RG-VQA: Leveraging Retriever-Generator Pipelines for Knowledge Intensive Visual Question Answering

Anonymous submission

Abstract

In this paper, we propose a method to improve the reasoning capabilities of Visual Question Answering (VQA) systems by integrating Dense Passage Retrievers (DPRs) with Vision Language Models (VLMs). While recent works focus on the application of knowledge graphs and chain-of-thought reasoning, we recognize that the complexity of graph neural networks and end-to-end training remain significant challenges. To address these issues, we introduce **R**elevance **G**uided **VQA** (**RG-VQA**), a retriever-generator pipeline that uses DPRs to efficiently extract relevant information from structured knowledge bases. Our approach ensures scalability to large graphs without significant computational overhead. Experiments on the ScienceQA dataset show that RG-VQA achieves state-of-the-art performance, surpassing human accuracy and outperforming GPT-4 by more than 8%. This demonstrates the effectiveness of RG-VQA in boosting the reasoning capabilities of VQA systems and its potential for practical applications.

1 Introduction

Visual Question Answering (VQA) has garnered significant attention in artificial intelligence for its potential to bridge the gap between visual perception and natural language understanding. VQA systems are designed to answer questions based on the content of a given image, necessitating the integration of visual and textual information. While early VQA approaches focused on answering straightforward questions that could be directly inferred from the visual content, recent research has shifted towards addressing complex, knowledge-intensive questions that require advanced reasoning capabilities in the models.

Recent studies, such as KAM-CoT (Mondal et al., 2024) and MM-CoT (Zhang et al., 2023), investigate how integrating Knowledge Graphs (KGs) and chain-of-thought (CoT) reasoning (Wei et al., 2022) can enhance these capabilities in multimodal models. A common approach involves using Graph Neural Networks (GNNs) to integrate KGs, but the complexity of these models increases significantly with the size of the graph. QA-GNN (Yasunaga et al., 2021a) and GreaseLM (Zhang et al., 2022) propose heuristics to extract a relevant sub-graphs from a KG before encoding it with a GNN. However, this sacrifices the ability to consider the entire KG and incorporate knowledge from diverse sections. Additionally, methods leveraging GNNs or CoT come with substantial latency, resulting in high inference times.

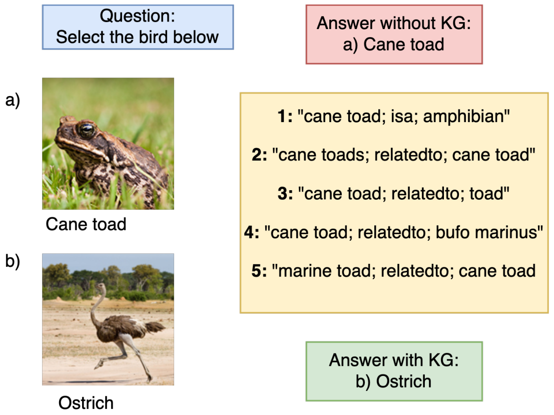

Question:
Select the bird below
Answer without KG:
a) Cane toad
a)
Cane toad
b)
Ostrich
1: "cane toad; isa; amphibian"
2: "cane toads; relatedto; cane toad"
3: "cane toad; relatedto; toad"
4: "cane toad; relatedto; bufo marinus"
5: "marine toad; relatedto; cane toad
Answer with KG:
b) Ostrich

Figure 1: An example from the ScienceQA dataset illustrating the significance of knowledge infusion. While the correct answer is not directly present in the triples, they effectively assist in eliminating incorrect options.

This paper aims to re-purpose Dense Passage Retrievers (DPRs) for structured knowledge, offering an alternative to sub-graph extraction and subsequently applying GNNs. DPRs were primarily designed to retrieve relevant passages from large text corpora (i.e., unstructured data) for a given query (Guu et al., 2020; Karpukhin et al., 2020; Lewis et al., 2020). They focus on transforming text into embeddings and performing simple similarity searches, in contrast to GNNs, which handle intricate graph structures and require iterative updates across the graph. Our empirical findings suggest that with appropriate training, DPRs can effectively substitute GNNs and proficiently extract relevant knowledge from structured knowledge bases like KGs. This has the potential to advance multimodal reasoning capabilities significantly. However, our work does not undermine the merits of GNNs; rather, it serves as a viable alternative for practical purposes.

This work represents, to the best of our knowledge, the first comprehensive study on the use of DPRs to improve the performance of Vision Language Models (VLMs) on complex, reasoning-based VQA tasks. Our contributions are:

1. We propose a multimodal retriever-generator model for VQA tasks, namely **RG-VQA** (Relevance Guided VQA). Our method exhibits competitive performance with GNN based methods. Moreover, RG-VQA is compatible with any retriever and VLM architecture. The method is also generalizable across diverse VQA datasets and KGs (Section 3).

2. We evaluate the effectiveness of our proposed RG-VQA pipeline on the ScienceQA dataset (Lu et al., 2022) using various VLMs. We provide a comprehensive analysis of the method's effectiveness along with a detailed comparison against approaches that utilize GNNs (Section 5). With RG-VQA, we achieve a test accuracy of 92.10% on the ScienceQA dataset without incorporating additional modules into the VLMs or relying on chain-of-thought prompting.

Notably, 9 of the top 10 methods on the ScienceQA leaderboard[1] (as of February 2025) necessitate multiple calls to the underlying language model, with the only exception being the Honeybee model (Cha et al., 2024). Our experiments also demonstrate that training the retriever results in an improvement of nearly 2.5% in zero-shot testing across different VLMs compared to using an off-the-shelf retriever, highlighting the significance of our training approach. We find that this improvement surpasses the accuracy of the model evaluated with triples extracted using greedy methods (Yasunaga et al., 2021a). To fully leverage knowledge augmentation, we train diverse VLMs and evaluate their ability to comprehend external knowledge.

2 Related Work

2.1 Vision Language Models

Several recent works have introduced a new paradigm for training VLMs that leverages pre-trained unimodal models instead of end-to-end pre-training (Li et al., 2021; Wang et al., 2021; Bao et al., 2022). BLIP-2 (Li et al., 2023) and LLaVA (Liu et al., 2024b) are two pivotal works in this direction, using frozen image encoders and language models with lightweight projection layers to align the modalities. This approach has led to the development of several efficient and performant VLMs, such as LaVIN (Luo et al., 2024), Honeybee (Cha et al., 2024), the Bunny family (He et al., 2024), and TinyLLaVA (Zhou et al., 2024), which employ various techniques to enhance the cross-modal alignment while maintaining the benefits of using pre-trained components. These models have consistently achieved strong results across a range of benchmarks, demonstrating the effectiveness of this new training paradigm for VLMs.

