the same number of tags per question from the same set of tags as RVF-

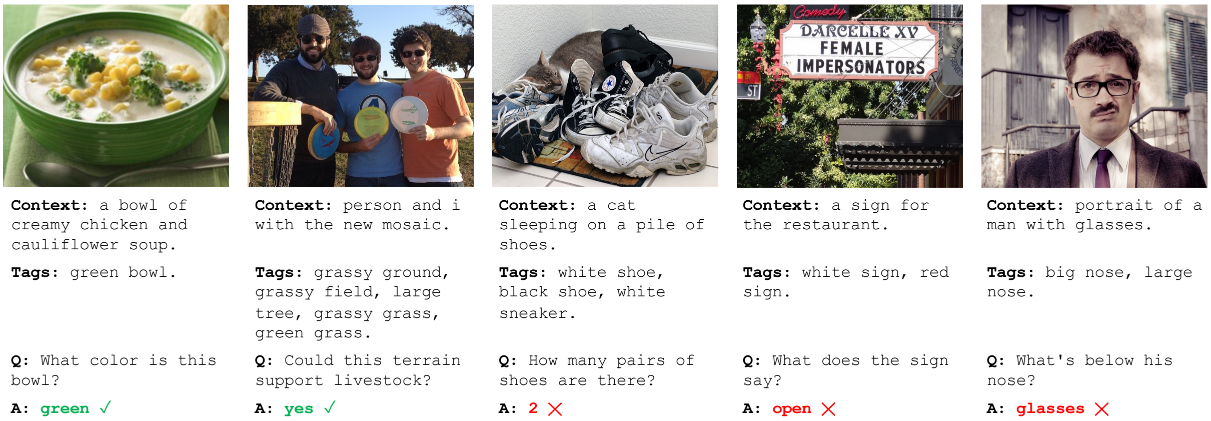
Context: a bowl of creamy chicken and cauliflower soup.
Tags: green bowl.
Q: What color is this bowl?
A: green ✓
Context: person and i with the new mosaic.
Tags: grassy ground, grassy field, large tree, grassy grass, green grass.
Q: Could this terrain support livestock?
A: yes ✓
Context: a cat sleeping on a pile of shoes.
Tags: white shoe, black shoe, white sneaker.
Q: How many pairs of shoes are there?
A: 2 ✗
Context: a sign for the restaurant.
Tags: white sign, red sign.
Q: What does the sign say?
A: open ✗
Context: portrait of a man with glasses.
Tags: big nose, large nose.
Q: What's below his nose?
A: glasses ✗

Figure 3: Example outputs from RVF-P, illustrating strengths and failure modes. **Far Left:** RVF-P is good at simple attributes (whereas PICA struggles). **Center Left:** RVF-P can identify relevant attributes ("grassy") even when queried indirectly. **Center:** RVF-P's tags are not designed for counting. **Center Right:** RVF-P's tags are not designed for reading text. **Far Right:** RVF-P struggles with spatial/relational queries. Although these failures are a limitation of RVF-P, they are not necessarily a limitation of the general RVF framework.

P, but which does not use relevance to the question when selecting. Table 2 demonstrates that PICA-MATCHTAGS remains worse than RVF-P.

Method	VQAv2
RVF-P	**51.5**
PICA-MATCHTAGS	48.3

Table 2: Performance of RVF-P compared to a version of PICA which selects the same number of tags from the same set of tags. RVF-P still performs better ($p < 10^{-7}$).

Error Analysis. VQAv2 questions can be categorized by answer format into three groups: "yes/no," "counting," and "other" (OK-VQA questions are almost exclusively "other"). We break down different methods' accuracy by category in Table 3.

RVF-P significantly improves over PICA on "yes/no" and "other" questions. In particular, we observe that RVF-P performs well on questions about attributes of objects (Fig. 3 far left). RVF-P can also identify tags which are indirectly related to the question (Fig. 3 center left). However, RVF-P struggles on "counting" questions ("How many elephants are there?"), which are a systematic failure mode of RVF-P (as well as PICA). Captions and tags are a poor match for such questions. Figure 3 shows additional failure modes, such as reading text and spatial/relational queries. But we emphasize that while such failures are a limitation of our implementation RVF-P, they could in principle be handled consistently by the RVF framework if the fact generator provided the necessary facts.

Method	Yes/No	Counting	Other
Total	1149	390	1461
NOIMAGE	68.1	27.1	22.3
PICA	69.3	30.7	37.6
RVF-P	**71.4**	32.2	**41.0**

Table 3: Accuracies on different VQAv2 question types for different methods. On the two categories where we would expect to see improvement—"yes/no" and "other" questions—RVF-P improves over PICA with $p < 0.01$.

