Retrieve, Summarize, Decide: A Two-Stage Recipe for Knowledge Based Visual Question Answering

Xueyan Li, Weizhe Lin, Bill Byrne
University of Cambridge

Abstract

Knowledge-based visual question answering (KBVQA) requires reasoning over an image together with knowledge that cannot be inferred from the image alone. Prior work typically leans on one of three tools, large vision–language models (VLMs), external document retrieval, or in-context prompting of frozen LLMs. Each of these methods have limitations when used in isolation. We present a two-stage framework that unifies all three. In Stage 1, we augment retrieval-augmented VQA with question-aware visual encoding (InstructBLIP) and train the retriever and generator jointly to produce *document-wise* answer candidates, which are concise, question-focused summaries grounded in retrieved passages. In Stage 2, we prompt a separate frozen LLM with few-shot examples that combine question-aware captions (PromptCap) and the Stage 1 candidates, allowing the model to either select a grounded candidate or fall back to its implicit knowledge when appropriate. On OKVQA, our Stage 1 model reaches 62.83% VQA score, surpassing prior retrieval-based systems, and our Stage 2 method attains 61.69% with only 5-shot prompting and no ensembling, outperforming comparable in-context approaches.

1 Introduction

Knowledge-based visual question answering (KB-VQA) extends traditional VQA by requiring knowledge that is rarely visible in the image alone. Effective systems must integrate (i) strong visual grounding, (ii) access to external knowledge, and (iii) robust language reasoning. In practice, two complementary sources of knowledge are commonly exploited: *implicit* knowledge internalized during large-scale pre-training of language or vision–language models, and *explicit* knowledge retrieved from external corpora and then consumed at inference time (Gao et al., 2024).

A standard retrieval-augmented VQA pipeline proceeds in two steps (Luo et al., 2021; Lin and Byrne, 2022): first, given an image and question, the system retrieves passages from a large knowledge base (e.g., Wikipedia) or the open web (Gao et al., 2022; Lin and Byrne, 2022; Luo et al., 2021); second, a generator conditions on the retrieved text to produce an answer. RA-VQA (Lin and Byrne, 2022) instantiates this paradigm with joint training of a dense retriever and an answerer, but relies on *text-based vision* (generic captions, OCR, and object tags) as a surrogate for the image. Because these textual surrogates are typically question-agnostic, they can be verbose or irrelevant, and may hinder retrieval and generation quality when the question depends on specific visual details.

A complementary line of work avoids explicit retrieval and instead prompts frozen LLMs using in-context examples, banking on the LLM's implicit world knowledge. PromptCap (Hu et al., 2023) trains OFA (Wang et al., 2022) to generate *question-aware* captions that better expose salient visual content to a text-only LLM (GPT-3; Brown et al., 2020). Prophet (Shao et al., 2023) follows a similar few-shot recipe but adds *answer candidates* sourced from a VQA classifier (MCAN-large trained with Oscar+ captions; Yu et al., 2019; Li et al., 2020). However, generic captions may omit the question's decisive visual cue, and candidate lists derived from a fixed VQA label space can be poorly aligned with open-ended answers, especially without an explicit, document-grounded evidence trail.

We combine the strengths of these strands in a two-stage framework that unifies a question-aware VLM, retrieval, and in-context prompting. **Stage 1** augments RA-VQA with question-aware visual encoding to strengthen grounding and evidence selection. The model retrieves passages and generates *document-wise* predictions which are concise, question-focused answers tied to specific sources. **Stage 2** constructs few-shot exemplars that pair PromptCap-style question-aware captions (Hu et al., 2023) with the Stage 1 document-wise candidates,

and prompts a frozen LLM to output a single short answer—either selecting a grounded candidate or, when evidence is insufficient or conflicting, leveraging implicit knowledge (Brown et al., 2020).

On OKVQA, this coupling yields state-of-the-art results among models of comparable sizes: our Stage 1 system attains 62.83% VQA score, and the Stage 2 in-context method reaches 61.69% with only five shots and no ensembling, outperforming prior in-context approaches such as Prophet (61.10% with 20-shot + 5-fold ensemble) (Shao et al., 2023) and PromptCap (60.4%) (Hu et al., 2023). Beyond accuracy, we find that (i) injecting question-aware visual features improves retrieval relevance and per-document answerability compared to text-only surrogates (Lin and Byrne, 2022), and (ii) candidate+caption exemplars help frozen LLMs balance explicit, verifiable evidence with implicit knowledge, while keeping prompts compact. Our main contributions are:

- A retrieval-augmented VQA model that integrates question-aware visual encoding into the RA-VQA paradigm, improving both retrieval and grounded answer generation.
- A lightweight in-context prompting recipe that fuses document-grounded candidates with question-aware captions to guide frozen LLMs.
- Empirical results on OKVQA demonstrating competitive or superior performance to prior retrieval-based and in-context baselines, with small prompt budgets and no ensembling.

2 Related Work

2.1 In-context learning for VQA

In-context learning uses frozen LLMs on new tasks without gradient updates by supplying carefully constructed examples. For KBVQA, two representative approaches are Prophet (Shao et al., 2023) and PromptCap (Hu et al., 2023), both of which rely on GPT-3's pre-trained, implicit knowledge (Brown et al., 2020).

Shao et al. (2023) augments few-shot prompts with *answer candidates* predicted by a VQA classifier (MCAN-large) trained on VQAv2 (Goyal et al., 2017) and Visual Genome (Li et al., 2019), using Oscar+ captions (Li et al., 2020) as text-only surrogates of the image. In-context exemplars are selected by CLIP similarity (Radford et al., 2021). Framing answer prediction as classification over a fixed vocabulary limits open-ended performance as valid strings unseen during training will not be considered. Prophet also depends on ensembling (prompting GPT-3 multiple times and majority-voting), with notable degradation without it. Finally, generic captions can be question-agnostic and visually underspecified.