2.2 Knowledge-based VQA

Several knowledge-based visual question answering (KB-VQA) benchmarks, such as FVQA (Lin et al., 2023b), OK-VQA (Marino et al., 2019), A-OKVQA (Schwenk et al., 2022), Encyclopedic VQA (Mensink et al., 2023), and InfoSeek (Wu et al., 2023), have been proposed to evaluate the performance of models in utilizing external knowledge to answer visually-grounded questions. Various approaches have been proposed to tackle these benchmarks, including concept-aware representations (Gardères et al., 2020), integration of implicit and symbolic knowledge (Marino et al., 2021) and retrieval-augmented VQA (Lin et al., 2023a). In contrast, ScienceQA is a large-scale multimodal dataset for science question answering that not only requires external knowledge but also demands reasoning capabilities posing an additional challenge compared to other KB-VQA benchmarks. Also, for ScienceQA dataset, KG infusion techniques are explored by only KAM-COT (Mondal et al., 2024). Hence, we select ScienceQA to test our method.

2.3 KG infusion in LMs for Question Answering

Several methods augment language models with structured knowledge from KGs specifically for question answering. KagNet (Lin et al., 2019) constructs a question-specific subgraph from the

[1] https://scienceqa.github.io/leaderboard.html

KG and employs a graph convolutional network (GCN) to propagate information to the question-and-answer representations. JointLK (Sun et al., 2021) utilizes a dense bidirectional attention module for joint reasoning between LMs and GNNs, allowing mutual updates through multi-step interactions. GreaseLM (Zhang et al., 2022) uses a greedy search algorithm to extract a relevant subgraph and linearizes it into text for LM input. QA-GNN (Yasunaga et al., 2021b) integrates a GNN-based KG reasoning module with a pre-trained LM, where the GNN updates node representations through message passing. KAM-CoT (Mondal et al., 2024) encodes a sub-graph using a stack of GNNs and fuses it with image and text embeddings, incorporating them into the chain-of-thought reasoning process of a VLM.

These works demonstrate the effectiveness of augmenting LMs with structured knowledge from KGs to improve their reasoning capabilities. However, the computational overhead associated with these methods, particularly those involving GNNs, can be significant (Wu et al., 2020; Meng et al., 2021). Our work aims to explore the potential of dense passage retrievers, specifically ColBERTv2 (Santhanam et al., 2022), as a more efficient alternative for knowledge-augmented VQA while still maintaining the benefits of incorporating external knowledge into the reasoning process. Previous research on knowledge augmentation from structured knowledge using DPRs (Wu and Mooney, 2022; Nangi et al., 2023) has primarily concentrated on the task of question answering, with little attention given to tasks demanding extensive reasoning capabilities.

3 Methodology

In this section, we discuss about the retriever and generator training in detail. An overview of our approach can be seen in Figure 2.

3.1 Relevance Guided Supervision

Our retrieval process draws inspiration from the RGS method (Khattab et al., 2021), initially designed to generate training data for the ColBERT model (Khattab and Zaharia, 2020) without any prior access to training examples. The original method extracts passages from a database and classifies them as positive or negative based on a heuristic, suitable for tasks where a single passage could directly answer the query. We adapt the approach to accommodate tasks where the knowledge base aids the question-answering process by providing relevant facts, rather than direct answers.

3.1.1 Descriptions

Since ColBERTv2 uses a text-only encoder, we first extract inputs from the image in the form of image descriptions. We use LLaVA-NeXT-Mistral-7B (Liu et al., 2024a) to generate descriptions for each image. These descriptions are then included as part of the query provided to ColBERTv2.

We observe that LLaVA's descriptions, when generated without specific guidance, often lack both expressiveness and relevance to the accompanying question. To address this issue, we adopt a method akin to Generate-then-Read (Yu et al., 2023), where models first generate relevant information to answer a question and then separately provide the answer. We adopt a similar approach, directing the LLaVA model to generate more useful image descriptions by prompting it to produce descriptions that aid in answering the question.

3.1.2 Heuristic

We represent the query given to ColBERTv2 as follows.

$$q = \texttt{image\_desc} \oplus \texttt{question} \oplus \texttt{choices}$$

Here, $\oplus$ denotes concatenation.

We define heuristic $h(q)$ as the set of nouns and adjectives in the `question` and the `correct_answer`. A triple t is labeled to be positive with respect to query q if

$$|t \cap h(q)| > \min\left(0.5 \times |h(q)|, 2\right)$$

In simpler terms, to classify as positive, t should have a significant overlap with $h(q)$; otherwise, it is labeled as a negative triple. This approach is crucial to minimize false classifications and maintain relevance. Without a sufficiently high overlap threshold, even partial matches on individual words could lead to incorrect classifications. Similarly, we limit the heuristic to nouns and adjectives to avoid overestimating the relevance of a triple. Specifically, verbs and pronouns are excluded as they can be generic across various triples.

3.2 Analysis of Heuristic

In this section, we analyze the different aspects of the heuristic (Section 3.1.2), shedding light on why it is chosen in this manner.

