Prompting Large Language Models with Answer Heuristics for Knowledge-based Visual Question Answering

Zhenwei Shao[1] Zhou Yu[1]* Meng Wang[2] Jun Yu[1]
[1]Key Laboratory of Complex Systems Modeling and Simulation,
School of Computer Science and Technology, Hangzhou Dianzi University, China.
[2]School of Computer Science and Information Engineering, Hefei University of Technology, China
{shaozw, yuz, yujun}@hdu.edu.cn, eric.mengwang@gmail.com
Code: https://github.com/MILVLG/prophet

Abstract

*Knowledge-based visual question answering (VQA) requires external knowledge beyond the image to answer the question. Early studies retrieve required knowledge from explicit knowledge bases (KBs), which often introduces irrelevant information to the question, hence restricting the performance of their models. Recent works have sought to use a large language model (i.e., GPT-3 [3]) as an implicit knowledge engine to acquire the necessary knowledge for answering. Despite the encouraging results achieved by these methods, we argue that they have not fully activated the capacity of GPT-3 as the provided input information is insufficient. In this paper, we present Prophet—a conceptually simple framework designed to **pro**m**p**t GPT-3 with answer **he**uris**t**ics for knowledge-based VQA. Specifically, we first train a vanilla VQA model on a specific knowledge-based VQA dataset without external knowledge. After that, we extract two types of complementary answer heuristics from the model: answer candidates and answer-aware examples. Finally, the two types of answer heuristics are encoded into the prompts to enable GPT-3 to better comprehend the task thus enhancing its capacity. Prophet significantly outperforms all existing state-of-the-art methods on two challenging knowledge-based VQA datasets, OK-VQA and A-OKVQA, delivering 61.1% and 55.7% accuracies on their testing sets, respectively.*

1. Introduction

Recent advances in deep learning have enabled substantial progress in visual question answering (VQA) which requires a machine to answer free-form questions by reasoning about given images. Benefiting from large-scale vision-language pretraining, the state-of-the-art methods have even surpassed human level on several representative benchmarks [1,41,48]. Despite the success of these methods, their reasoning abilities are far from satisfactory, especially when *external knowledge* is required to answer the questions. In this situation, the task of knowledge-based VQA is introduced to validate models' abilities to leverage external knowledge. Early knowledge-based VQA benchmarks additionally provide structured knowledge bases (KBs) and annotate required knowledge facts for all the questions [38,39]. More recently, benchmarks emphasizing on *open-domain* knowledge have been established [29, 32], which means KBs are no longer provided and any external knowledge resource can be used for answering. We focus on the task with open-domain knowledge in this paper.

A straightforward solution for knowledge-based VQA is to retrieve knowledge entries from explicit KBs, *e.g.*,

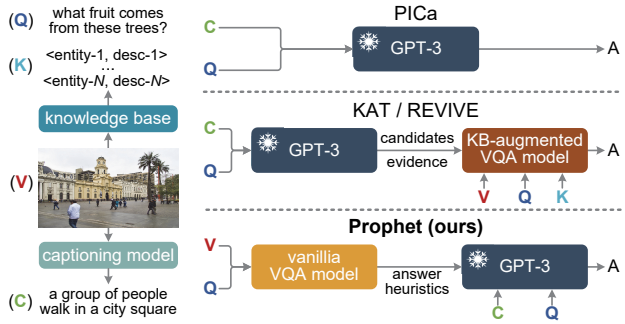


Figure 1. Conceptual comparisons of three knowledge-based VQA frameworks using a frozen GPT-3 model [3]. While PICa [43], KAT [11], and REVIVE [22] directly feed the caption (C) and question (Q) into GPT-3 as the prompt, we argue that the information they provide for GPT-3 is insufficient thus cannot fully activate GPT-3's potential. In contrast, our Prophet learns a vanilla VQA model without external knowledge to produce *answer heuristics*, which endows GPT-3 with richer and more task-specific information for answer prediction.

*Zhou Yu is the corresponding author

Wikipedia and ConceptNet [23]. Then a KB-augmented VQA model performs joint reasoning over the retrieved knowledge, image, and question to predict the answer [7, 8, 28, 42, 51]. However, the performance of these retrieval-based approaches is limited for two reasons: (i) the required knowledge may not be successfully retrieved from the KBs; and (ii) even if the required knowledge is retrieved, plenty of irrelevant knowledge is inevitably introduced, which hampers the learning of VQA models.

Apart from those studies using explicit KBs, another line of research resorts to pretrained large language models, *e.g.*, GPT-3 [3], as implicit knowledge engines for knowledge acquisition. A pioneering work by PICa employs the frozen GPT-3 model to answer the question with formatted prompt as its input [43]. Given a testing image-question pair, PICa first translates the image into a caption using an off-the-shelf captioning model. The question, caption, and a few in-context examples are then integrated into a textual prompt that can induce GPT-3 to predict the answer directly. Thanks to the powerful knowledge reasoning ability of GPT-3, PICa achieves significant performance improvements compared to those retrieval-based methods using explicit KBs. Inspired by PICa, KAT [11] and REVIVE [22] learn KB-augmented VQA models to exploit both the implicit knowledge from GPT-3 and explicit knowledge from KBs for answer prediction. The synergy of the two knowledge resources brings further improvements to their models. Despite the promising results achieved by these methods, they have not fully activated GPT-3 due to the following limitations:

(i) The generated captions cannot cover all the necessary information in the image. Consider the example in Figure 1, the caption "a group of people walk in a city square" contribute nothing to answering the question "what fruit comes from these trees". In this situation, GPT-3 has to make an aimless and biased guess to answer the question.