5 Discussion

We have proposed the RVF framework for few-shot VQA, which uses the question to select relevant facts about the image. Our implementation RVF-P outperforms the previous state-of-the-art PICA, but several avenues for improvement remain.

One major direction is to improve the set of initially extracted facts about the given image. The tags output by an object detector represent only a limited subset of true facts, resulting in failures on certain types of questions as analyzed in Sec. 4.1. More sophisticated methods for extracting facts relating to e.g., object counts or spatial/relational information could substantially improve performance. Moreover, the initial fact extraction model could itself be question-conditioned in principle.

Finally, due to the modular nature of the approach—using several independent pretrained models for tagging, captioning, question relevance, and final answer extraction—one can improve performance by exchanging any one of these models for a better-performing version.

Ethical Considerations

As with any work relying heavily upon large pre-trained models such as GPT3, we may inherit the biases of such models (Brown et al., 2020). Nevertheless, we believe our work makes a positive impact overall; for example, advances in VQA have the potential to improve accessibility for the visually impaired.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Christopher Clark and Matt Gardner. 2017. Simple and effective multi-paragraph reading comprehension. *arXiv preprint arXiv:1710.10723*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liangke Gui, Borui Wang, Qiuyuan Huang, Alex Hauptmann, Yonatan Bisk, and Jianfeng Gao. 2021. Kat: A knowledge augmented transformer for vision-and-language. *arXiv preprint arXiv:2112.08614*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Ronghang Hu, Anna Rohrbach, Trevor Darrell, and Kate Saenko. 2019. Language-conditioned graph networks for relational reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10294–10303.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. 2021. Grounded language-image pre-training. *arXiv preprint arXiv:2112.03857*.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer.

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 *European conference on computer vision*, pages 740–755. Springer.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3195–3204.

Ron Mokady, Amir Hertz, and Amit H Bermano. 2021. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. 2021. How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*.

Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2021. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*.

Jialin Wu, Zeyuan Hu, and Raymond J Mooney. 2019. Generating question relevant captions to aid visual question answering. *arXiv preprint arXiv:1906.00513*.

Jialin Wu, Jiasen Lu, Ashish Sabharwal, and Roozbeh Mottaghi. 2021. Multi-modal answer validation for knowledge-based vqa. *arXiv preprint arXiv:2103.12248*.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. 2021. An empirical study of gpt-3 for few-shot knowledge-based vqa. *arXiv preprint arXiv:2109.05014*.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588.

A Additional Analyses

Language Model Size. We run versions of RVF-P with different sizes of GPT3 (our main results use GPT3-13B). Table 4 demonstrates that the size of the language model substantially impacts the accuracy. We hypothesize that using the full GPT3-175B would further improve performance; indeed, the original PICA work obtains higher numbers than our re-implementation due to their use of larger models such as GPT3-175B. This analysis also highlights the modularity of this approach: any of the component models can be exchanged for better-performing versions without affecting the rest of the system.

Method	VQAv2
RVF-P-GPT3-13B	**51.5**
RVF-P-GPT3-6.7B	46.1
RVF-P-GPT3-2.7B	41.2

Table 4: Comparison of RVF-P with different language model sizes. We use GPT3-13B for our main results. The performance improves with better language models ($p < 10^{-11}$), and we hypothesize that using the full GPT3-175B or an even larger model would yield further benefits.

Context Components. We conduct further analyses on which parts of the context are necessary for GPT3 to extract the answer, by ablating the tags (also explored in Yang et al. (2021)) and also the caption itself. As shown in Table 5, removing either component results in a drop in performance, although it is interesting that one can obtain decent performance with only a tagging model.

Method	VQAv2
RVF-P	**51.5**
RVF-P-NOTAGS	47.6
RVF-P-NOCAPTION	48.7

Table 5: Ablations on the image description used as context for GPT3. RVF-P uses both a caption and a list of tags; removing either the tags or the caption results in lower performance ($p < 10^{-4}$).

B Computational Details

Code primarily relies on four off-the-shelf models. The captioning model Clipcap (Mokady et al., 2021), trained on Conceptual Captions (Sharma et al., 2018), has 156M parameters. The object detector from VinVL (Zhang et al., 2021) is also trained on Conceptual Captions and is 147M parameters. The semantic relevance model (Reimers and Gurevych, 2019) can be found at https://huggingface.co/sentence-transformers/multi-qa-mpnet-base-dot-v1 and has 109M parameters. Finally, the GPT3 language model (Brown et al., 2020) that we use has 13B parameters.

The total computational budget was roughly $600 for the GPT3 API and fewer than 100 GPU hours for other models, including both preliminary investigations and final experiments.