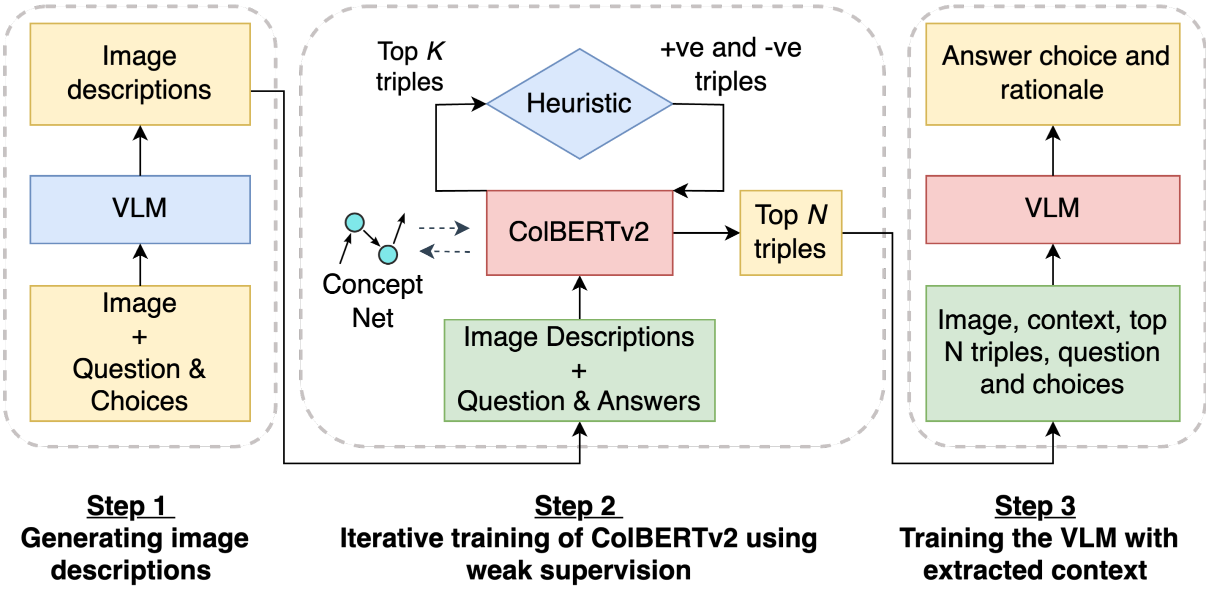


Image descriptions
VLM
Image + Question & Choices
Top K triples
Heuristic
+ve and -ve triples
ColBERTv2
Top N triples
Concept Net
Image Descriptions + Question & Answers
Answer choice and rationale
VLM
Image, context, top N triples, question and choices
Step 1
Generating image descriptions
Step 2
Iterative training of ColBERTv2 using weak supervision
Step 3
Training the VLM with extracted context

Figure 2: Overview of the RG-VQA training pipeline. In step 1, image descriptions are generated using a *frozen* VLM, which is then used as inputs for the ColBERTv2 model. In step 2, the ColBERTv2 model undergoes weakly supervised training through multiple rounds (using the heuristic defined in Section 3.1.2). Finally, in step 3, the VLM is trained to generate answers and rationales, with the top N triples from step 2 added as context.

- In any VQA dataset, certain samples contain highly generic questions that lack sufficient information to guide triple selection. To address this, we incorporate both the question and the correct answer in $h(q)$.

- We set a high threshold of 2 for classifying triples as positive or negative with respect to a given query. This choice is crucial to minimize misclassification and ensure that retrieved triples are truly relevant. For instance, consider a query representation $h(q) = \{\text{"West Virginia", "capital"}\}$. If the threshold were too low, any triple containing either "West Virginia" or "capital" could be a match, leading to the inclusion of distracting samples that do not directly contribute to finding the correct answer. By keeping a high threshold, we ensure that only triples containing both terms are marked positive, thereby improving the quality of retrieved knowledge and reducing noise in the training process.

- ConceptNet contains multiple similar triples, which may bias the training process by leading the retriever to extract the same type of triples repeatedly. This can hinder effective training. To tackle this, we perform a similarity check before categorizing the retrieved triples as positive or negative, as outlined in Stage II of Section 3.2.1.

	$A \in h(q)$, no threshold	$Q + A \in h(q)$, no threshold	$Q + A \in h(q), 2$ threshold
Round 1	465,814	**1,003,773**	306,805
Round 2	**693,865**	583,886	376,623
Round 3	434,520	192,713	**380,318**

Table 1: Number of training triples per round of retriever training under different heuristics. Comparison is made between answer-only vs. when both question and answer are considered, and no threshold vs. a threshold of 2.

Our observations were also validated quantitatively. Specifically, we train the retriever in rounds, where each round is expected to guide the model towards better retrieval. This should also increase the size of the training data as the rounds proceed. In Table 1, we see that the training data size does not show this expected trend unless the heuristic is properly chosen.

3.2.1 Training

The training process consists of R rounds. For each round $r \in [1, R]$, we follow these 4 stages:

1. **STAGE I: Indexing**
We first index the triples in the KG using the

ColBERTv2 model from round $r-1$ to form a pre-computed index $\mathcal{I}$. Here, round 0 represents the base ColBERTv2 model. Index $\mathcal{I}$ is not changed throughout round r.

2. **STAGE II: Base Retrieval**
   The second stage involves retrieving the top K triples, denoted as $\mathcal{T}(q)$, for each query q using the pre-computed index $\mathcal{I}$ from the previous stage. To avoid redundancy, if two similar triples (those with identical subjects and objects) are present in $\mathcal{T}(q)$, only the one with a higher ranking is retained.

3. **STAGE III: Training Data Instantiation**
   For every query q and its corresponding triples $\mathcal{T}(q)$ extracted in Stage II, we partition $\mathcal{T}(q)$ into positive and negative sets, $\mathcal{T}(q) = \mathcal{T}_{\mathcal{P}}(q) \bigcup \mathcal{T}_{\mathcal{N}}(q)$ using the heuristic $h(q)$ (as described in Section 3.1.2). Then, the training data for round r is obtained as,

$$\mathcal{D} = \bigcup_q \{(q, p, n) \mid p \in \mathcal{T}_{\mathcal{P}}(q),\ n \in \mathcal{T}_{\mathcal{N}}(q)\}$$

   Here, we only consider queries q for which both $|\mathcal{T}_{\mathcal{P}}(q)| > 0$ and $|\mathcal{T}_{\mathcal{N}}(q)| > 0$ hold.

4. **STAGE IV: Training ColBERTv2**
   The fourth and final stage involves training the ColBERTv2 model from round $r-1$ using the training data $\mathcal{D}$ generated in Stage III. This yields the model after round r.

3.3 Answer Generation

Finally, the top N triples extracted using the trained ColBERTv2 model are added to the prompt of the VLM as context. In the zero-shot setting, the VLM is prompted to choose the correct answer out of the given choices. Furthermore, we employ LoRA fine-tuning of various VLMs to generate a rationale and the correct answer. This is chosen over full fine-tuning since prior works have shown empirically that LoRA leads to better performance in VLMs (Zhai et al., 2023b; Laurençon et al., 2024; He et al., 2024).