(ii) GPT-3 employs a few-shot learning paradigm that requires a few in-context examples to adapt to new tasks. Therefore, the choice of these examples is critical to model performance. As reported in [43], all its example selection strategies achieve far inferior performance to the oracle strategy that uses the similarity of ground-truth answers.

We ask: *Is it possible to endow GPT-3 with some* ***heuristics*** *to enhance its capacity for knowledge-based VQA?*

In this paper, we present **Prophet**—a conceptually simple framework designed to **pro**m**p**t GPT-3 with answer **he**uris**t**ics for knowledge-based VQA. By answer heuristics, we mean some promising answers that are presented in a proper manner in the prompt. Specifically, we introduce two types of answer heuristics, namely *answer candidates* and *answer-aware examples*, to overcome the limitations in (i) and (ii), respectively. Given a testing input consisting of an image and a question, the answer candidates refer to a list of promising answers to the testing input, where each answer is associated with a confidence score. The answer-aware examples refer to a list of in-context examples, where each example has a similar answer to the testing input. Interestingly, these two types of answer heuristics can be simultaneously obtained from any vanilla VQA model trained on a specific knowledge-based VQA dataset. A schematic of Prophet is illustrated at the bottom of Figure 1.

Without bells and whistles, Prophet surpasses all previous state-of-the-art single-model results on the challenging OK-VQA and A-OKVQA datasets [29, 32], including the heavily-engineered Flamingo-80B model trained on 1.8B image-text pairs [1]. Moreover, Prophet is friendly to most researchers, as our results can be reproduced using a single GPU and an affordable number of GPT-3 invocations.

2. Related Work

Visual Question Answering (VQA). VQA has been of growing interest over the last few years. Recent studies in VQA research can be roughly divided into the following categories: better visual features [2, 15, 33, 49], more powerful model architectures [13, 17, 20, 45, 47], and more effective learning paradigms [4, 5, 19, 21, 25, 34, 50]. Most current state-of-the-art VQA methods employ the Transformer architecture [36]. By incorporating vision-language pretraining on large-scale datasets, they have approached or even surpassed human-level performance on several representative benchmarks [1, 40, 41, 44, 48]. Besides these studies on general-purpose VQA, there is also a growing trend towards exploring more granular VQA tasks with specific reasoning skills, *e.g.*, neural-symbolic reasoning [14, 16] and knowledge utilization [29, 30, 38].

Knowledge-based VQA. The core of this task lies in knowledge acquisition and integration. Early explorations parse the inputs into structured queries and retrieve supporting knowledge from fixed knowledge bases (KBs) to obtain the answers [38, 39]. As the provided knowledge resources are not sufficient to represent general knowledge, subsequent research mainly focuses on acquiring explicit knowledge from multiple open-domain knowledge resources, *e.g.*, ConceptNet [23], Wikipedia [37], and Google Images [42]. This retrieved knowledge is integrated with the image-question pair for answer prediction [8, 27, 42]. Motivated by the promising capacities of large language models (*e.g.*, GPT-3 [3]) in knowledge reasoning, recent state-of-the-art approaches regard GPT-3 as an implicit knowledge engine. They either utilize it to get answer prediction directly [43] or to extract answer candidates with evidence to improve

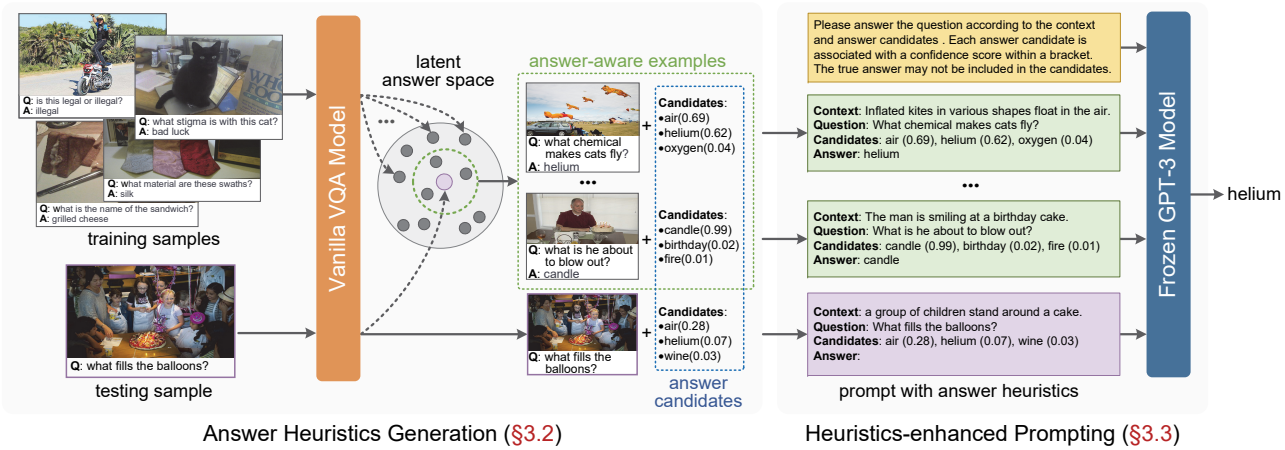


Figure 2. **Our Prophet framework** has two stages: answer heuristics generation and heuristics-enhanced prompting. In the answer heuristics generation stage, a vanilla VQA model trained on the knowledge-based VQA dataset is employed to generate two types of answer heuristics, *i.e.*, answer candidates and answer-aware examples. In the heuristics-enhanced prompting stage, the answer heuristics, question, and caption are integrated into a formatted prompt to instruct GPT-3 to predict an answer. As shown in the example, both answer heuristics contribute to the answer of "helium".

answer prediction [11, 22]. Similar to [43], our Prophet uses GPT-3 to predict answers directly. We believe the few-shot learning capability of GPT-3 has not been fully activated and this motivates us to prompt GPT-3 with answer heuristics.

