# Answering Complex Geographic Questions by Adaptive Reasoning with Visual Context and External Commonsense Knowledge

**Fan Li, Jianxing Yu*, Jielong Tang, Wenqing Chen, Hanjiang Lai, Yanghui Rao, Jian Yin**
School of Artificial Intelligence, Sun Yat-sen University, Zhuhai, 519082, China
School of Software Engineering, Sun Yat-Sen University, Zhuhai, 519082, China
Key Laboratory of Sustainable Tourism Smart Assessment Technology, Ministry of Culture and Tourism
{lifan77, yujx26, tangjlong3, chenwq95, laihanj3, raoyangh, issjyin}@mail.sysu.edu.cn

## Abstract

This paper focuses on a new task of answering geographic reasoning questions based on the given image (called *GeoVQA*). Unlike traditional *VQA* tasks, *GeoVQA* asks for details about the image-related culture, landscape, etc. This requires not only the identification of the objects in the image, their properties and relations, but also the understanding of the geographic knowledge of the objects, such as location, transportation, landmark, cuisine, etc. This background knowledge does not explicitly appear in the image, nor is there an extra-textual description. Without this missing but necessary knowledge, it is difficult for existing matching-based methods to infer the correct answer. To tackle these challenges, we propose a new geographic reasoning framework for our task. We first analyze the image and describe its fine-grained content by text and keywords using a multi-modal retrieval augmented technique, so as to deduce an answer in a unified textual modality. Next, we retrieve the crucial geographic commonsense knowledge. To reduce the retrieval complexity, we design a dynamic method that can adaptively collect the relevant clues for each reasoning step. The step in the incorrect direction will be pruned according to some judgment criteria. The remaining steps can help us form a reasoning chain to derive a correct answer. Moreover, we create a large-scale dataset *GVQA* with 41,329 samples to conduct the evaluation. The results demonstrate the effectiveness of our approach.

## 1 Introduction

The visual question answering (*VQA*) task is an important research topic in the field of *CV* and *NLP*. It aims to answer the textual questions by referring to the visual content in the given image. Traditional *VQA* work mainly studies the recognition of objects, scenes, and actions in the image, such as

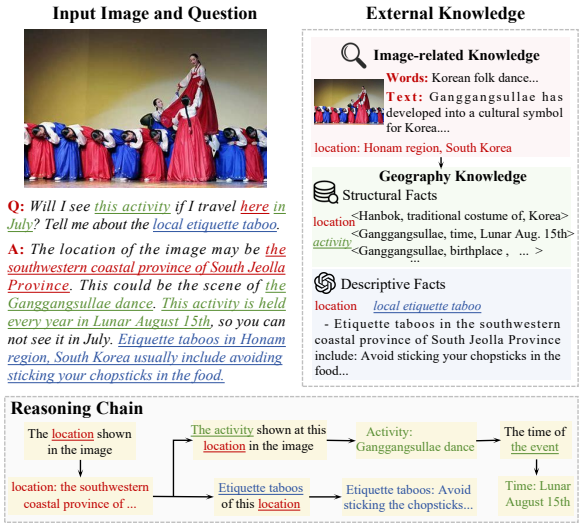


Figure 1: An example of the *GeoVQA* question.

"*What is the blue object on the red chair*?" These object detection-based tasks have been well studied. The answers can be found in the image by matching. In addition to these questions, there are still many complex ones that require deep reasoning and have not been explored. As shown in Fig.(1), the users ask whether they could see the activity depicted in the image when they visit the site in July. To get an accurate answer, we need to not only identify the image objects, but also understand the background context. For example, through the traditional clothes people wore, we can infer that the event took place in Korea. This geographic context helps us to better determine that the activity in the image is *Ganggangsullae* rather than other dances. However, this important context is not explicitly depicted in the image, nor is it provided in the question. Without this context, traditional methods are hard to use to accurately infer the answer based on the identified objects in the image. Such complex reasoning questions are common in many scenarios, and solving them can better support a wide range of applications like education, transportation,

* Corresponding author.



*Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25498–25514

tourism consulting, etc. They have great research and commercial value, but big challenges. There is a research gap in answering these geographic reasoning questions. Thus, we formulate it as a new task, called *GeoVQA*.