4 Experimental Setup

We utilize ConceptNet as our KG because of its extensive general knowledge coverage. However, the vast scale of ConceptNet, with over 2 million nodes and 21 million edges, significantly increases the retriever's training time. To address this, we selectively extract triples corresponding to the 20 most frequent relations out of the 34 available in ConceptNet. This reduction results in a smaller KG containing approximately 2.5 million triples.

4.1 Datasets

We conduct experiments on ScienceQA (Lu et al., 2022), a benchmark for multi-modal learning and reasoning covering diverse science topics, and containing annotations of answers with corresponding lectures and explanations. The dataset consists of 12726, 4241, and 4241 train, dev, and test samples respectively.

4.2 Retriever

We employ the 110M-parameter ColBERTv2 as the retriever, as it captures better cross-sequence (query and triple) interaction than a traditional bi-encoder with its token-level embeddings. It is also faster than most cross-encoders since the triple embeddings can be pre-computed and retrieved independently of the query. We train the model for $R = 3$ rounds using the strategy described in Section 3.2.1 on 1 NVIDIA A-100 80GB GPU. In each round, stage I takes around 30 minutes on average, stage II takes around 60 -70 minutes, and stage IV requires another 40 - 60 minutes (depending on the amount of training data created in stage III). Training takes $<$ 3 hours for one round. In stage II, we choose the value of K as 200. The maximum length for the query encoder is 256, while that for the triple encoder is 64. We train the model with a batch size of 32. We use the open-source RAGatouille library[2] for training.

4.3 Generator

We consider the following recent models: LLaVA-1.5 Vicuna-13B (Liu et al., 2023), LLaVA-NeXT Mistral-7B (Liu et al., 2024a), Bunny Llama3-8B (He et al., 2024), and MM-CoT FlanT5-Base (250M) (Zhang et al., 2023). While all models except Bunny employ the CLIP encoder (Radford et al., 2021), Bunny and TinyLLaVA-Gemma utilizes the SIGLIP vision encoder (Zhai et al., 2023a). All large models follow a similar two-stage training paradigm, which involves feature alignment followed by supervised fine-tuning.

4.3.1 Zero-shot Evaluations

For the zero-shot evaluations, we employ the standard prompt given for ScienceQA, outlined in (Liu

[2] https://github.com/bclavie/RAGatouille

Model	Size	NAT	SOC	LAN	TXT	IMG	NO	G1-6	G7-12	Avg
Human	-	90.23	84.97	87.48	89.60	87.50	88.10	91.59	82.42	88.40
CoT (ChatGPT)	175B	78.82	70.98	83.18	77.37	67.92	86.13	80.72	74.03	78.31
CoT (GPT-4)	1T+	85.48	72.44	90.27	82.65	71.49	92.89	86.66	79.04	83.99
Chameleon (GPT-4)	1T+	89.83	74.13	89.82	88.27	77.64	92.13	88.03	83.72	86.54
MM-CoT (T5 Base)	223M	84.06	92.35	82.18	82.75	82.75	84.74	85.79	84.44	85.31
DDCoT (T5 Base)	223M	88.72	86.84	84.91	87.59	83.34	88.08	88.58	85.10	87.34
MM-CoT (Flan T5)	250M	91.5	74.92	90.09	91.69	84.28	90.52	88.14	87.01	87.74
MC-CoT (Flan T5)	248M	**93.56**	83.58	90.73	94.13	89.24	90.94	90.93	90.38	90.73
LaVIN (LoRA)	13B	89.88	94.49	89.82	88.95	87.61	91.85	91.45	89.72	90.83
TinyLLaVA-Gemma	2B	87.39	94.15	85.18	91.14	85.03	88.01	86.35	89.28	88.21
Bunny Llama3 (LoRA)	8B	89.96	94.38	91.45	94.96	87.21	92.82	91.81	90.31	91.28
LLaVA-1.5 (LoRA)	13B	91.79	96.06	88.18	91.06	90.48	90.31	92.22	90.90	91.75
RG-VQA (MM-CoT)	360M	92.67	80.31	90.09	92.13	87.01	91.43	89.90	88.53	89.41
RG-VQA (TinyLLaVA-Gemma)	2B	88.37	95.05	85.27	91.10	86.61	87.67	89.50	88.00	88.96
RG-VQA (LLaVA-1.5)	13B	92.81	95.61	87.45	91.89	**91.03**	90.03	**92.62**	90.90	92.01
RG-VQA (Bunny)	8B	90.45	**96.85**	**91.64**	**95.37**	88.50	**93.24**	92.47	**91.43**	**92.10**

Table 2: We compare our results on the ScienceQA dataset with different baselines that do not use knowledge. All scores are for exact match accuracy (in %). Here, Size = size of the backbone model, NAT = Natural Science, SOC = Social Science, LAN = Language Science, TXT = Text context, IMG = Image context, NO = No context, G1-6 = Grade 1 to 6, G7-12 = from Grade 7 to 12, Avg = Average accuracy. The best score for each category is marked in **bold**. Segment 1 compares against the human average. Segment 2 has the performance of the GPT family. Segment 3 compares with models that utilize multimodal CoT. In Segment 4, we show parameter-efficient finetuned versions of different LLMs. The score of MM-CoT (Flan T5 Base) is taken from Mondal et al. (2024), where a caption is also given as context along with the vision features. Results, other than ours and that of fine-tuned Bunny and LLaVA-1.5 models, are taken from respective papers and the ScienceQA leaderboard.

et al., 2024b), with triples added as part of the context. The exact prompt can be found in Appendix D.1. Models are instructed to generate only the correct answer choice, and we report exact match accuracies. To ensure reproducibility and obtain more reliable results, all evaluations are conducted using greedy decoding with 3 beams, and $N = 10$ triples (Section 3.3).

4.3.2 Generator Fine-tuning

We perform LoRA fine-tuning on the LLaVA-1.5 Vicuna-13B and Bunny-Llama3-8b models and full fine-tuning for TinyLLaVA-Gemma. We also train the MM-CoT FlanT5-Base model, with captions added as context (similar to the approach followed in Mondal et al. (2024)). The training is performed on 4 NVIDIA A-100 80GB GPUs. The triples retrieved after each round of training are provided as additional context during fine-tuning. Due to limitations in the context length (set as 2048), we only add the top $N = 25$ retrieved triples, and train the models to generate both explanation and the correct answer.