In-context learning. Unlike the *pretrain-then-finetune* paradigm for language models like BERT [6], GPT-3 introduces a novel in-context few-shot learning paradigm. To adapt to a new task, GPT-3 only needs to concatenate a few examples of the task with the input as the *prompt* at inference time and requires no parameter updates. This appealing property has inspired research on training multimodal few-shot learners [1, 35]. Empirical studies show that a huge model (*e.g.*, 80B parameters in Flamingo [1]) is required for effective few-shot learning, which is unaffordable for most people to reproduce their results.

3. The Prophet Framework

Our Prophet is a conceptually simple two-stage framework. In the answer heuristics generation stage, a vanilla VQA model is learned to generate two types of answer heuristics, *i.e.*, answer candidates and answer-aware examples (detailed in §3.2). In the heuristics-enhanced prompting stage, the answer heuristics, question, and caption are integrated into a formatted prompt to instruct GPT-3 to predict an answer (detailed in §3.3). An overview of the Prophet framework is depicted in Figure 2.

3.1. Preliminaries

Before presenting the Prophet, we briefly introduce the in-context learning paradigm developed by GPT-3 and its adaptation to knowledge-based VQA by PICa [43].

GPT-3 is an autoregressive language model pretrained on a tremendous dataset. During inference, in-context few-shot learning formulates a new downstream task as a text sequence generation task on the frozen GPT-3 model. Given a testing input $\boldsymbol{x}$, its target $\boldsymbol{y}$ is predicted conditioned on a formatted prompt $\boldsymbol{p}(\boldsymbol{h}, \mathcal{E}, \boldsymbol{x})$, where $\boldsymbol{h}$ refers to a prompt head, *aka* instruction, that describes the task, $\mathcal{E} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_n\}$ corresponds to n in-context examples. Denoting the target $\boldsymbol{y} = (y^1, y^2, ..., y^L)$ as a text sequence of L tokens, at each decoding step l, we have:

$$y^l = \underset{\hat{y}^l}{\text{argmax}}\, p_{\text{GPT-3}}(\hat{y}^l | \boldsymbol{p}, y^{<l}) \tag{1}$$

where each in-context example $\boldsymbol{e}_i = (\boldsymbol{x}_i, \boldsymbol{y}_i)$ contains an input-target pair of the task, which is constructed manually or sampled from the training set.

To adapt GPT-3 to address the knowledge-based VQA task, the key is to design the appropriate prompts. Given a question q and an image v as inputs, the VQA task aims to predict a target answer a. Since GPT-3 does not understand images intrinsically, the image needs to be translated into a caption c using an off-the-shelf captioning model. PICa formulates the testing input $\boldsymbol{x}$ as the following template:

```
Context: c \n Question: q \n Answer:
```

where the variables marked in blue will be substituted by specific testing inputs. \n stands for a carriage return in the template. Accordingly, each in-context example $\boldsymbol{e_i}$ is formulated into a similar template as follows:

```
Context: c_i \n Question: q_i \n Answer: a_i
```

where c_i, q_i, and a_i refer to an image-question-answer triplet collected from the training set. The complete prompt

of PICa consists of a fixed prompt head, a few in-context examples, and a testing input. This prompt is fed into GPT-3 for answer prediction.

Our Prophet inherits the pipeline of PICa. In addition, we introduce answer heuristics into the prompt structure to better activate the capacity of GPT-3, which leads to more accurate answers.

3.2. Stage-1. Answer Heuristics Generation

We introduce two types of answer heuristics: answer candidates and answer-aware examples. Given a testing input consisting of an image and a question, the answer candidates refer to a list of promising answers to the testing input, where each answer is associated with a confidence score. The answer-aware examples refer to a list of in-context examples, where each example has similar answers to the testing input.

Interestingly, these two types of answer heuristics can be obtained simultaneously from any vanilla VQA model trained on the knowledge-based VQA task.

Denote a VQA dataset as $\mathcal{D} = \{(v_i, q_i, a_i)\}_{i=1}^{M}$, where v_i, q_i, a_i refer to the image, question, and answer, respectively. The most frequent answers in the training set form an answer vocabulary $\mathcal{W} = \{w_j\}_{j=1}^{S}$. A vanilla VQA model $\mathcal{M}$ is learned from $\mathcal{D}$ to perform an S-way classification over the answers. Generally, the VQA model can be separated into two submodels, *i.e.*, a backbone $\mathcal{M}_b$ and a classification head $\mathcal{M}_h$. The backbone $\mathcal{M}_b$ acts as an encoder to fuse multimodal inputs v and q and obtain a fused feature z:

$$z = \mathcal{M}_b(v, q) \tag{2}$$

The classification head $\mathcal{M}_h$ simply adopts a linear layer followed by a sigmoid function to project the fused feature z into a prediction vector $y \in \mathbb{R}^S$ over the answer vocabulary:

$$y = \mathcal{M}_h(z) \tag{3}$$

where $y_{[i]}$ denotes the i-th element of y, representing the confidence score for answer w_i. Based on the above definitions, we explain how to generate the two types of answer heuristics below. Note that although the learned VQA model $\mathcal{M}$ does not incorporate any external knowledge, it can be used for knowledge-based VQA when trained properly. We regard it as a reference model and compare its performance to Prophet in the experiments to show the effectiveness of GPT-3 for knowledge-based VQA.