PromptCap (Hu et al., 2023) addresses the caption quality issue by training OFA (Wang et al., 2022) to produce *question-aware* captions that better expose salient visual cues to a text-only LLM. GPT-3 is then prompted with these captions and few-shot exemplars to generate final answers. However, because PromptCap does not surface explicit external evidence or candidate sets, it leans even more heavily on implicit knowledge.

Our Stage 2 combines the strengths of both: we keep Prophet's candidate-driven prompting but replace generic captions with question-aware ones (Hu et al., 2023), and we swap classifier-derived candidates for *document-grounded* candidates produced by a retrieval-augmented model (Stage 1), improving both relevance and coverage.

2.2 Combining implicit and explicit knowledge

A complementary line fuses external, *explicit* knowledge with LLMs' *implicit* priors. KAT (Gui et al., 2022) and REVIVE (Lin et al., 2022) curate Wikidata-derived graphs (Vrandečić and Krötzsch, 2014) relevant to OKVQA, retrieve entries aligned to image regions, and then condition a generator (e.g., T5 (Raffel et al., 2020)) on the concatenation of explicit facts, region features, and text prompts to produce answers. REVIVE further emphasizes object-centric retrieval and region-level grounding.

Beyond curated KBs, TRiG (Gao et al., 2022) employs dense passage retrieval (DPR; Karpukhin et al., 2020) over Wikipedia to fetch free-text evidence and then conditions generation on the retrieved passages. However, TRiG, KAT, and REVIVE can retrieve generic or only loosely relevant text when vision cues are weak or text surrogates are question-agnostic.

RA-VQA (Lin and Byrne, 2022) advances retrieval for VQA by jointly training the retriever and the generator end-to-end, improving the likelihood that at least one retrieved passage contains a gold answer span (e.g., PRRecall 96.55 with 50 docs vs. 85.56 for TRiG with 100 docs). Nonetheless, RA-VQA models the image via *text-based*

vision (captions, OCR, tags), which may ignore question-specific visual details and thereby limit both retrieval and answer generation.

Our Stage 1 follows the RA-VQA training paradigm but replaces text-only surrogates with *question-aware* visual encoding, strengthening grounding during retrieval and yielding *per-document* answers that we later reuse as high-quality, evidence-tied candidates in Stage 2.

3 Methods

Our pipeline has two stages that cleanly separate *grounded evidence construction* from *answer selection*. In **Stage 1** (Sec. 3.1), we augment RA-VQA with a *question-aware visual prefix*: image features are injected directly into the language space so the retriever and generator are trained jointly with stronger visual grounding. This stage retrieves passages and produces *document-wise* answers which are short, question-focused predictions conditioned on each retrieved document. In **Stage 2** (Sec. 3.2), we build few-shot exemplars that pair *question-aware captions* with the *document-wise candidates* from Stage 1, and prompt a frozen LLM to produce a single, short answer by either selecting a grounded candidate or falling back to implicit knowledge.

3.1 Incorporating a Question-Aware Visual Prefix

Prior RA-VQA systems represent the image via *text-based vision* (generic captions, OCR strings, object tags) (Lin and Byrne, 2022), which can be verbose and question-agnostic. We instead fuse *image embeddings* directly into the language model (LM) as a prefix that is *conditioned on the question*.

Notation. Let I denote the image and q the question text. We write F_V for the frozen vision encoder, F_Q for the Q-Former, F_M for a linear/projection mapping into the LM embedding space, and F_L for the LM's text encoder. We use d_v for the vision feature width, d_L for the LM embedding width, and L_v for the number of visual prefix tokens.

Visual prefix construction. We obtain an $L_v \times d_L$ matrix V that will be concatenated to the LM's text embeddings:

$$\tilde{V} = F_Q\big(F_V(I)\big) \in \mathbb{R}^{L_v \times d_v}, V = F_M(\tilde{V}) \in \mathbb{R}^{L_v \times d_L}.$$

For **BLIP-2** (Li et al., 2023), F_Q is trained via image–text contrastive learning, image-grounded generation, and matching, providing task-agnostic visual tokens suitable for language conditioning. For **InstructBLIP** (instruction-tuned BLIP-2), the Q-Former conditions on the question, producing a *question-aware* visual prefix:

$$\tilde{V} = F_Q\big(F_V(I), F_L(q)\big), \qquad V = F_M(\tilde{V}).$$

This conditioning makes the visual tokens selectively emphasize regions/features relevant to q.

Input packing and training objective. Following Lin and Byrne (2022), an instance may include the question q, optional text-based vision c (short caption/OCR), and (during retrieval-augmented training) a single document z_k. We pack the LM input embeddings as

$$X = \big[V : F_L(q) : F_L(c) : F_L(z_k)\big],$$

where [:] denotes concatenation (omitting c or z_k if absent). Let the answer sequence be $s^{1:T}$. We fine-tune the generator with standard left-to-right NLL:

$$\mathcal{L}_{\text{fine-tune}} = -\sum_{t=1}^{T} \log p\big(s^t \mid X, s^{<t}\big). \quad (1)$$

Intuitively, the question-aware prefix V shifts both *retrieval* (by improving the learned query representation over (I, q)) and *generation* (by injecting visual evidence into the decoder's context).

3.2 In-Context Learning

Stage 1 yields multiple per-document answers per question; some are correct and well-grounded, others noisy. Rather than fusing them inside a single model head, we delegate *final selection* to a stronger frozen LLM (e.g., GPT-3.5 / Flan-T5-XXL) that can weigh candidates against captions and, if needed, supply an answer from implicit knowledge. We keep the two-stage design lightweight and transparent.