For both the LLaVA-1.5 and Bunny models, we conduct supervised fine-tuning starting from the base checkpoint, which has not been pre-trained on any instruction-following datasets. The LLaVA-1.5 model is fine-tuned for 12 epochs, while the Bunny model undergoes fine-tuning for 2 epochs. This training is carried out using LoRA and PEFT, keeping the number of trainable parameters to around 0.4% of the total model size. The global batch size is kept at 128, with a training batch size of 8 per device, and gradient accumulation steps set as 4. The learning rate is set at 2e-5 for LLaVA-1.5 and 2e-4 for Bunny. Training takes around 30 minutes per epoch. To ensure optimal performance, we evaluate our model on the development set after each epoch and save the model with the best checkpoint upon the completion of training.

5 Results and Analysis

We compare RG-VQA with 3 different categories of models from the official leaderboard of ScienceQA: (i) Techniques using GPT (Achiam et al., 2023; Wei et al., 2022; Lu et al., 2024), (ii) Models based on MM-CoT, with T5 or Flan T5 as the underlying LLM (Zhang et al., 2023; Zheng et al., 2023; Tan et al., 2023) and (iii) VLMs fine-tuned with LoRA (Luo et al., 2024; He et al., 2024; Liu et al., 2023). As shown in Table 2, the proposed RG-VQA technique outperforms all models not utilizing knowledge, with the highest accuracy of 92.10% achieved by training the Bunny model.

Our results demonstrate an improvement of nearly 2.5% in the Social Science category when applying RG-VQA to the Bunny model, achieving a score of 96.85%. This surpasses the previous best accuracy of 96.74% on the ScienceQA leaderboard, achieved by the LLaVA and GPT-4 synergy on the Social Science subset. Similarly, a substantial improvement of nearly 6.5% is observed in the

	LLaVA-1.5 Vicuna-13B	**LLaVA-NeXT Mistral-7B**	**Bunny Llama3-8B**	**TinyLLaVA-Gemma-2B**
Without KG	69.96	76.56	81.49	55.56
Base Triples	65.27	70.05	75.01	51.00
Round 1 Triples	66.85	71.40	77.29	52.74
Round 2 Triples	66.87	72.08	**77.53**	52.95
Round 3 Triples	**67.37**	**72.53**	77.36	**53.93**
KAM-CoT Triples	65.6	72.01	76.26	51.00

Table 3: Zero-shot results with different VLMs (Section 4.3.1). "Without KG" refers to the evaluation done without the addition of any knowledge triples. "Base triples" consider the addition of triples retrieved using the pre-trained ColBERTv2 model, and the triples extracted with the ColBERTv2 model trained for r rounds (Section 3.2.1) are denoted as "Round r triples" (where $r \in [1, 3]$). "QA-GNN triples" is used to show the inclusion of triples, extracted using the method followed in (Mondal et al., 2024), as part of the context. All results are with the top 10 triples (Appendix D.1), and report top-1 accuracy (in %).

	LLaVA-1.5 Vicuna-13B (LoRA)	**Bunny Llama3-8B (LoRA)**	**MM-CoT FlanT5-Base**	**TinyLLaVA-Gemma-2B**
Without KG	91.75	91.28	87.74	88.21
Base Triples	91.46	91.84	89.15	88.42
Round 3 Triples	**92.01**	**92.10**	**89.41**	**88.96**

Table 4: Performance comparison of different models after fine-tuning. Training and evaluation are done with 25 triples added to the prompt (Appendix D.3). The other settings are the same as that for Table 3.

same category when applying RG-VQA to MM-CoT. However, we generally observe a decline in the accuracy of Language Science questions, likely due to the absence of triples necessary to deduce linguistic features within sentences, which constitutes the majority of questions in this category.

We also provide a few generated samples in Appendix D.4, to visualize the effect of knowledge augmentation in RG-VQA. Results for sensitivity analysis test with varying temperature settings on TinyLLaVA-Gemma model can be seen in appendix B. Results of RG-VQA method on A-OKVQA dataset are presented in appendix A. Alongside the KAM-COT method, we compare RG-VQA with another end-to-end model, Unifer, with the corresponding results detailed in C

5.1 Effect of Retriever Training

Tables 3 and 4 present the performance of various VLMs on the ScienceQA dataset, both in zero-shot and fine-tuned settings. In the zero-shot setting, incorporating KG triples extracted using the base ColBERTv2 retriever results in a notable decline in accuracy compared to the baseline without KG integration. This decline can be attributed to two main factors: the challenge VLMs face in effectively utilizing the additional knowledge without appropriate training, and the irrelevance of the triples extracted by ColBERTv2 without fine-tuning. As we conduct multiple rounds of training, we notice a continuous rise in accuracy, reaching an improvement of over 2% across all models. This consistent performance enhancement indicates that the iterative refinement of the retriever helps VLMs better utilize the KG triples. Our training approach enables the models to identify and focus on the most relevant knowledge, progressively aligning the retrieved information with the task at hand.

In comparison to the zero-shot setting, the fine-tuned models show significantly higher accuracy, demonstrating their ability to adapt to the dataset's specific characteristics and utilize the provided KG triples effectively. Remarkably, the performance of models without KG integration is already quite high, with LLaVA-1.5-Vicuna-13B and Bunny-Llama3-8B achieving accuracies above 91%. However, incorporating KG triples during training further improves their performance. Specifically, for RG-VQA based on Bunny and MM-CoT as the VLM, we see an accuracy increase of 0.8% and 1.7%, respectively. Additionally, we observe that retriever training, followed by fine-tuning with ex-

Model	Accuracy	Size	Trainable params	#KG Entities	Training time ↓	Inference time ↓
KAM-CoT	93.87	280M	280M	200	20 hours	116 minutes
RG-VQA	92.10	8B	**140M**	**800k**	**10 hours**	**80 minutes**

Table 5: Comparison of RG-VQA and KAM-CoT. The inference time is for the complete test set.

tracted triples improves accuracy across all three VLMs.