Answer candidates. Given a testing input (v, q), we obtain its prediction vector y from Eq.(3). After that, we select the top-K answers with the highest scores:

$$\mathcal{I}_{\text{AC}} = \underset{j \in \{1,2,\ldots,S\}}{\text{argTopK}} \; y_{[j]} \tag{4}$$

where $\mathcal{I}_{\text{AC}}$ denotes an index set of the top-K answer candidates. The answer candidates $\mathcal{C}$ are defined as follows:

$$\mathcal{C} = \{(w_j, y_{[j]}) \mid j \in \mathcal{I}_{\text{AC}}\} \tag{5}$$

where w_j and $y_{[j]}$ are an answer candidate and its confidence score, respectively. To make the formats of the in-context examples and testing input consistent, for each example $\boldsymbol{e_i}$ we also calculate and provide a set of answer candidates $\mathcal{C}_i$.

Answer-aware examples. Several previous studies have shown that the choice of in-context examples is crucial for GPT-3's few-shot learning performance [24, 43]. Their results motivate us to devise an *answer-aware* example selection strategy.

Given a testing input (v, q) and any training input (v_i, q_i), we can obtain their corresponding fused features z and z_i from Eq.(2) using the trained model. Since the fused features are linearly projected for answer prediction, we conjecture that these fused features lie in a *latent answer space* that contains rich semantics of the answers to the given image-question pairs. If z and z_i are close in the latent space, they are more likely to share similar answers and image-question inputs.

We calculate the cosine similarity of the fused feature between the testing input and each training input, then select top-N nearest neighbors in the latent space as the answer-aware examples:

$$\mathcal{I}_{\text{AE}} = \underset{i \in \{1,2,\ldots,M\}}{\text{argTopN}} \; \frac{z^T z_i}{\|z\|_2 \|z_i\|_2} \tag{6}$$

where $\mathcal{I}_{\text{AE}}$ is an index set of the top-N similar samples in $\mathcal{D}$. The answer-aware examples $\mathcal{E}$ are defined as follows:

$$\mathcal{E} = \{(v_i, q_i, a_i) \mid i \in \mathcal{I}_{\text{AE}}\} \tag{7}$$

Note that the fused features of the training inputs can be computed and stored beforehand, allowing efficient answer-aware example selection.

3.3. Stage-2. Heuristics-enhanced Prompting

In this stage, we use the obtained answer heuristics, *i.e.*, answer candidates $\mathcal{C}$ and answer-aware examples $\mathcal{E}$, to obtain a heuristics-enhanced prompt that facilitates the few-shot learning capacity of GPT-3 for knowledge-based VQA.

A prompt consists of a prompt head, a set of in-context examples, and a testing input. The prompt head describe the VQA task in natural language. We refer to the prompt head designed in PICa and supplement it with a new description of the answer candidates. Although we encourage GPT-3 to generate answers according to the answer candidates, we also allow it to take broad explorations and generate answers beyond the candidates. The complete format of our prompt head is shown in the yellow box of Figure 2.

Our in-context examples are derived from the obtained N answer-aware examples $\mathcal{E} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_N\}$. Based on PICa's template in §3.1, for example $\boldsymbol{e}_i$, we introduce its answer candidates $\mathcal{C}_i$ by adding *one* line of code as follows:

```
Context: c_i \n Question: q_i \n
Candidates: w_j1(y_[j1]), w_j2(y_[j2]),...,w_jK(y_[jK]) \n
Answer: a_i
```

where $j_1, j_2, \cdots, j_K$ correspond to the actual indices of the elements in $\mathcal{C}_i$. Each answer candidate w_{j_k} is paired with its confidence score $y_{[j_k]}$ within a bracket. The confidence scores additionally offer the reliability of the corresponding answer candidates, which helps GPT-3 focus more on the promising candidates and be more tolerant of the less relevant candidates. For the testing input, its template is similar to that for the in-context examples, except that the answer slot is left blank for GPT-3 to fill with.

To better exploit available examples, we use the multi-query ensemble strategy [43]. Specifically, we increase the number of answer-aware examples to N*T to obtain T paralleled prompts, where each prompt still contains N examples. By prompting GPT-3 for T times, we obtain T answer predictions. The majority voting is performed over the T predictions to determine the final answer. The effects of different N and T will be verified in the experiments.

4. Experiments

We evaluate the performance of Prophet on two prevalent knowledge-based VQA datasets: OK-VQA [29] and A-OKVQA [32]. We conduct comprehensive ablation experiments to explore the effectiveness of Prophet. By taking the ablation results into account, we perform thorough comparisons of Prophet and state-of-the-art methods.

4.1. Datasets

OK-VQA is a commonly used knowledge-based VQA dataset [29]. The dataset contains 9K and 5K image-question pairs for training and testing, respectively. All questions are manually filtered to ensure that outside knowledge is required to answer the questions. Each data sample is annotated with ten open-ended answers. The accuracy computed by the soft scores is used as the evaluation metric [10]. We use the 1.1 version of OK-VQA in the experiments.

A-OKVQA is currently the largest knowledge-based VQA dataset [32]. The dataset is split into three subsets: 17K training, 1K validation, and 7K testing. Each question is annotated with ten open-ended answers for direct answer (DA) evaluation. In addition, it provides a multiple choice (MC) evaluation to ask models to choose the correct answer from four choices.

4.2. Implementation Details

We use the MCAN-large [46] as our default VQA model to generate answer heuristics. To improve the model capability, we modify the original MCAN model by: (i) replacing the original bottom-up-attention region-based features with the grid-based features extracted from CLIP's visual encoder with a RN50×64 backbone [31]; and (ii) replacing the original LSTM network with a pretrained BERT-large model [6].