Step 1: Document-wise candidate generation. Given a question $x = (I, q)$, the trained retriever returns top-K documents $\{z_1, \ldots, z_K\}$ with relevance scores $\{p_\theta(z_k \mid x)\}_{k=1}^{K}$ (higher implies more relevant). Conditioning on each z_k, the Stage 1 generator produces an answer y_k. Since different documents can yield identical strings, we deduplicate into a set of *distinct* candidates $\{y_1, \ldots, y_{K_{AC}}\}_{\neq}$ and keep the *best associated scores* (e.g., the maximum $p_\theta(z_k \mid x)$ among documents that generated the same string). We cap the list by $K_{AC} \leq K$—keeping the K_{AC} highest-scoring distinct strings. If fewer than K_{AC} unique candidates exist, we use all available ones.

Step 2: Few-shot exemplar construction. Each exemplar e contains: (i) a *question-aware caption* c (from PromptCap) that describes image content salient to q; (ii) the question q; (iii) the candidate list with their document scores $\{(y_i,\ p_\theta(z_i|x))\}_{i=1}^{K_{AC}}$; and (iv) the gold answer a for supervision in the exemplar. We use the Prophet-style schema (Shao et al., 2023):

```
Context: c
Question: q
Candidates: y_1 (p_θ(z_1|x)),..., y_{K_AC} (p_θ(z_{K_AC}|x))
Answer: a
```

A short *prompt head* precedes the exemplars to explain that candidates include confidence-like scores and that the true answer may be absent (as in Shao et al., 2023).

Selecting exemplars via FAISS. At test time, we build a feature z for the input (I, q) and retrieve the N most similar training features $\{z_i\}$ to form $\varepsilon = \{e_i\}_{i=1}^{N}$. We consider several choices for z:

$$z_q = F_L(q) \quad \text{(question only)},$$
$$z_{q,c} = [F_L(q) : F_L(c)] \quad \text{(question+caption)},$$
$$z_{q,I} = [F_L(q) : F_M(F_Q(F_V(I)))] \text{(question+emb)}.$$

Similarity is cosine or inner product; we retrieve $I = \underset{i}{\text{argTopN}}\ \text{sim}(z, z_i)$ and set $\varepsilon = \{e_i : i \in I\}$.

Prompting the frozen LLM. The final input prompt to a frozen LLM is the concatenation of the prompt head p, the N exemplars ε, and the test instance $(c, q, \{(y_i, p_\theta(z_i|x))\}_{i=1}^{K_{AC}})$. The frozen LLM is instructed to output a single short string. The desired behavior is:

- If a candidate is well-supported (high score and consistent with c/q), *select it*.
- If candidates are missing/contradictory, *answer from implicit knowledge*, guided by c.

This separates evidence construction (Stage 1) from answer selection (Stage 2), while keeping both steps interpretable: candidates are tied to documents via $p_\theta(z|x)$, and the LLM's choice can be audited against the provided evidence.

4 Experiments

Setup overview. We evaluate the two-stage pipeline by (i) assessing the effect of question-aware visual encoding within a retrieval-augmented VQA framework, and (ii) measuring how document-wise candidates and question-aware captions affect few-shot in-context performance. Unless stated otherwise, all results are on OKVQA using the standard VQA metric. Additional training details and hyperparameters are provided in Appendix A.

4.1 Evaluating Visual Encoders

We compare two visual encoders, **BLIP-2** and **InstructBLIP**, keeping both the visual encoders and the linear projection layer *frozen* during training. We first fine-tune on OKVQA *without documents* to isolate the effect of instruction-tuning in InstructBLIP. We then consider retrieval-augmented variants where the dense retriever is either frozen or trained jointly with the generator.

Frozen retriever (FrDPR). In this setting, only the mapping network and LM are trainable. The retriever consumes a representation of the *question plus text-based vision* (encoded by `BERT-base`) and returns $K_{\text{train}} = 5$ documents per instance. Each document z_k is concatenated with the visual prefix, the question, and the (short) text-based vision, encoded by the Flan-T5-XL encoder, and the generator produces a per-document answer. Because we process 5 documents in parallel, the effective batch is 5×. We therefore set the base batch size to 1 so the total token budget fits within memory.

Joint retriever–generator (RAVQA-style). Finally, we fine-tune the retriever together with the generator ("InstructBLIP-RAVQA" in the style of Lin and Byrne (2022)) to test whether a trainable retriever further boosts relevance and downstream VQA performance compared to FrDPR.

4.2 In-Context Few-Shot Learning

We take the best-performing retrieval-augmented checkpoint from above and use it to generate *document-wise* answer candidates for few-shot prompting. We study: (1) how the number of test-time retrieved documents K_{test} affects candidate diversity and quality; (2) the effect of question-aware captions (PromptCap; Hu et al., 2023) versus generic captions; (3) prompt-format ablations; and (4) the behavior across multiple LLM backbones (13B-class and GPT-3.5).