To tackle this task, traditional *VQA* methods often first encode the image content and textual questions by networks like *BERT*, *ViT*, etc. These encodings are then combined and fed through the workflow network to the decoder to get answers. However, these methods only identify surface-level image content but neglect the deeper contextual background, resulting in weak reasoning capabilities. They may be able to solve object recognition-based questions, but difficult to answer *GeoVQA* questions requiring complex multi-modal reasoning. Another solution is to convert multi-modal inputs into text by techniques like image captioning. In such a uniform modality, a text-based QA model can be used to derive answers more easily. Early QA methods rely on rules and templates that are hand-crafted, with poor scalability. Subsequent researches turn to neural networks, which are data-driven and have better scalability. Training these networks demands substantial data, which incurs high costs in labeling. Most studies resort to large language models (*LLMs*), which are good few-shot learners. However, *LLMs* often have an inherent hallucination that leads to misjudgment. To solve these issues, the mainstream method proposes to inject external knowledge to augment the reasoning ability of the model. An intuitive approach is to extract all question-related knowledge clues at once and use a graph network to encode them into the QA model. However, the complex reasoning questions involve $k$-order clues. The number of clues grows exponentially with the length of the reasoning chain, resulting in huge computational costs. Not all clues are conducive to correct reasoning, extraneous ones can introduce noise detrimental to performance. These challenges make it hard for current methods to tackle the *GeoVQA* task well.

Motivated by the above observations, we propose a new adaptive reasoning framework. It can retrieve sufficient but nonexcessive geographic knowledge related to the question step-by-step. That can guide the model in a correct inference direction, helping to form a reasoning chain to deduce the correct answer. In detail, we first analyze the image content and convert it into a text description by using the multi-modal retrieval augmentation technique. In this way, we can perform reasoning conveniently in a unified text modality, rather than crossing different modalities that would suffer from the heterogeneous gap. Since complex questions often involve external geographic knowledge, we then adaptively retrieve it as supplementary clues and conduct multi-hop reasoning based on it. The question often involves multiple reasoning steps, each of which has several directions, leading to a large number of relevant knowledge clues. To reduce the computational cost, we design a dynamic retrieval framework, which can retrieve clues tailored to just one step from an appropriate source. Considering the clues may contain redundancy and noise, we also design a multi-criteria verifier to evaluate them, identify and prune the noise that leads to the wrong inference direction. Since we only need to retrieve relevant external knowledge that fits the current step rather than the full amount, and do not need to cover the knowledge in the wrong direction, the computation can be greatly reduced. We then perform one-step inference to determine the next direction based on an *LLM*-based model. In this way, the reasoning result of the current step serves as the context for the next step. Finally, we can obtain a reasoning chain to the answer. This chain can guide the model to generate the right answer. The whole model with a retriever and generator is jointly trained by a reinforced framework. Due to the lack of available datasets to evaluate our task and approach, we construct a large-scale dataset called *GVQA*. We conduct extensive experiments on it and obtain significant improvement over other baselines.

The main contributions of this paper include

- We propose a new task called *GeoVQA*. In contrast to traditional simple matching-based *VQA*, our task aims to answer complex geographic questions that involve reasoning with multiple knowledge related to a given image.
- We develop a new framework that adaptively retrieves relevant knowledge from multiple sources to deduce the correct answer.
- We construct a large-scale evaluation dataset for this task to facilitate research in this field. Extensive experiments are conducted to examine the effectiveness of our approach fully.

## 2 Approach

This task aims to derive the accurate answer $a$ to the geography-related question $q$ based on the given