Similar to [28], we apply the transfer learning paradigm to further enhance the model capability. The model is first pretrained on the VQAv2 dataset [10] and Visual Genome dataset [18]. To prevent data contamination, we remove those samples from the pretraining dataset, whose images are used in the testing split of OK-VQA. After that, the pretrained model is further finetuned on the training split of OK-VQA to obtain our final VQA model. Note that the answer vocabulary of the pretrained model (with 3,129 answers) is quite different from the vocabulary of OK-VQA. To bridge this gap, we merge the answer vocabulary of OK-VQA[1] with the existing vocabulary, resulting in an expanded answer vocabulary with 4,477 answers for model finetuning. This model is trained on a *single* Nvidia RTX 3090 GPU, which is affordable for most people.

To show the improvements of the above strategies over the original MCAN model, we report the accuracies on the testing set of OK-VQA as follows:

from scratch, original model [46]	from scratch, improved model	transfer learning, improved model
31.5	35.6	53.0

More details are provided in the supplementary material.

During the prompting stage, we follow PICa to use OSCAR+ as the captioning model [49]. Unless otherwise noted, we set the number of answer candidates K=10, the number of in-context examples N=16, and the number of queries T=5 as our default settings. The version of GPT-3 used in our experiments is `text-davinci-002`. Sampling temperature is set to 0.

The settings and strategies for OK-VQA can be directly transferred to A-OKVQA to address its DA task. For the MC task, we follow the strategy in [32] to project the predicted answer to the nearest answer choice. Moreover, we design a Prophet variant for the MC task. It uses a slightly different prompt by adding the multiple choices to in-context examples and testing input, and instructs GPT-3 to *choose* the correct one from four choices.

4.3. Ablation Studies

We conduct ablation experiments for Prophet on OK-VQA using the default settings above. Results shown in

[1] Similar to [2], we collect answers that appear more than eight times in the training set of OK-VQA, resulting in 2,794 answers.

VQA model, paradigm	stage-1 acc.	accuracy
ViLBERT, retrieval [42]	35.20	40.28 (+5.08)
ViLBERT, prompt†	35.28	44.97 (+9.69)

(a) **Prompting *vs*. retrieval.** Our prompting-based paradigm is more effective than the retrieval-based paradigm in MAVEx [42]. †: our re-implementation.

visual features	stage-1 acc.	accuracy
Bottom-Up [2]	46.83	55.34 (+8.51)
VinVL [49]	47.88	56.23 (+8.35)
CLIP-ViT-L/14 [31]	52.03	60.12 (+8.09)
CLIP-RN50×64 [31]	**53.04**	**60.84 (+7.80)**

(b) **Capability of VQA models.** More powerful VQA models lead to higher accuracies, but obtain slightly less relative improvements from stage-2.

#candidates (K)	hit rate	accuracy
0	-	49.63
1	53.04	56.04
5	75.20	60.17
10	**79.83**	**60.84**

(c) **Answer candidates.** They are critical to Prophet, and increasing the number K leads to better performance.

example selection	hit rate	accuracy
(a) rand	5.31	58.66
(b) ques + img [43]	59.58	59.82
(c) fused	**83.63**	**60.84**
(d) fused + ques + img	82.45	60.38
(e) answer logits	79.25	60.40

(d) **Example selection strategy.** Our answer-aware example selection based on fused features is more effective than the others.

#examples (N)	accuracy (T=1)	accuracy (T=5)
0	49.97	49.97
1	54.89	56.75
8	57.49	59.91
16	57.52	60.84
20	57.91	**61.10**

(e) **Numbers of examples and queries.** Increasing the numbers of examples N and queries T improves the performance with linearly increasing overheads.

variants	accuracy
(a) default	**60.84**
(b) w/o prompt head	60.54
(c) w/o confidence scores	55.46
(d) w/o image captions	58.27
(e) default+tags [43]	60.51

(f) **Prompt contents.** The default settings contain the exact necessary information for prompting.

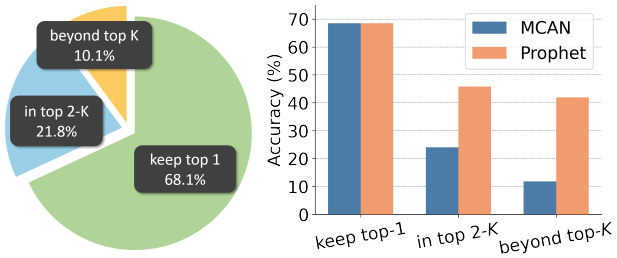
Table 1. **Ablation experiments for Prophet**. All the reported results are evaluated on the testing set of OK-VQA v1.1. The best result in each table is bolded and the result with the default settings is marked in gray.

Table 1 and Figure 3 are discussed in detail below.

Prompting *vs*. retrieval. Prophet uses a prompting-based paradigm to predict the answer based on a set of promising answer candidates. In contrast, a previous work MAVEx [42] exploits answer candidates but adopts a retrieval-based paradigm to search knowledge from external KBs to determine the answer. As both Prophet and MAVEx train a VQA model to generate answer candidates (stage-1), we can compare the superiority of the two paradigms (stage-2). In Table 1a, we show the performance of the two paradigms in terms of stage-1 accuracy and final accuracy, respectively.

For a fair comparison, we re-implement the VQA model used in MAVEx, *i.e.*, ViLBERT [25], to generate answer heuristics for our Prophet. From the results, we can see that based on the same VQA model, our Prophet outperforms MAVEx by a large margin (44.97% *vs*. 40.28%), showing the superiority of our prompting-based paradigm over MAVEx's retrieval-based paradigm in external knowledge acquisition and integration.