Answer candidate quality. Preliminary experiments indicated low diversity when retrieving few documents (e.g., $K_{\text{test}} = 5$ often yields only 1–2 distinct strings). We therefore increase K_{test} up to 50 to surface more distinct candidates. To avoid very

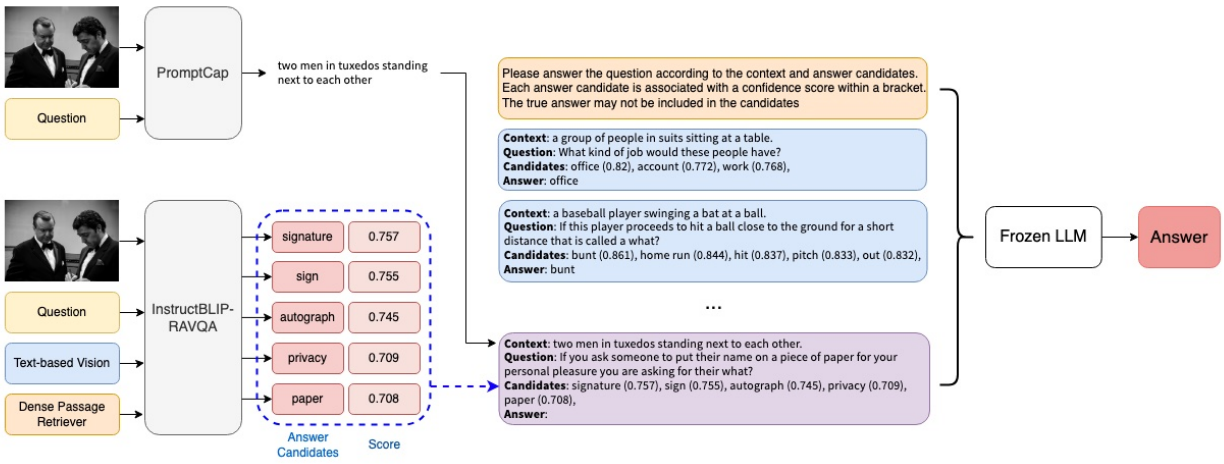

Figure 1: Two-stage pipeline. Stage 1 (left): retrieval with question-aware visual prefix and per-document generation. Stage 2 (right): few-shot prompting with question-aware captions and document-grounded candidates.

long lists, we cap the number of distinct strings at $K_{AC} \leq K_{\text{test}}$ by keeping the top-K_{AC} unique candidates with the highest associated document scores. If fewer than K_{AC} unique strings exist, we keep all available candidates. We evaluate both *document relevance* and *candidate quality* (Sec. 4.5) and also vary K_{AC} to inspect the trade-off between coverage and noise.

4.3 Image Caption Generation

We compare three sources of text-based vision to pair with candidates inside exemplars:

1. **Oscar+ captions** (Li et al., 2020), as used by RA-VQA (Lin and Byrne, 2022) and Prophet (Shao et al., 2023).
2. **BLIP-2 captions**, generated with the OPT-2.7B variant (best COCO captioning within BLIP-2; Li et al., 2023).
3. **PromptCap captions** (Hu et al., 2023), which are *question-aware* and emphasize details salient to q.

We produce captions for both OKVQA train and validation images and measure downstream differences in few-shot performance.

4.4 In-Context Hyperparameter Tuning

We select exemplars via a FAISS index (Johnson et al., 2021). We build indexes over several feature choices to probe which signal best predicts transfer: (i) question-only $F_L(q)$; (ii) question+caption $[F_L(q) : F_L(c)]$; and (iii) question+visual $[F_L(q) : F_M(F_Q(F_V(I)))]$. Text encodings (Flan-T5-XL) are padded to 32×1024 to match visual token shapes. Question-aware visual tokens are produced by InstructQFormer. We then retrieve the top-N nearest training features (inner product or cosine, as in Sec. 3.2) to form the in-context examples set.

Backbones for few-shot prompting. Beyond Flan-T5-XXL, we evaluate 13B-class models (LLAMA 2 and LLAMA 2-Chat (Touvron et al., 2023), Vicuna (Chiang et al., 2023)) and two GPT-3.5 variants: `gpt-3.5-turbo` (chat-optimized) and `text-davinci-003` (instruction-following). Our analysis focuses on (i) sensitivity to K_{AC} and N shots and (ii) the frequency with which models *select* a candidate versus *generate* an alternative from implicit knowledge.

4.5 Evaluation Metrics

VQA Score. Overall task accuracy on OKVQA is measured using the standard VQA metric (Goyal et al., 2017).

Pseudo Relevance Recall (PRRecall)@K. Because candidates are generated per retrieved document, we quantify how often at least one of the top-K documents contains a gold answer string, following Lin and Byrne (2022). Let S denote the set of human-annotated answers for a question and define a pseudo-relevance function $H(z, S) = \mathbb{1}\{\text{document } z \text{ contains any } a \in S\}$. Then

$$\text{PRRecall@}K = \min\left(\sum_{k=1}^{K} H(z_k, S),\ 1\right). \quad (2)$$

Higher values indicate better evidence retrieval.

Table 1: Results achieved on OKVQA. Results from this paper are in bold. Encoding: image encodings from visual models. Text-based vision (TBV): text that describes the image. Google: Google search results.

Model	Visual Features	Knowledge Source	VQA Score
Visual-language model			
PaLM-E (562B)	Encoding	-	66.1
PaLI-X (55B)	Encoding	-	66.1
PaLI (17B)	Encoding	-	64.5
InstructBLIP (7B)	Encoding	-	62.1
PaLM-E (12B)	Encoding	-	60.1
InstructBLIP (4B)	Encoding+TBV	-	58.91
Retrieval based methods			
TRiG	TBV	Wikipedia	50.50
KAT (Ensemble)	TBV	GPT-3+Wikidata	54.41
RA-VQA	Encoding+TBV	Google	54.48
REVIVE (Ensemble)	Encoding+TBV	GPT-3+Wikidata	58.0
InstructBLIP-FrDPR	Encoding+TBV	Google	62.05
InstructBLIP-RAVQA ($K_{test} = 5$)	Encoding+TBV	Google	62.51
InstructBLIP-RAVQA ($K_{test} = 50$)	Encoding+TBV	Google	**62.83**
In-context prompting			
Prophet (20-shot, Single)	Caption	GPT-3	57.91
PromptCap (16-shot)	Caption	GPT-3	58.4
PromptCap (32-shot)	Caption	GPT-3	60.4
Prophet (20-shot, Ensemble)	Caption	GPT-3	61.10
Text-Davinci-003 (5-shot)	Caption	GPT-3.5+Google	60.24
GPT-3.5-Turbo (5-shot)	Caption	GPT-3.5+Google	61.11
Flan-T5-XXL (5-shot)	Caption	Google	**61.69**