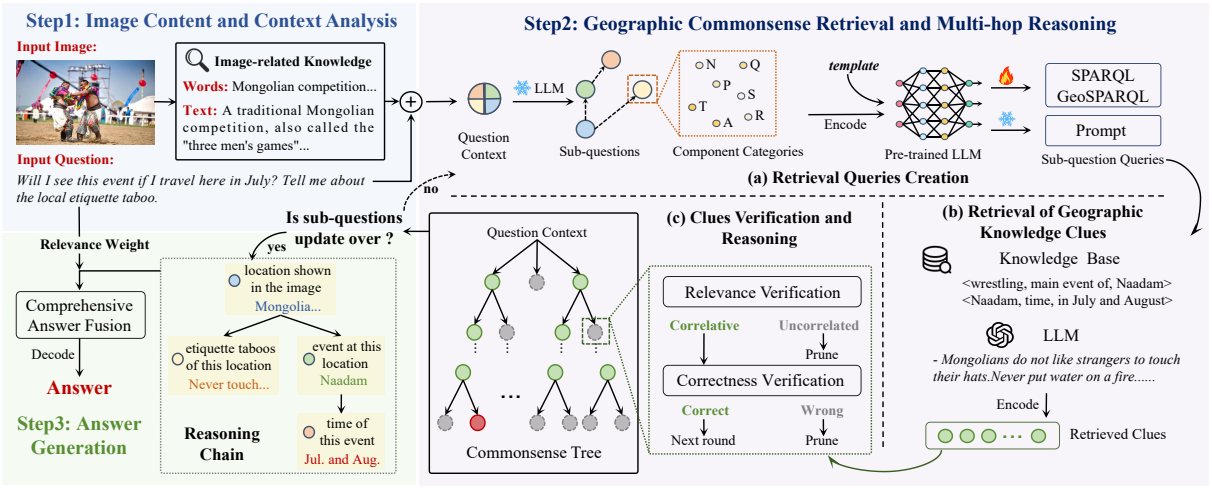


Figure 2: An overview of our proposed framework.

image $m$. As shown in Fig.(2), our approach consists of three steps. First, we analyze the image content and its context. Next, we retrieve related geographic background knowledge for reasoning and form a reasoning chain to the answer. Finally, we integrate this chain with the question to yield the answer. Next, we elaborate on each component.

## 2.1 Image Content and Context Analysis

Since the answer is often hidden in the given image, we first need to understand its content and context. However, traditional work often focuses on capturing the surface content of the image, such as discovering the names, colors of objects, or simple scene descriptions based on the methods of detection, captioning, *LLMs*, etc. That is hard to tackle our task, whose answers involve the deep semantics of images, such as geographical context beyond the surface content. This knowledge is not given in the questions or images. To address this issue, we develop a multi-modal retrieval argumentation technique that can capture this missing but indispensable context of the images. In detail, we search a list of similar images and metadata, including image-related descriptions and keywords from external sources on facts, events, participants, locations, etc. We further perform cross-validation and use a majority voting approach to reduce noise. This verified knowledge is concatenated with the question to form a contextual representation $Q$ as an input to the next module.

## 2.2 Knowledge Retrieval and Reasoning

Based on the questions, we reason over the image's descriptive content to get an answer. Since reasoning for complex questions often involves a large amount of geographic knowledge, we design a dynamic retrieval-reasoning framework. It can adaptively retrieve relevant knowledge from multiple sources and prune the noise to form a reasoning chain, so as to derive a correct answer accordingly.

**Retrieval Queries Creation:** Given that there are many entities and relations involved in the geographic reasoning process, retrieving all external knowledge at one time may be computationally heavy and introduce noise. Thus, we propose to decompose $q \in Q$ into a series of one-hop sub-questions $x \in \mathcal{X}$ using *LLM* like *GPT-4o*. Each $x$ involves some proprietary external knowledge from a variety of sources. We then design a retrieval query $\mathcal{Z}^*$ tailored to $x$. Inspired by Ma et al. (2023), the query generator consists of two steps. We first analyze the sub-question by parsing and then yield a query to retrieve clues. The quality of clues is verified as feedback to refine the queries. The whole generator can be learned by a reinforcement framework. To analyze the sub-question, we use the *Stanford CoreNLP* with *Apache Solr* to carry out *part-of-speech* (*POS*) tagging for $x$ and yield a dependency parse tree in *CoNLL-U* format (Nivre et al., 2016). Referring to the work of Hamzei et al. (2022), we define seven categories based on the content of $x$, represented by the code names: *N* for '*place name*', *T* for '*place type*', *P* for '*properties*', *A* for '*activities*', *S* for '*situations*', *Q* for '*qualities*' and *R* for '*spatial relations*'. These categories are used to help identify the type of sub-question $x$ and the aspects it involves. Based on these categories and aspects, we first generate the corresponding logical

statements composed of terms and functions for $x$. For example, in place-related questions, the terms are places, events, or their properties. Functions are symbols that either declare terms or describe their relations. Similarly, spatio-temporal relations, situation/activity relations, and qualities are defined as functions. These logical statements represent the basic meaning of the sub-question. We match the entities in the logical statements with those in the knowledge base through entity recognition and ontology mapping, and then generate standard queries based on specific templates. Details are provided in Appendix A.