Capability of VQA models. In Table 1b we study how the VQA models of different capabilities impact the performance of Prophet. To better control the model capability, we use the same MCAN model trained with four visual features: region-based Bottom-Up [2] and VinVL [49] features and grid-based CLIP features from two backbones (ViT-L/14 and RN50×64) [31]. Results show that more powerful VQA models (reflected in the stage-1 accuracies) lead to better performance of Prophet, as they provide answer heuristics of higher quality. Combining the results in Table 1a, we also observe that more powerful VQA models achieve less relative improvements from GPT-3, which can be explained by the intrinsic diminishing return property. As a by-product, we verify that the visual features are important to the performance of knowledge-based VQA, which is consistent with the observations in [22]. The models with CLIP-based visual features significantly outperform those with region-based features, indicating that the CLIP's visual features contain richer visual knowledge due to large-scale pretraining.



(a) behavior distribution (b) per-type accuracy

Figure 3. We conduct a statistical analysis of Prophet's prediction behaviors in terms of (a) distribution and (b) per-type accuracy. As Prophet takes K answer candidates from MCAN as inputs, we define three prediction behaviors for Prophet as follows: "keep top-1", "in top 2-K", and "beyond top K". All the testing samples can be categorized into one of the three classes.

We have observed a significant performance improvement of Prophet over its corresponding MCAN model in stage-1 (60.84% *vs*. 53.04%). To better understand this improvement, we conduct a statistical analysis of Prophet's prediction behaviors. As Prophet takes K answer candidates from MCAN as inputs, we define three prediction behaviors for Prophet: "keep top-1", "in top 2-K", and "beyond top-K". All the testing samples can be categorized into one of the three classes. The statistical results in Figure 3 show that: (i) for 68.1% of the testing samples (the green slice), Prophet keeps the top-1 predictions of MCAN. These samples achieve a 69% accuracy and are

mostly easy samples; (ii) for 21.8% of the testing samples (the blue slice), Prophet selects answers from the top 2-K answer candidates. These samples are relatively hard, so that MCAN delivers a 24% accuracy while Prophet has a much higher 40% accuracy; (iii) for the remaining 10.1% of the testing samples (the yellow slice), Prophet predicts answers beyond the answer candidates[2]. These are the most difficult samples that MCAN delivers a 12% accuracy while Prophet magnificently achieves a 42% accuracy. In a word, Prophet acts like a real *prophet* that adaptively selects the essence and discards the dross from MCAN.

Answer candidates. Table 1c varies the number of answer candidates K from 0 to 10 to explore its effect on Prophet. For each testing sample, if the ground-truth answer is hit by one of the K answer candidates, we accumulate the soft score of that ground-truth answer[3]. The hit rate is calculated over the testing set by dividing the accumulated score by the number of samples.

From the results, we can see that: (i) without any answer candidates, Prophet's accuracy drops by 6.4 points (K=0 *vs*. K=1), showing the importance of answer candidates in Prophet; (ii) with the increase of answer candidates, the hit rate and final accuracy grow accordingly but they exhibit a tendency to saturate. This is because the quality of answer candidates eventually meets saturation as K increases; (iii) when K=1, the final accuracy is even higher than the hit rate (56.04% *vs*. 53.04%), which implies that GPT-3 has a strong capability to correct the wrong answer candidates while keeping the correct ones.

Example selection strategy. To show the effectiveness of our answer-aware example selection strategy, we compare it to other example selection strategies in Table 1d. The compared strategies include: (a) *rand*: examples that are randomly selected; (b) *ques + img*: examples that are selected based on the joint similarity of question and image features, which is used in PICa; (c) *fused*: our default strategy that selects examples based on the similarity of fused features; (d) *fused + ques + img*: a combination of our default strategy and PICa's strategy; and (e) *answer logits*: examples that are selected based on the similarity of answer logits obtained in Eq.(3). Besides the final accuracy, we also report the hit rate of the answers within the selected examples for each strategy.

The results show that the accuracy is positively correlated with the hit rate of answers, which verifies our hypothesis that answer-aware examples contribute significantly to the performance of Prophet. Compared with other strategies, our default strategy (c) achieves the best performance with the highest hit rate. The strategy (d) that integrates other information (ques + img) into the (c) leads to worse performance due to the introduction of irrelevant and noisy information. Finally, strategy (e) reports slightly worse performance than (c). We conjecture that is because the answer logits have lost too much information of the input question and image, which is also useful for GPT-3 to perform knowledge reasoning.

Numbers of examples and queries. Table 1d contains the ablation studies for the numbers of examples and queries. We choose different numbers of examples $N \in \{0, 1, 8, 16, 20\}$ for each query and different numbers of queries $T \in \{1, 5\}$, respectively. The results show that the performance of Prophet improves with the increase of N and T, which is consistent with the results in PICa. By increasing T from 1 to 5, the entries with larger N enjoy greater performance improvements at the expense of linearly increasing overheads.

Interestingly, the Prophet variant with N=0 delivers worse performance than the VQA model in stage-1 (49.97% *vs*. 53.04%), even though answer candidates are provided. Meanwhile, when given one example (N=1), the Prophet variant distinctly surpasses the VQA model (56.75% *vs*. 53.04%). This suggests the necessity of few-shot in-context examples for GPT-3 to activate its capability to adapt to the knowledge-based VQA task.

Prompt contents. In Table 1f, we ablate the prompt contents in the default settings by: (b) removing the prompt head; (c) removing the confidence scores for answer candidates; (d) removing image captions; and (e) adding predicted tags from external models [43].

The results lead to the following observations: First, the confidence scores are of critical importance to the performance of our Prophet. This is because they carry the necessary information for GPT-3 to understand the answer candidates. Second, without image captions, Prophet still works steadily. This reflects the fact that our answer heuristics in prompts already provide sufficient information for Prophet to solve the task. Third, the prompt head is of less importance, indicating that GPT-3 is capable of understanding the task directly from the in-context examples. Finally, introducing extra information like object tags leads to a slight performance drop, which is contrary to the results in PICa. We conjecture this information has already been encoded in answer heuristics implicitly.