Hit Rate. To upper-bound few-shot performance given a fixed candidate list, we follow Shao et al. (2023) and compute the best achievable VQA score if a model could always pick the most correct candidate. Given candidates $\{AC_c\}_{c=1}^{C}$,

$$\text{HitRate} = \max_{c \in \{1,\ldots,C\}} \text{VQAScore}(AC_c, S). \quad (3)$$

We also report HitRate as a function of K_{AC} to study the marginal utility of longer lists.

5 Results

Table 1 shows the results in this paper compared to state-of-the-art results in the OKVQA dataset. We divide the methods into three categories, visual-language models, document retrieval based methods and in-context prompting. Our InstructBLIP model without retrieval is smaller than other visual-language models but achieves similar performance. Once retrieval process is added, InstructBLIP-RAVQA improves by around 4% and outperforms other retrieval-based models. This shows that our method to incorporate visual encoders to RA-VQA is effective. Our in-context learning method also outperforms all other in-context learning based methods. The currently best performing model Prophet (Shao et al., 2023) uses 5-fold ensembling to achieve 61.10%. We surpass this performance with no ensembling.

5.1 Incorporating Vision Encoder

Table 2: Documents are more likely to include at least one ground truth answer when more documents are retrieved for testing K_{test}. InstructBLIP-RAVQA has better document retrieval performance than RAVQA from (Lin and Byrne, 2022). Training each model always uses $K_{train} = 5$.

K_{test}	RAVQA		InstructBLIP-RAVQA	
	PRRecall	VQA Score	PRRecall	VQA Score
5	82.84	53.81	84.78	62.51
20	93.62	54.2	93.62	62.69
50	96.47	54.45	97.05	62.83

Table 2 shows that InstructBLIP-RAVQA has higher PRRecall than RAVQA from (Lin and Byrne, 2022), meaning documents are more likely to contain ground truth answers. When the number of documents retrieved for testing is increased, PRRecall and VQA Score both increase, showing that using more documents makes it more likely for the model to find the correct answer amongst them. It is important to retrieve a larger number of documents not only for better answer generation, but to also

get a more diverse range of answer candidates for in-context learning later.

Table 3: Ablation results of fine-tuned models on various inputs. Full InstructBLIP checkpoint from Huggingface is used as training starting point. $K_{train} = K_{test} = 5$

Model	Question	Visual Prefix	Text-based Vision	Documents	VQA Score
Flan-T5-XL	✓				30.79
BLIP2	✓	✓			55.03
	✓	✓	✓		55.81
InstructBLIP	✓	✓			58.16
	✓	✓	✓		58.91
InstructBLIP-FrDPR	✓	✓	✓	✓	62.05
InstructBLIP-RAVQA	✓	✓	✓	✓	62.51

The results of the fine-tuned Flan-T5-XL models on various inputs are shown in Table 3. Training with a question and image-prefix has VQA Score 58.16. Adding text-based vision adds 0.75 points. FrDPR retrieves 5 documents, and results in further performance boost by 3.14. This is slightly lower than the same ablation experiment presented in (Lin and Byrne, 2022) that showed 5.06 improvement with the addition of frozen retriever. Similarly, improvement from FrDPR to RAVQA is 0.46, lower than 2.59 in (Lin and Byrne, 2022). However, considering the already high baseline performance in InstructBLIP, it becomes increasingly difficult to get further improvement. Thus, it is reasonable to get smaller improvement from the addition of documents.

Table 4: Answer candidates are more likely to contain at least one ground truth answer as the maximum number of distinct candidates increase. Additional candidates become less accurate with more than 10 candidates. The overall quality of answer candidates is a lot higher than that in Prophet (Shao et al., 2023).

K_{AC}	Training Set		Validation Set		
	ACRecall	Hit Rate	ACRecall	Hit Rate	Prophet Hit Rate
1	94.77	90.65	66.31	61.86	53.04
2	98.28	95.77	80.00	75.61	
5	99.23	97.41	87.67	83.73	75.20
10	99.36	97.65	89.50	85.57	79.83
15	99.39	97.68	89.67	85.76	
max	99.39	97.68	89.71	85.78	

The overall quality of answer candidates is a lot higher than that in Prophet (Shao et al., 2023). Prophet relies on implicit knowledge from GPT-3 and do not use any explicit knowledge source. In our case, answer candidates come from external documents. They are a form of question-aware document summary. Thus, they are a lot more relevant than that in Prophet. Since Hit Rate is 8.5 points higher than that of Prophet with 5 answer candidates (Table 4), it is expected that our better answer candidates directly result in better few-shot performance.

Note that the number of answer candidates in Prophet is different from in this paper. In Prophet, the number of answer candidates is always the same for each question. However, in this report, this number varies between questions and a maximum cap is imposed by K_{AC}.

5.2 Ablation on in-context experiments

Table 5: Ablation on components for in-context prompting. One variable is changed at a time compared to the baseline. The baseline model has prompt head, confidence scores, $K_{AC} = 5$, $N = 5$ and PromptCap captions. The LLM used is Flan-T5-XXL.