In the reinforced framework, we initialize the policy network with the pre-trained *LLAMA3-8B*, denoted as $\pi_\theta$. We fine-tune it on warm-up data, which consists of sub-questions, predefined templates and target queries. Each template contains slots (strings beginning with an underscore), which are replaced by the matched sentence constituents identified in the sub-question parsing phase. The ground truth comes from the "evidence" field in our dataset. We match the knowledge retrieved by the query against these ground-truth relations based on multiple similarities, like exact matching, semantic matching, and synonym terms matching. To avoid accidental errors caused by using only a single query, we yield multiple queries for each $x$ and the overall set of queries is denoted as $\mathcal{Z}$. Samples confirmed as correct after subsequent filtering are added to the warm-up dataset, denoted as $D_{\text{w}} = \{(x, z^*) \,|\, y = y^*\}$. Using the negative log-likelihood loss as the training objective, for each time step $t$ at the token level, the formula is expressed as Eq.(1), where $T$ denotes the length of $z^*$, $z = \{z_1, z_2, \ldots, z_T\}$ represents the complete query sequence generated according to the policy network $\pi_\theta$, $z_t$ is the token yielded at the $t^{th}$ step.

$$\mathcal{L}_{\text{warm}} = -\sum_{(x,z^*)\in D_{\text{w}}} \sum_{t=1}^{T} \log \pi_\theta(z_t \mid z_{<t}, x). \quad (1)$$

Furthermore, we adopt a policy gradient approach to optimize the policy network. We set the warmed policy network to be the initial policy model $\pi_0$. We aim to maximize the expected reward $J(\theta)$ across the distribution of generated queries. The formula can be expressed as Eq.(2), where $R(z, x)$ is the reward to evaluate $z$.

$$J(\theta) = \mathbb{E}_{z\sim\pi_\theta(z|x)} \left[R(z, x)\right]. \quad (2)$$

We denote the state at $t^{th}$ step as $s_t = (x, z_{<t})$ and the action as $a_t = z_t$. The policy $\pi_\theta(a_t \mid s_t)$ is the probability of generating the token $a_t$ given the state $s_t$. Lastly, the value function $V^\theta(s_t) = \mathbb{E}_{z_{>t}\sim\pi_\theta} \left[R(z, x) \mid s_t\right]$ denotes the expected cumulative reward starting from the state $s_t$. We approximate the value function $V^\theta(s_t)$ with a parameterized network $V_\phi(s_t)$, where $\phi$ denotes the parameters of $V_\phi(s_t)$ that follows the update rule of $\phi \leftarrow \phi - \beta\nabla_\phi\mathcal{L}_V(\phi)$. The loss function for $V_\phi(s_t)$ is defined as Eq.(3), where $G_t = \sum_{k=t}^{T} \gamma^{k-t} r_k$ is the cumulative discount return, $\gamma$ is the discount factor and $\beta$ is the learning rate.

$$\mathcal{L}_V(\phi) = \mathbb{E}_{s_t} \left[(V_\phi(s_t) - G_t)^2\right]. \quad (3)$$

The advantage function $A^\theta(s_t, a_t)$ measures the relative benefit of taking action $a_t$ in the state $s_t$. We utilize the *Temporal-Difference* (*TD*) residuals and the *Generalized Advantage Estimation* (*GAE*) to estimate it (Schulman et al., 2015), which are defined as $\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$ and $A_t^{GAE} = \sum_{l=0}^{T-t} (\gamma\lambda)^l \delta_{t+l}$, respectively. $r_t$ is the immediate reward at $t^{th}$ step and $\lambda$ is the bias-variance trade-off parameter. Therefore, the policy gradient can be derived as Eq.(4).

$$\nabla_\theta J(\theta) = \mathbb{E}_t \left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) \cdot A_t^{GAE}\right]. \quad (4)$$