4.4. Main Results

We use most of the default settings for the comparisons below, except that the number of examples N is set to 20.

Comparative results on OK-VQA. Table 2 contains the comparisons of our Prophet and existing state-of-the-art methods on OK-VQA. The table is split into three sections.

[2]The probability that Prophet's prediction is constituted of the combination of candidates is rare that can be neglected.

[3]In practice, multiple ground-truth answers are provided. If multiple answers are hit simultaneously, we choose the answer with the largest soft score for accumulation.

method	accuracy
methods with external knowledge bases	
Mucko [51]	29.2*
ConceptBERT [9]	33.7*
KRISP [28]	38.9
Visual Retriever-Reader [27]	39.2
MAVEx [42]	40.3
TRiG [8]	49.4
UnifER [12]	42.1
methods with multimodal pretraining	
Unified-IO (2.8B) [26]	54.0
Flamingo (80B) [1]	57.8
methods with GPT-3 API	
PICa [43]	48.0
KAT† [11]	53.1
REVIVE† [22]	56.6
Prophet (ours)	**61.1**

Table 2. **Comparisons to the state-of-the-art methods on OK-VQA**. The compared methods are split into three groups based on their knowledge resources and usages. *: accuracy is evaluated on OK-VQA v1.0. †: method needs to query GPT-3 during training.

The first section lists the retrieval-based methods leveraging external KBs [8, 9, 27, 28, 42, 51]. The second section contains the methods that are directly pretrained on a large-scale multimodal corpus [1, 26]. The last section shows the methods that incorporate the large language model GPT-3, which is publicly available via an online API [11, 22, 43].

Our Prophet belongs to the last section. It outperforms all the compared methods by a distinct margin. Prophet is 13.1 points higher than PICa [43] when both methods use GPT-3 as the only knowledge resource. This confirms our hypothesis that the capacity of GPT-3 has not been fully activated in previous studies. Compared to KAT [11] and REVIVE [22], which utilize GPT-3 and other external KBs together in sophisticated systems, our Prophet is much simpler and more effective. Moreover, KAT and REVIVE need to use GPT-3 to process all the training samples for their model training, which significantly increases the costs. In contrast, our Prophet only uses GPT-3 at inference time, which is more economical. Compared to the Flamingo-80B [1], Prophet delivers a 3.3 point improvement and is more resource-efficient from the perspective of reproducibility[4].

Comparative results on A-OKVQA. Table 3 contains the comparative results on the challenging A-OKVQA dataset. We compare our Prophet to the strong baselines in [32] and the current state-of-the-art method Unified-IO [26]. The results show the superiority of our Prophet over the counterparts on both the DA and MC tasks, reflecting the effectiveness and generalization of our method. Furthermore, we also provide a Prophet variant called Prophet-MC, which is specifically designed for the MC task. Specifically, we slightly modify the prompt in Prophet by adding the information of multiple choices into the in-context examples and testing input, and instruct GPT-3 to *choose* the correct one from four choices. More details are provided in the supplementary material. Compared to the original Prophet, Prophet-MC achieves significantly higher accuracy on the MC task, showing the enormous potential of Prophet to be applied to other related tasks.

[4]Flamingo-80B is trained on 1,536 TPUv4 for 15 days which is unaffordable for most researchers, but Prophet uses one RTX-3090 to train a VQA model for 4 days and a certain number of GPT-3 invocations.

method	DA		MC	
	val	test	val	test
ClipCap [32]	30.9	25.9	56.9	51.4
ViLBERT [32]	30.6	25.9	49.1	41.5
LXMERT [32]	30.7	25.9	51.4	41.6
KRISP [32]	33.7	27.1	51.9	42.2
GPV-2 [32]	48.6	40.7	60.3	53.7
Unified-IO [26]	-	45.2	-	-
Prophet	**58.2**	**55.7**	59.3	57.3
Prophet-MC	-	-	**76.4**	**73.6**

Table 3. **Comparisons to previous results on A-OKVQA**. DA and MC refer to the direct-answer and multiple-choice tasks, respectively. Prophet-MC is a variant of Prophet that is specifically designed for the MC task.

5. Conclusion

We present Prophet, a conceptually simple framework which uses GPT-3 as the knowledge engine for knowledge-based VQA. To better activate the few-shot learning capacity of GPT-3, we introduce a novel paradigm to prompt GPT-3 with answer heuristics. Extensive ablations, comparative experiments, and comprehensive analyses on two challenging datasets show the superiority of Prophet over all existing state-of-the-art methods, including the heavily-engineered Flamingo-80B model. Notably, Prophet is implemented with limited resources—a single GPU and an affordable number of GPT-3 invocations. We hope that our work will serve as a solid baseline to inspire future research on the knowledge-based VQA task and beyond.

Acknowledgment

This work was supported in part by the National Key R&D Program of China (2020YFB1406701), in part by the NSFC (62125201), in part by the Fundamental Research Funds for the Provincial Universities of Zhejiang (GK229909299001-001), in part by the NSFC (62072147, 62020106007, 61836002), and in part by the Zhejiang Provincial Natural Science Foundation of China (LR22F020001, DT23F020007).