Variants	VQA Score
Baseline	61.16
(a) Without answer candidates or scores	32.50
(b) Without document confidence scores	57.17
(c) Without prompt head	59.24
(d) BLIP2 caption	59.55
(e) Oscar+ caption	60.72

Ablation experiments verify that each component of in-context examples are necessary. The baseline prompt is designed based on Prophet (Shao et al., 2023) results that showed prompt head, confidence scores and image captions are integral parts of in-context examples. Performance degrades without them . We reach similar conclusions with Flan-T5-XXL in Table 5. A very large performance drop is observed without answer candidates (28.66%). A small performance drop is observed without prompt head (1.92%). A larger performance drop is observed without document confidence scores (3.99%) showing that relevance of documents measured by the retriever indeed provide useful information during in-context learning for the LLM to make a judgement on which answer candidate to use.

We test various LLMs with 13 billion parameters. Table 6 shows that the best LLM tested is Flan-T5-XXL. LLAMA2 performs worse, despite being released the most recently with claims of being a 'suitable alternative for closed-source models'.

Investigation into the generated answers shows that LLAMA2 models interestingly continue generating after giving the answer (not shown). Since in-context examples end with the gold answer, and

Table 6: Comparison of various LLMs of similar sizes for in-context learning. Post-processing has been applied to remove trailing sentences and punctuation.

LLM	VQA Score
Flan-T5-XXL (11B)	61.16
LLAMA2-13B	55.65
LLAMA2-13B-chat	55.92
Vicuna-13B	41.15

start with the next image caption, LLAMA2 fails to recognize that it should stop generating after answering, but follows the in-context pattern to generate the next caption. Thus, post-processing had to be applied to only extract only the answer. Additionally, we observe that LLAMA2-chat frequently put a period or line break symbol \n after the answer, despite none of in-context examples having those symbols after answers. This is likely due to the dialogue-based data that was used for training. Turn-based chat data end with periods or line breaks so this behavior is retained in in-context learning. This is undesirable since it shows that LLAMA2-chat fails to learn the expected answer format shown in in-context examples. Again, post-processing had to be done to remove periods.

Table 7: The effect of increasing the number of answer candidates K_{AC} on few-shot model performance.

Model	K_{AC}	VQA Score
Flan-T5-XXL	3	61.69
	Baseline	61.16
	10	60.96
	15	61.05
GPT-3.5-Turbo	Baseline	59.60
	15	59.91

Table 7 shows the effect of including more answer candidates in in-context examples. For Flan-T5-XXL, when K_{AC} is increased, performance degrades. This shows that Flan-T5-XXL is not capable of making the right choice from more answer candidates. This implies that better performance can be obtained when the model only chooses from the first few answer candidates. In contrast, GPT-3.5-Turbo yields improved performance with more answer candidates, showing its superior reasoning ability. It can distinguish between good and bad answer candidates and make appropriate choices.

Table 8 shows the effect of increasing number of in-context examples (shots) on model performance. For Flan-T5-XXL, few-shot performance

Table 8: The effect of increasing the number of in-context examples N.

Model	N	VQA Score
Flan-T5-XXL	Baseline	61.05
	10	61.18
	20	61.07
GPT-3.5-Turbo	Baseline	59.91
	20	60.03

is the best with 10 shots. This is likelu due to context window being 512 tokens during training (Chung et al., 2022). Thus, if the input sequence is longer than 512, Flan-T5-XXL might ignore the content in the middle of the input sequence. This phenomenon is explained in (Liu et al., 2023) where encoder-decoder models have difficulty accessing information in the middle of long sequences when the length of the sequence is longer than the model's pre-training context window. The input length is around 500 tokens with 10 shots, which fit Flan-T5-XXL's context window. This explains why 10 shots have the best in-context performance.

GPT-3.5-Turbo allows a maximum of 4096 tokens in the input. It is unclear how big its actual context window is, but it's likely to be much bigger than that of Flan-T5-XXL. Thus, there is no performance degrade with 20-shots for GPT-3.5-Turbo. At the same time, very little improvement is seen with more shots (improve by 0.12). This is similar to the result in Prophet (Shao et al., 2023) that showed improvement by 0.42 from 8 to 20 shots. We hypothesize that GPT-3.5 was able to learn the in-context examples' pattern and the expected answer format with a small number of shots. Thus, more shots do not result in big performance gain.

5.3 Answer Diversity Limitations

Table 9 shows an example of answer candidates for a question about the founding date of the Coca-Cola brand. The model generates the same answer for each document, despite the documents containing different possible years. The answer 1886 is not among ground truth answers. However, it is almost correct since 1886 is the date that the drink was first invented, rather than when the brand was founded. It is likely that the model ignored the documents, and used its implicit knowledge instead to generate the answer. The ideal behavior would be to generate different answers that correspond to the years mentioned in the document.

An explanation for InstructBLIP-RAVQA ignor-

Table 9: Examples that demonstrate the lack of diversity in answer generation. Documents are shortened to contain only the parts that include numbers. Question: When was the cola brand on the signs founded? Ground truth annotations: 1892, 1892, 1892, 1892, 1851,1851, 1870's, 1870's, 1800, 1800

Prediction	Document Score	Document
1886	0.858	history of coca cola dates all the way back to 1886. 1885 saw the birth of this still-popular soda ...
1886	0.849	coca-cola vs pepsi: the soda logo war by matthew roberts – on july 24, 2019 american culture ... he established the "coca-cola company" in 1892, getting his trademark sign...
1886	0.849	...in 1892, the newly incorporated coca-cola company allocated $ 11,401 for advertising...
1886	0.845	coca-cola was first introduced on may 8, 1886 by a pharmacist named ... he incorporated the coca-cola company in 1892 and...
1886	0.845	beginning with its birth at a soda fountain in downtown atlanta, georgia, in 1886, see all the milestones throughout coca-cola's memorable, 125+ year history.

ing documents is that many questions do not actually require external knowledge. Many questions only require object recognition or common sense as shown in Figure 2. Thus, the performance of InstructBLIP in Table 3 without document retrieval is 58.91%. Due to a large number of questions in training data that can be answered without documents, the model learns to ignore documents even when they contain useful information. This highlights a limitation of the OKVQA dataset, which contains too many questions that might not require outside knowledge.