To further optimize the strategy, we employ the *entropy regularization* and *proximal policy optimization* (*PPO*) algorithm, as Eq.(5), where $\hat{r}_t(\theta)$ is the clipped ratio, $\alpha$ and $\epsilon$ are hyperparameters. $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)}$. $\mathcal{L}_{\text{e}}(\theta) = \alpha \sum_{t=1}^{T} H\left(\pi_\theta(\cdot \mid s_t)\right)$ is the entropy regularization term.

$$\begin{aligned} \mathcal{L}_\theta &= \mathcal{L}_{\text{CLIP}}(\theta) - \mathcal{L}_{\text{e}}(\theta), \\ \hat{r}_t(\theta) &= \text{CLIP}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right), \\ \mathcal{L}_{\text{CLIP}}(\theta) &= -\mathbb{E}_t \left[\min\left(r_t(\theta)A_t, \hat{r}_t(\theta)A_t\right)\right]. \end{aligned} \quad (5)$$

Finally, we formulate the loss function of the model as Eq.(6), which is composed of the policy loss and value loss, where $\lambda_v$ is the value loss coefficient.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_\theta + \lambda_v \mathcal{L}_V(\phi). \quad (6)$$

**Retrieval of Geographic Knowledge Clues:** Based on the created query, we then retrieve the geographic knowledge involved in the questions from some external sources. There are some popular ones, such as *DBpedia* and *OpenStreetMap* (*OSM*) knowledge bases, etc. *DBpedia* provides a broad coverage of geographic commonsense like

capital cities and terrain types. *OSM* excels at offering finer-grained knowledge of spatial geography, including traffic directions and intricate administrative boundaries. To facilitate access *DBpedia* database, we utilize the tool of *Virtuoso endpoint*[1] to retrieve in the form of *SPARQL*. Similarly, we employ the tool *Strabon endpoint* to visit *OSM*. Each retrieved data is represented as an *RDF* triple $k_w = \langle subject, predicate, object \rangle$, such as $\langle wrestling, main\ event\ of, Naadam \rangle$, where the subject and object may be phrases, and the predicate captures a dependency relation. To expand the coverage of our knowledge, we further adopt the *GPT-4o* model to derive some relevant clues. Details of this process are given in Appendix A.

**Clues Verification and Reasoning:** Considering the clues retrieved may be irrelevant to the current reasoning step, this noise can mislead the inference direction or increase the computational cost. To facilitate reasoning, we verify these clues and use them as rewards in the reinforced learning process to reversely optimize the query generator. That is, we create queries for each sub-question of a certain reasoning step, so as to retrieve relevant clues. We then verify them, so as to filter out noise and use this feedback to guide the direction of the next reasoning step. To do it, we define the observation space $O$ for storing the correct clues and the decision space $D$ for deciding whether the current clue is pruned or not. At the $t^{th}$ reasoning step, the model receives $k$ answers $y_t = \{y_t^1, \ldots, y_t^k\}$ to the sub-question $x$ based on the $k$ queries. It then makes a decision $d_t \in \{IsCorrelative, IsAccept\} \subset D$ according to a discriminative strategy. If $D$ returns '*IsAccept*' tag, $O$ records the pairs $o_t = (x_t^i, y_t^i) \in O$, which are used to guide the next decision $d_{t+1}$. To verify the clues, we design a multi-criteria pruning mechanism from both relevance and correctness perspectives. (1) *Relevance*: We evaluate each clue obtained from different sources. We employ semantic similarity to examine the triple-clues retrieved from the geographic knowledge bases. For passage-clues acquired from *LLM* like *GPT-4o*, we explore a *BERT-BM25* hybrid method (Zhang et al., 2023), where the *BM25* score $s_{bm25}$ is computed using the natural language form of clues, $s_{bert}$ denotes the score for calculating semantic similarity based on *BERT* embedding. The final relevance score is calculated as a weighted sum of these two scores, as defined in Eq.(7), where $\tilde{\alpha}$ is the relevance score weight. If $S_{re}$ does not exceed a predetermined threshold $\tilde{\epsilon}$, the clue will be pruned.