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 1, 2, 3, 8

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, pages 6077–6086, 2018. 2, 5, 6

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, volume 33, pages 1877–1901, 2020. 1, 2

[4] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, pages 104–120, 2020. 2

[5] Yuhao Cui, Zhou Yu, Chunqi Wang, Zhongzhou Zhao, Ji Zhang, Meng Wang, and Jun Yu. Rosita: Enhancing vision-and-language semantic alignments via cross-and intra-modal knowledge integration. In *ACM MM*, pages 797–806, 2021. 2

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019. 3, 5

[7] Yang Ding, Jing Yu, Bang Liu, Yue Hu, Mingxin Cui, and Qi Wu. Mukea: Multimodal knowledge extraction and accumulation for knowledge-based visual question answering. In *CVPR*, pages 5089–5098, 2022. 2

[8] Feng Gao, Qing Ping, Govind Thattai, Aishwarya Reganti, Ying Nian Wu, and Prem Natarajan. Transform-retrieve-generate: Natural language-centric outside-knowledge visual question answering. In *CVPR*, pages 5067–5077, 2022. 2, 8

[9] François Gardères, Maryam Ziaeefard, Baptiste Abeloos, and Freddy Lecue. Conceptbert: Concept-aware representation for visual question answering. In *EMNLP*, pages 489–498, 2020. 8

[10] 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 *CVPR*, 2017. 5

[11] Liangke Gui, Borui Wang, Qiuyuan Huang, Alex Hauptmann, Yonatan Bisk, and Jianfeng Gao. Kat: A knowledge augmented transformer for vision-and-language. In *NAACL*, 2021. 1, 2, 3, 8

[12] Yangyang Guo, Liqiang Nie, Yongkang Wong, Yibing Liu, Zhiyong Cheng, and Mohan Kankanhalli. A unified end-to-end retriever-reader framework for knowledge-based vqa. In *ACM MM*, pages 2061–2069, 2022. 8

[13] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *ICCV*, pages 804–813, 2017. 2

[14] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pages 6700–6709, 2019. 2

[15] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. In *CVPR*, pages 10267–10276, 2020. 2

[16] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, pages 2901–2910, 2017. 2

[17] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. In *NeurIPS*, volume 31, 2018. 2

[18] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017. 5

[19] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 2

[20] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering. In *ICCV*, pages 10313–10322, 2019. 2

[21] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, pages 121–137, 2020. 2

[22] Yuanze Lin, Yujia Xie, Dongdong Chen, Yichong Xu, Chenguang Zhu, and Lu Yuan. REVIVE: Regional visual representation matters in knowledge-based visual question answering. In *NeurIPS*, 2022. 1, 2, 3, 6, 8

[23] Hugo Liu and Push Singh. Conceptnet: a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004. 2

[24] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? In *DeeLIO@ACL*, pages 100–114. Association for Computational Linguistics, 2022. 4

[25] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, volume 32, 2019. 2, 6

[26] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. *arXiv preprint arXiv:2206.08916*, 2022. 8

[27] Man Luo, Yankai Zeng, Pratyay Banerjee, and Chitta Baral. Weakly-supervised visual-retriever-reader for knowledge-based question answering. In *EMNLP*, pages 6417–6431, 2021. 2, 8

[28] Kenneth Marino, Xinlei Chen, Devi Parikh, Abhinav Gupta, and Marcus Rohrbach. Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa. In *CVPR*, pages 14111–14121, 2021. 2, 5, 8

[29] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *CVPR*, pages 3195–3204, 2019. 1, 2, 5

[30] Medhini Narasimhan, Svetlana Lazebnik, and Alexander Schwing. Out of the box: Reasoning with graph convolution nets for factual visual question answering. In *NeurIPS*, volume 31, 2018. 2

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 5, 6

[32] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *ECCV*, pages 146–162. Springer, 2022. 1, 2, 5, 8

[33] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? In *ICLR*, 2022. 2

[34] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019. 2

[35] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. In *NeurIPS*, volume 34, pages 200–212, 2021. 3

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, volume 30, 2017. 2

[37] Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014. 2

[38] Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton Van Den Hengel. Fvqa: Fact-based visual question answering. *IEEE TPAMI*, 40(10):2413–2427, 2017. 1, 2

[39] Peng Wang, Qi Wu, Chunhua Shen, Anthony R Dick, and Anton van den Hengel. Explicit knowledge-based reasoning for visual question answering. In *IJCAI*, 2017. 1, 2

[40] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. OFA: unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *ICML*, pages 21218–23340, 2022. 2

[41] Wenhui Wang, Hangbo Bao, Li Dong, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. In *NeurIPS*, 2021. 1, 2

[42] Jialin Wu, Jiasen Lu, Ashish Sabharwal, and Roozbeh Mottaghi. Multi-modal answer validation for knowledge-based vqa. In *AAAI*, pages 2712–2721, 2022. 2, 6, 8

[43] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *AAAI*, pages 3081–3089, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[44] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *TMLR*, 2022. 2

[45] Zhou Yu, Yuhao Cui, Jun Yu, Meng Wang, Dacheng Tao, and Qi Tian. Deep multimodal neural architecture search. In *ACM MM*, pages 3743–3752, 2020. 2

[46] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *CVPR*, pages 6281–6290, 2019. 5

[47] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *ICCV*, pages 1821–1830, 2017. 2

[48] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 1, 2

[49] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *CVPR*, pages 5579–5588, 2021. 2, 5, 6

[50] Mingyang Zhou, Licheng Yu, Amanpreet Singh, Mengjiao Wang, Zhou Yu, and Ning Zhang. Unsupervised vision-and-language pre-training via retrieval-based multi-granular alignment. In *CVPR*, pages 16485–16494, 2022. 2

[51] Zihao Zhu, Jing Yu, Yujing Wang, Yajing Sun, Yue Hu, and Qi Wu. Mucko: Multi-layer cross-modal knowledge reasoning for fact-based visual question answering. In *IJCAI*, pages 1097–1103, 2020. 2, 8