5.2 Comparison with GNNs

We compare our method, RG-VQA, with the baseline KAM-CoT (Mondal et al., 2024) on the ScienceQA benchmark, which employs GNNs for KG infusion. KAM-CoT also utilizes ConceptNet as its KG, first extracting a relevant subgraph and then encoding it using a combination of graph layers. Table 5 presents a comparison of key aspects between RG-VQA (Bunny) and KAM-CoT (FlanT5 Base). We observe that RG-VQA achieves comparable performance after training the Bunny model for just 2 epochs, despite utilizing only half the trainable parameters (110M for ColBERTv2 and 30M for Bunny) and leveraging the entire KG during both training and evaluation, unlike KAM-CoT. Additionally, evaluating the entire ScienceQA test set (4,241 samples) takes only 80 minutes on RG-VQA, compared to 116 minutes for KAM-CoT. This difference arises because RG-VQA requires just 20 minutes to retrieve relevant entities, whereas KAM-CoT takes approximately 84 minutes for subgraph retrieval, significantly increasing its overall inference time. The faster retrieval process in RG-VQA shows the efficiency of our method as an alternative to GNN based models and eliminates the costly subgraph extraction steps by considering the entire KG during both training and evaluation.

5.3 Importance of Image Descriptions

To assess the effectiveness of adding image descriptions to the retriever training process, we conduct an ablation study. We evaluate the benefits of using descriptions against the additional computational overhead and potential noise introduced by including an extra model. The results presented in Table 6 show that using image descriptions enhances the answer accuracy. Notably, models trained with descriptions consistently achieve higher accuracy across all training rounds compared to the no description setting. Moreover, without descriptions, there was no observed accuracy improvement over successive training rounds, emphasizing their crucial role in effective retriever training within our proposed pipeline.

	With description	No description
Base Triples	70.05	68.88
Round 1	71.40	**71.19**
Round 2	72.08	70.93
Round 3	**72.53**	70.90

Table 6: Zero-shot evaluation with LLaVA-NeXT Mistral-7B when image descriptions are provided to the ColBERTv2 model vs. when training is done without any image input (as discussed in Section 3.1.1).

6 Conclusion and Future Work

In this paper, we introduce RG-VQA, a new framework designed to enhance knowledge-augmented multimodal reasoning. Our approach is adaptable to various retrievers, VLMs, and KGs, and involves weakly supervised training of the retriever. Experiments on the ScienceQA dataset demonstrate the effectiveness of our retriever training pipeline, showing improvements post-training. With the Bunny Llama3-8B model, RG-VQA achieves an accuracy of 92.10%. The RG-VQA paradigm consistently improves performance across various model scales, ranging from 250M to 13B parameters. Our technique presents a viable alternative to methods involving GNNs and CoT reasoning, offering similar results with the ability to scale to large KGs efficiently while maintaining reasonable inference latencies. Future work includes integrating a visual retriever to replace the first two steps (as in Figure 2) for direct image feature incorporation, and using a KG with multimodal nodes, instead of a text-only KG.

7 Limitations

We acknowledge certain limitations in our work that highlight areas for future research. One notable issue is the insufficient knowledge triples for enhancing reasoning on Language Science questions, indicating a need for a Knowledge Graph with better coverage of linguistic devices. Additionally, our current pipeline is unable to recognize named entities in images, which may be crucial for some datasets. This limitation can be addressed by employing well-trained vision encoders in combination with multimodal Knowledge Graphs, enabling the retrieval of relevant multimodal context about the entities.

Ethics Statement

This research was conducted in accordance with the ACL Ethics Policy.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. 2024. Honeybee: Locality-enhanced projector for multimodal llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

François Gardères, Maryam Ziaeefard, Baptiste Abeloos, and Freddy Lécué. 2020. Conceptbert: Concept-aware representation for visual question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 489–498.

Yangyang Guo, Liqiang Nie, Yongkang Wong, Yibing Liu, Zhiyong Cheng, and Mohan S. Kankanhalli. 2022. A unified end-to-end retriever-reader framework for knowledge-based vqa. In *ACM Multimedia Conference*. ACM.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Muyang He, Yexin Liu, Boya Wu, Jianhao Yuan, Yueze Wang, Tiejun Huang, and Bo Zhao. 2024. Efficient multimodal learning from data-centric perspective. *arXiv preprint arXiv:2402.11530*.

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.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Relevance-guided supervision for OpenQA with ColBERT. *Transactions of the Association for Computational Linguistics*, 9:929–944.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. 2024. What matters when building vision-language models? *arXiv preprint arXiv:2405.02246*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

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. In *International conference on machine learning*, pages 19730–19742. PMLR.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151*.

Weizhe Lin, Jinghong Chen, Jingbiao Mei, Alexandru Coca, and Bill Byrne. 2023a. Fine-grained late-interaction multi-modal retrieval for retrieval augmented visual question answering. *Advances in Neural Information Processing Systems*, 36:22820–22840.

Weizhe Lin, Yifeng Jiang, Yichong Zhu, Yelong Guo, Qi Zhang, and Xuanjing Huang. 2023b. Fvqa 2.0: Introducing adversarial samples into fact-based visual question answering. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024a. Llava-next: Improved reasoning, ocr, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024b. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2024. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36.

Gen Luo, Yiyi Zhou, Tianhe Ren, Shengxin Chen, Xiaoshuai Sun, and Rongrong Ji. 2024. Cheap and quick: Efficient vision-language instruction tuning for large language models. *Advances in Neural Information Processing Systems*, 36.

Kenneth Marino, Xinlei Chen, Devi Parikh, Abhinav Gupta, and Marcus Rohrbach. 2021. Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14111–14121.

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

Yuxian Meng, Shi Zong, Xiaoya Li, Xiaofei Sun, Tianwei Zhang, Fei Wu, and Jiwei Li. 2021. Gnn-lm: Language modeling based on global contexts via gnn. *arXiv preprint arXiv:2110.08743*.

Thomas Mensink, Jasper Uijlings, Lluis Castrejon, Arushi Goel, Felipe Cadar, Howard Zhou, Fei Sha, André Araujo, and Vittorio Ferrari. 2023. Encyclopedic vqa: Visual questions about detailed properties of fine-grained categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3113–3124.

Debjyoti Mondal, Suraj Modi, Subhadarshi Panda, Rituraj Singh, and Godawari Sudhakar Rao. 2024. Kam-cot: Knowledge augmented multimodal chain-of-thoughts reasoning. In *AAAI Conference on Artificial Intelligence*.