$$S_{\text{re}} = \frac{\tilde{\alpha}}{1 + e^{-s_{\text{bm25}}}} + (1 - \tilde{\alpha}) s_{\text{bert}}. \quad (7)$$

(2) *Correctness*: We use $o_{t-1}$ in the previous round as a prefix and input it into the *LLM* (e.g. *GPT-4o*) with the clues in the current round. The *LLM* provides a score $S_{ac}$ for the clue based on our well-designed prompt. Clues that meet the relevance and accuracy thresholds are assigned a positive '*IsAccept*' tag. We take weighted of the relevance and accuracy scores linearly as the reward function. To avoid bias, we standardize the scores by *z-score* and optimize the *RL* algorithm with *soft Q-learning* (Guo et al., 2021). We sample a set of queries from the current batch to calculate the mean and standard deviation, use a sliding window to record the most recent $K$ rewards, and calculate the total mean and standard deviation. The specific formulas are as Eq.(8) and Eq.(9), where $\mathcal{B}$ is the current set of corresponding computations, $\hat{\epsilon}$ is set to ensure numerical stability of the normalization process, $n_{cl}$ is the number of clues retrieved for query $z$, $\omega$ is the weight parameter used to balance the relevance score and accuracy score, $z'$ denotes the queries other than $z$ in the current batch.

$$R(z, x) = \frac{1}{n_{cl}} \sum_{i=1}^{n_{cl}} \left( \omega S_{\text{re}}^{(i)} + (1 - \omega) S_{\text{ac}}^{(i)} \right), \quad (8)$$

$$Z - R(z, x) = \frac{R(z, x) - \text{mean}_{R \in \mathcal{B}} R(z', x)}{\text{stdev}_{R \in \mathcal{B}} R(z', x)} + \hat{\epsilon}. \quad (9)$$

The reward score is then fed back into the query-reinforced learning module. We iterate through this reasoning process until all sub-questions are resolved. Following this step-by-step reasoning process, we can construct a commonsense reasoning tree, with the question context $Q$ as the root node, the observed pairs in $O$ as leaf nodes, and their relations as edges. We then apply a depth-first method to collect pairs with the highest scores at each step, forming the reasoning chain set $\mathbb{E}$.

### 2.3 Answer Generation

Since the reasoning chain can indicate the answer, we combine it with the question to the decoder. That can capture asked points in the question as well as the reasoning logic in the chain to help generate a correct answer. Concretely, we integrate

[1] https://dbpedia.org/sparql

the sub-question-clue pairs in the chain with the relevant parts in the image context $Q$. For each $o_i$, we calculate the correlation between $Q$ and $x_i$. The relevance attention weights are calculated as Eq.(10), where $E_{\text{bert}}(\cdot)$ stands for *BERT* embeddings. $d$ is the dimension of the embeddings. $N$ is the number of split sub-questions.

$$w_{\text{re}}^{(i,j)} = \frac{\exp\left(\frac{E_{\text{bert}}(Q)\cdot E_{\text{bert}}(o_i^j)^\top}{\sqrt{d}}\right)}{\sum_{i'=1}^{N}\sum_{m=1}^{k}\exp\left(\frac{E_{\text{bert}}(Q)\cdot E_{\text{bert}}(o_{i'}^m)^\top}{\sqrt{d}}\right)}. \quad (10)$$

All the $E_{\text{bert}}(o_i)$ are weighted by attention scores $w_{\text{re}}^{(i)}$ then summed and concatenated with $E_{\text{bert}}(Q)$. As shown in Eq.(11), the vector representation of the answer $f_{\text{an}}$ is obtained via a linear transformation $W$. Finally, we employ a transformer-based decoder to generate the answer.

$$f_{\text{an}} = \left[E_{\text{bert}}(Q) \parallel \sum_{i=1}^{N}\sum_{j=1}^{k} w_{\text{re}}^{(i,j)} \cdot E_{\text{bert}}(o_i^j)\right] W. \quad (11)$$

## 3 Experiments

We extensively evaluated the effectiveness of our method with quantitative and qualitative analysis.