6 Conclusion

We presented a simple two-stage framework for knowledge-based VQA that unifies three complementary ingredients, question-aware visual encoding, retrieval of explicit evidence, and few-shot prompting of a frozen LLM. In **Stage 1**, injecting a question-aware visual prefix into a retrieval-augmented generator strengthened both document selection and grounded answer generation, yielding competitive performance on OKVQA (62.83%) while remaining modular and interpretable through per-document predictions. In **Stage 2**, we converted those per-document answers into concise, evidence-tied candidates and paired them with question-aware captions inside compact few-shot exemplars. This *candidate+caption* recipe enabled a frozen LLM to balance explicit documents with implicit knowledge, achieving strong results with only five shots and no ensembling (61.69%), surpassing prior in-context approaches with larger prompt budgets.

Beyond raw accuracy, our analysis highlighted practical levers that affect reliability and cost: (i) increasing test-time retrieval breadth (K_{test}) improves candidate recall but must be tempered by a cap on distinct candidates (K_{AC}) to control noise; (ii) question-aware visual features materially outperform text-only surrogates for guiding retrieval; and (iii) in-context examples selection via FAISS features and caption choice meaningfully impacts few-shot transfer. The pipeline's modularity provides clear interpretable intermediate steps, from retrieved passages to per-document predictions to the final LLM choice that facilitates error diagnosis.

Future directions. Promising extensions include tighter coupling between retrieval scores and candidate ranking, confidence and calibration-aware selection during few-shot prompting, multimodal in-context examples that pass compact visual tokens and evaluations on datasets that stress diverse or long-tail knowledge. More broadly, integrating lightweight verification steps (e.g., evidence consistency checks) could further increase faithfulness without sacrificing the simplicity that makes this approach practical.

Overall, our results suggest that carefully combining grounded retrieval with question-aware visual cues and disciplined in-context prompting is an effective and scalable path for KBVQA—bridging the gap between purely retrieval-based systems and prompt-only methods while preserving interpretability and compute efficiency.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens

Image	Prompt	Ground Truth Labels	Prediction	Knowledge Required
	Context: a black motorcycle parked in a parking lot. Question: What sport can you use this for? Candidates: race (0.848), motorcycle (0.83), ride (0.827), motocross (0.826), bike (0.805), Answer:	race, motocross, ride	motocross	Object detection External knowledge
	Context: a bathroom with a plant on the wall. Question: Name the type of plant this is? Candidates: fern (0.836), succulent (0.832), wall (0.828), house (0.82), vine (0.814), Answer:	vine, climb, look like some kind of ivy, ficus	fern	Object detection
	Context: a man with a backpack with a teddy bear in it. Question: What toy is this? Candidates: teddy bear (0.846), Answer:	stuffed animal, teddy bear	teddy bear	Object detection
	Context: a man holding a dog on his back. Question: Which part of this animal would be in use of it was playing the game that is played with the items the man is holding? Candidates: tail (0.783), mouth (0.755), arm (0.744), Answer:	mouth	mouth	Object detetion External knowledge
	Context: a bathroom with a toilet and a sink. Question: Who leaves a toilet like this? Candidates: man (0.85), person (0.836), human (0.805), clean (0.791), Answer:	man, men	person	Object detection External knowledge
	Context: a kitchen with a center affixed unit. Question: A center affixed unit like this one in a kitchen is called a what? Candidates: island (0.827), mixer (0.81), bowl (0.775), stove (0.769), Answer:	island	island	Object detection External knowledge
	Context: a busy city street with many people walking around. Question: Why might someone go to this place? Candidates: shop (0.792), crowded (0.761), advertising (0.742), night (0.74), Answer:	shop, nyc, business	shop	Object detection Common sense
	Context: a baseball player holding a bat. Question: What is that man doing with the bat? Candidates: bat (0.859), swing (0.843), hit (0.841), Answer:	swing, hit, try to hit hte ball	swing	Object detection
	Context: a group of people swimming in the ocean at a salt water beach. Question: Is this at a salt water beach or a lake? Candidates: salt (0.804), lake (0.796), both (0.782), Answer:	salt water beach, salt water, lake, beach	salt	Object detection
	Context: two hot dogs with onions and peppers. Question: What is the name of the items the hot dog are topped with? Candidates: relish (0.914), pickle (0.911), onion (0.881), condiment (0.867), Answer:	condiment, onion relish, vegetable, relish	onion	Object detection
	Context: a desk with four computers and a phone. Question: What is this desk used for? Candidates: work (0.867), computer (0.842), compute (0.824), office (0.816), Answer:	work, compute, office	work	Object detection Common sense
	Context: a passenger jet sitting on top of an airport tarmac. Question: What type of plane is that? Candidates: passenger (0.832), 747 (0.829), Answer:	commercial, passenger, quanta, md 80	passenger	Object detection
	Context: a display case filled with lots of different types of donuts in 2012. Question: In what year was this desert first introduced? Candidates: 1800 (0.774), 1950 (0.77), 1953 (0.757), 1914 (0.751), 1804 (0.748), Answer:	1847, 1860, 1934, 1900s	1804	Object detection External knowledge

Figure 2: Examples of prompts for OKVQA validation questions. Prompt head and in-context prompts are not shown. Predictions are generated by Text-Davinci-003. Knowledge required shows the author's subjective view of the necessary inforation required to answer each question.

Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Feng Gao, Qing Ping, Govind Thattai, Aishwarya N. Reganti, Ying Nian Wu, and Prem Natarajan. 2022. A thousand words are worth more than a picture: Natural language-centric outside-knowledge visual question answering. *CoRR*, abs/2201.05299.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

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, Alexander Hauptmann, Yonatan Bisk, and Jianfeng Gao. 2022. KAT: A knowledge augmented transformer for vision-and-language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 956–968, Seattle, United States. Association for Computational Linguistics.

Yushi Hu, Hang Hua, Zhengyuan Yang, Weijia Shi, Noah A. Smith, and Jiebo Luo. 2023. Promptcap: Prompt-guided task-aware image captioning.

J. Johnson, M. Douze, and H. Jegou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(03):535–547.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models.

Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Relation-aware graph attention network for visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

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

Weizhe Lin and Bill Byrne. 2022. Retrieval augmented visual question answering with outside knowledge. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11238–11254, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuanze Lin, Yujia Xie, Dongdong Chen, Yichong Xu, Chenguang Zhu, and Lu Yuan. 2022. Revive: Regional visual representation matters in knowledge-based visual question answering. *arXiv preprint arXiv:2206.01201*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts.

Man Luo, Yankai Zeng, Pratyay Banerjee, and Chitta Baral. 2021. Weakly-supervised visual-retriever-reader for knowledge-based question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6417–6431, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Zhenwei Shao, Zhou Yu, Meng Wang, and Jun Yu. 2023. Prompting large language models with answer heuristics for knowledge-based visual question answering. In *Computer Vision and Pattern Recognition (CVPR)*, pages 14974–14983.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57:78–85.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *CoRR*, abs/2202.03052.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6281–6290.

A Hyperparameters

All fine-tuning hyperparameters and setups are given in this section for replication.

Table 10: Hyperparameters for Q-only and question + text-based vision OKVQA fine-tuning

LLM	T5-Large	Flan-T5-Large	Flan-T5-XL
Batch size	2		
Gradient accumulation	16		
Start learning rate	6e-5		
LR scheduler	Linear decay at 2e-8/step		
Precision	fp32		
Epoch	6		
LoRA r	NA	NA	8
LoRA α	NA	NA	32
LoRA dropout	NA	NA	0.1

Table 11: Hyperparameters for pre-training on Conceptual Captions

LLM	Flan-T5-Large	Flan-T5-XL
Batch size	64	
Gradient accumulation	2	
Start learning rate	3e-4	
LR scheduler	Constant	
Precision	fp32	
Epoch	10	
LoRA r	NA	8
LoRA α	NA	32
LoRA dropout	NA	0.1

Table 12: Hyperparameters for fine-tuning visual-language models on OKVQA without documents.

LLM	Flan-T5-Large	Flan-T5-XL
Batch size	2	
Gradient accumulation	16	
Start learning rate	1e-4	
LR scheduler	Linear decay at 2e-8/step	
Precision	fp32	
Epoch	6	
LoRA r	NA	8
LoRA α	NA	32
LoRA dropout	NA	0.1

Table 13: Hyperparameters for fine-tuning InstructBLIP-Flan-T5-XL-FrDPR on OKVQA.

LLM	Flan-T5-XL
Batch size	1
Gradient accumulation	32
Retriever learning rate	1e-5
LLM learning rate	6e-5
Precision	bf16
Epoch	4
LoRA r	8
LoRA α	32
LoRA dropout	0.1

B Image Caption Quality

The image caption is an important component of in-context examples since it describes the context to each question. Since it is not possible to include image encoding directly to GPT-3.5, we need to use text-based vision. Three visual models are used to generate captions: Oscar+ (Li et al., 2020), BLIP2 (Li et al., 2023) and PromptCap (Hu et al., 2023).

Table 14 shows that Oscar+ captions are generally more detailed than that of BLIP2 despite BLIP2 achieving higher COCO Caption performance (Li et al., 2023, 2020). BLIP2's short caption style likely aligns better with COCO's ground truth. BLIP2 also has better text-recognition ability. BLIP2's better caption translates to better in-

Table 14: Examples of image captions by Oscar+, BLIP2 and PromptCap.

Question	What profession would you say this guy has?
Oscar+	a man is working on a motorcycle in front of a tent.
BLIP2	a man working on a motorcycle.
PromptCap	a man in blue overalls working on a motorcycle.
Question	The birds on the television derive their name from what country?
Oscar+	a cat sitting in front of a television watching birds
BLIP2	a cat sitting on a television
PromptCap	a cat sitting in front of a tv with a picture of geese on it.
Question	Where is this building located?
Oscar+	a large building with a clock tower on top of it.
BLIP2	a building with a clock tower.
PromptCap	a building with a clock tower in england.
Question	What does the blue p represent?
Oscar+	a black box sitting next to a brick wall.
BLIP2	a parking meter and a brick wall.
PromptCap	a parking meter with a blue p on it.
Question	What is the name of the beer?
Oscar+	a bottle of beer next to a plate of food
BLIP2	a beer and food
PromptCap	a bottle of kingfish beer and a plate of food.

context performance. The baseline model that uses Oscar+ captions and performs worse than using BLIP2 captions (59.32 vs 59.55) in Table 5.

However, both Oscar+ and BLIP2 do not add enough relevant information that can help answer the question. In contrast, PromptCap provides captions that include details about what the question is asking about. As seen in Table 14, PromptCap describes the key object of interest in more detail. For example, when the question asks about birds on TV, PromptCap states that the birds are geese, whereas the other models do not include details about the birds. PromptCap also occasionally directly answers the question. This advantage is reflected in its better performance in in-context testing. It scores 1.2 better than BLIP2 (Table 5.)