Sharmila Reddy Nangi, Michihiro Yasunaga, Hongyu Ren, Qian Huang, Percy Liang, and Jure Leskovec. 2023. Dense retrieval of knowledge graphs for question answering.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-okvqa: A benchmark for visual question answering using world knowledge. In *European conference on computer vision*, pages 146–162. Springer.

Yueqing Sun, Qi Shi, Le Qi, and Yu Zhang. 2021. Jointlk: Joint reasoning with language models and knowledge graphs for commonsense question answering. *arXiv preprint arXiv:2112.02732*.

Cheng Tan, Jingxuan Wei, Zhangyang Gao, Linzhuang Sun, Siyuan Li, Xihong Yang, and Stan Z Li. 2023. Boosting the power of small multimodal reasoning models to match larger models with self-consistency training. *arXiv preprint arXiv:2311.14109*.

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

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

Jialin Wu and Raymond J Mooney. 2022. Entity-focused dense passage retrieval for outside-knowledge visual question answering. *arXiv preprint arXiv:2210.10176*.

Zhuofan Wu, Yifeng Jiang, Weizhe Lin, Yichong Zhu, Qi Zhang, and Xuanjing Huang. 2023. Can pre-trained vision and language models answer visual information-seeking questions? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021a. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021b. Qagnn: Reasoning with language models and knowledge graphs for question answering. *arXiv preprint arXiv:2104.06378*.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, S Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *International Conference on Learning Representations*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023a. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986.

Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. 2023b. Investigating the catastrophic forgetting in multimodal large language models. *arXiv preprint arXiv:2309.10313*.

X Zhang, A Bosselut, M Yasunaga, H Ren, P Liang, C Manning, and J Leskovec. 2022. Greaselm: Graph reasoning enhanced language models for question answering. In *International Conference on Representation Learning (ICLR)*.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*.

Ge Zheng, Bin Yang, Jiajin Tang, Hong-Yu Zhou, and Sibei Yang. 2023. Ddcot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. *Advances in Neural Information Processing Systems*, 36:5168–5191.

Baichuan Zhou, Ying Hu, Xi Weng, Junlong Jia, Jie Luo, Xien Liu, Ji Wu, and Lei Huang. 2024. Tinyllava: A framework of small-scale large multimodal models. *arXiv preprint arXiv:2402.14289*.

A Performance on A-OKVQA dataset

To validate the effectiveness of our approach we have evaluated RG-VQA method on A-OKVQA dataset (Schwenk et al., 2022). A-OKVQA is another VQA dataset made to ensure that all questions would need external knowledge to be answered. It comes with 17k training, 1k validation and 6.7k test samples. But since the test-set is closed, we use the validation split as the held-out set for reporting our scores. Out of the 17k training samples, we randomly picked 13k samples for training and use the remaining 4k samples as the dev-set. As shown in Table 7, the RG-VQA method significantly outperforms prior approaches, demonstrating its strong generalizability to knowledge-intensive VQA datasets.

	Results on A-OKVQA dataset
KAM-CoT	59.65
MM-CoT	55.98
RG-VQA (ours)	**77.38**

Table 7: Results on A-OKVQA dataset using RG-VQA method.

B Evaluating Model Robustness: Sensitivity Analysis of Temperature Variations

We conducted a sensitivity analysis test with varying temperature settings on TinyLLaVA-Gemma model (Zhou et al., 2024). By systematically altering the temperature values and assessing the model's outputs, we aim to understand how these variations affect the accuracy and reliability of our method. We observe that the results are consistent across temperatures.

temperature	0.0	0.2	0.4	Average
Base Triples	88.42	86.10	86.20	86.90
Round 1	88.51	86.22	87.34	87.35
Round 2	88.70	**88.78**	87.70	**88.39**
Round 3	**88.96**	87.20	**88.10**	88.08

Table 8: Results on ScienceQA dataset for TinyLLaVA-Gemma across different temperatures. Each column corresponds to the temperature of the model during inference on test set.

C End-to-End VQA Model

To demonstrate the effectiveness of our proposed method, we compared it with another end-to-end VQA model (refer Table 9), based on the UnifER approach (Guo et al., 2022), which uses a weak supervision signal from the generated answer to guide the retriever. We made the following modifications in UnifER:

1. **Adaptation to MCQA**: Converted the model from Open-ended Visual Question Answering to Multiple-choice Question Answering (MCQA).

2. **Positional Bias Mitigation**: Introduced permuted answer choices by passing multiple instances of the same example from the ScienceQA (Lu et al., 2022) dataset.

3. **Separate Encoders**: Unlike UnifER, which shares an encoder between the retriever and generator, we found this setup ineffective for

Model	Size	NAT	SOC	LAN	TXT	IMG	NO	G1-6	G7-12	Avg
E2E-VQA	614M	76.67	91.09	84.80	81.67	81.98	79.10	83.20	79.31	81.80

Table 9: End-to-End VQA model results on the ScienceQA dataset. All scores are for exact match accuracy (in %). Here, Size = size of the backbone model, NAT = Natural Science, SOC = Social Science, LAN = Language Science, TXT = Text context, IMG = Image context, NO = No context, G1-6 = Grade 1 to 6, G7-12 = from Grade 7 to 12, Avg = Average accuracy.

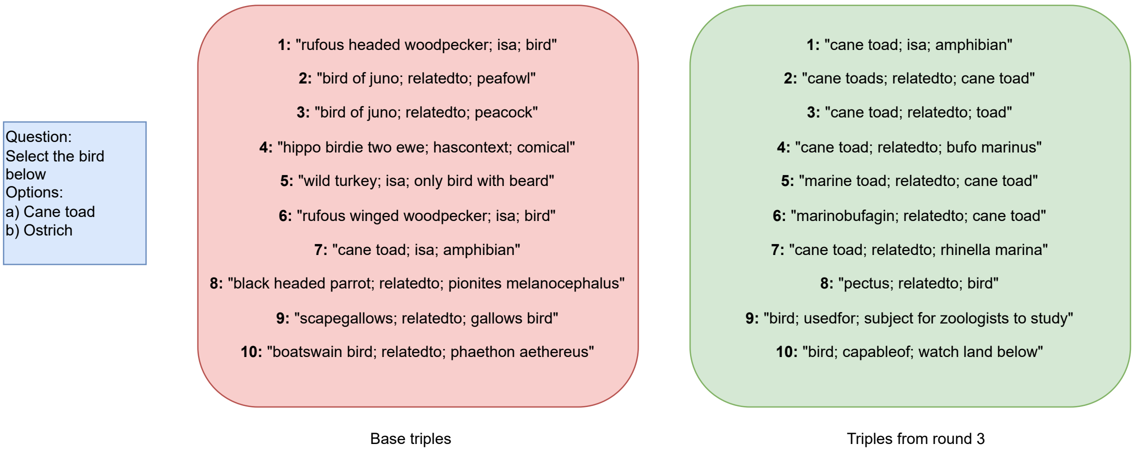

Question:
Select the bird below
Options:
a) Cane toad
b) Ostrich
1: "rufous headed woodpecker; isa; bird"
2: "bird of juno; relatedto; peafowl"
3: "bird of juno; relatedto; peacock"
4: "hippo birdie two ewe; hascontext; comical"
5: "wild turkey; isa; only bird with beard"
6: "rufous winged woodpecker; isa; bird"
7: "cane toad; isa; amphibian"
8: "black headed parrot; relatedto; pionites melanocephalus"
9: "scapegallows; relatedto; gallows bird"
10: "boatswain bird; relatedto; phaethon aethereus"
Base triples
1: "cane toad; isa; amphibian"
2: "cane toads; relatedto; cane toad"
3: "cane toad; relatedto; toad"
4: "cane toad; relatedto; bufo marinus"
5: "marine toad; relatedto; cane toad"
6: "marinobufagin; relatedto; cane toad"
7: "cane toad; relatedto; rhinella marina"
8: "pectus; relatedto; bird"
9: "bird; usedfor; subject for zoologists to study"
10: "bird; capableof; watch land below"
Triples from round 3

Figure 3: An example to demonstrate the effect of training on the quality of the triples. We can observe that the triples from round 3, i.e., from the trained ColBERTv2 model, show a higher degree of relevance to the question.

MCQA and used separate encoders for each module.

Our model consistently outperformed the end-to-end VQA system in every category, with an average accuracy improvement of over 10%.

D Prompts

In this section, we present the prompts that are used for training and evaluating VLMs. The variable parts are shown within ⟨·⟩, and may be removed if they are not available.

D.1 Zero-shot

Question: ⟨ question ⟩
Context: ⟨ hint + triples ⟩
Answer Choices: ⟨ answer choices ⟩
Choose the correct answer choice number from 1 to ⟨ len(choices) ⟩. Correct answer choice: "

Figure 5: Prompt template for zero-shot evaluation.

D.2 Generate-then-Read prompt

Instruction: Generate descriptions of the image that will be useful to answer the question below. Only give the description.
Question: ⟨ question ⟩
Answer Choices: ⟨ answer choices ⟩

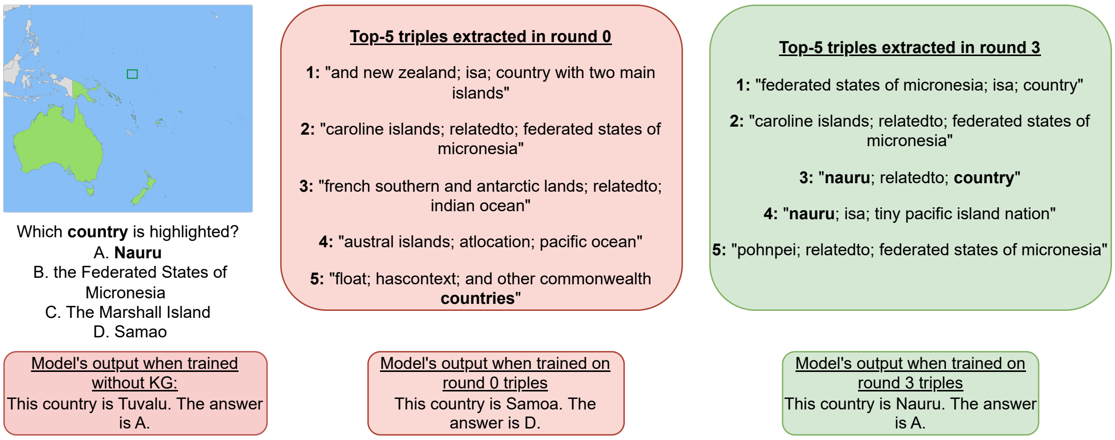
Figure 6: Prompt template for generating image descriptions.

D.3 Fine-tuning

⟨ question ⟩
Context: ⟨ hint ⟩
Options: ⟨ options ⟩
Use the following triples for additional context: ⟨ triples ⟩

Figure 7: Input template when fine-tuning the VLMs.

⟨ lecture ⟩
⟨ solution ⟩
The answer is ⟨ choice ⟩.



Which **country** is highlighted?
A. **Nauru**
B. the Federated States of Micronesia
C. The Marshall Island
D. Samao

Top-5 triples extracted in round 0

1: "and new zealand; isa; country with two main islands"

2: "caroline islands; relatedto; federated states of micronesia"

3: "french southern and antarctic lands; relatedto; indian ocean"

4: "austral islands; atlocation; pacific ocean"

5: "float; hascontext; and other commonwealth **countries**"

Top-5 triples extracted in round 3

1: "federated states of micronesia; isa; country"

2: "caroline islands; relatedto; federated states of micronesia"

3: "**nauru**; relatedto; **country**"

4: "**nauru**; isa; tiny pacific island nation"

5: "pohnpei; relatedto; federated states of micronesia"

Model's output when trained without KG:
This country is Tuvalu. The answer is A.

Model's output when trained on round 0 triples
This country is Samoa. The answer is D.

Model's output when trained on round 3 triples
This country is Nauru. The answer is A.

Figure 4: An example that highlights the importance of training the retriever. The words in **bold** represent the words that form part of the heuristic, showing how training aligns the triples with the question. The outputs are generated by the Bunny Llama3-8B model. Round 0 represents the triples extracted with the base ColBERTv2 model, and round 3 denotes the triples retrieved with the ColBERTv2 model obtained after 3 rounds of training.

Figure 8: Expected output during training of VLMs.

D.4 Qualitative Examples

In Figures 4 and 3, we present some examples from our experiments to illustrate the impact of training the retriever and demonstrate how the retrieved triples can be crucial in guiding the model to the correct answer.