### 3.1 Dataset Construction

Existing geographic-related datasets, such as *Geospatial Gold Standard dataset* (Punjani et al., 2018) and *GeoAnQu* (Xu et al., 2020), are mostly used for text-based *QA* instead of *VQA* tasks. Current *VQA* datasets, such as *OK-VQA* (Marino et al., 2019), *A-OKVQA* (Schwenk et al., 2022) and *FVQA* (Wang et al., 2017) are mainly used to evaluate *VQA* task based on image recognition, *InfoSeek* (Chen et al., 2023) is a *VQA* dataset tailored for information-seeking questions that cannot be answered with only commonsense knowledge, they do not involve enough inference of geographical knowledge, so they are not suitable for our *GeoVQA* task. To address this issue, we constructed a large-scale dataset named *GVQA*. In detail, we extracted a total of 29,871 images and created several multi-hop QA pairs for each image, with corresponding reasoning evidence included in each pair. The dataset includes 41,329 QA pairs. Each question requires $k$-hop geographic reasoning, where $k$ is 2 to 4 for most samples. To avoid bias, we partitioned our dataset into *train*, *validation*, *test* sets in an 8:1:1 ratio. The set size was given in Tab.(1).

Table 1: Categorical statistics of dataset *GVQA*.

| Classification | 2-hop | 3-hop | 4-hop | Factual | Flexible |
|---|---|---|---|---|---|
| *Train* | 9986 | 13228 | 10073 | 19972 | 13315 |
| *Val* | 1162 | 1629 | 1084 | 2325 | 1550 |
| *Test* | 1250 | 1718 | 1199 | 2500 | 1667 |

It is not trivial to create a dataset with complex reasoning questions. The questions have to be answerable and follow a suitable reasoning logic. To address this issue, we first collected a series of images depicting landmarks and cultural events from around the world. The collected domains and regions were balanced and representative. We then identified images' concepts corresponding to those in *DBpedia* and geographic databases. We selected semantically meaningful concepts randomly to formulate a candidate question tailored to the geographic domain, each requiring a single reasoning step. We employed an *LLM* like *GPT-4o* to answer these candidate questions. Since complex questions were often associated with entities or answers mentioned in simple questions, we randomly extracted entities from the candidate question and its answer as seeds. Based on them, we then generated a larger question with more reasoning hops. Each generated sub-question, along with its corresponding evidence, was stored as a pair. After 2 to 3 rounds, we can form a complete multi-hop question, where the sub-questions and supporting evidence at each step constitute the intermediate reasoning process. This question format offers many advantages: the questions include clear intermediate steps, making them easy to decompose into sub-questions to verify. In addition, the question was complex, reducing the likelihood of randomly guessing an answer. The dataset was manually verified at the end to ensure rationality. Regarding the security of data, we verified the copyright and license of the images in the dataset. During usage, we adhered to the requirements of these licenses and performed desensitization to ensure no privacy violations. We followed the licenses and did not use it for commercial purposes.

### 3.2 Experimental Settings

We utilized four evaluation metrics that were popular on the generation task, including *ROUGE* (Chin-Yew, 2004), *METEOR* (Banerjee and Lavie, 2004), *BERTScore* (Zhang et al., 2019) and *F1-score*.

Table 2: Performance comparison. The best results are represented in **bold**. The second-best results are underlined.

| Method | ROUGE$_{-2}$ | ROUGE$_{-4}$ | ROUGE$_{-L}$ | METEOR | BERTScore | F1 |
|---|---|---|---|---|---|---|
| *CogVLM* | 21.8±0.2 | 12.5±0.3 | 26.8±0.2 | 24.9±0.3 | 78.2±0.2 | 71.2±0.1 |
| *InstructBLIP* | 23.0±0.2 | 13.2±0.2 | 28.1±0.3 | 23.3±0.2 | 73.9±0.3 | 72.5±0.1 |
| *CiVQA* | 8.30±0.4 | 4.6±0.6 | 13.7±0.4 | 11.1±0.2 | 67.0±0.3 | 62.9±0.2 |
| *LAMOC* | 10.1±0.4 | 5.6±0.4 | 15.0±0.3 | 13.8±0.2 | 60.3±0.2 | 53.9±0.2 |
| *IRCoT* | 15.0±0.1 | 9.3±0.2 | 22.9±0.2 | 20.3±0.3 | 61.8±0.2 | 61.8±0.2 |
| *HT* | 14.2±0.3 | 8.0±0.4 | 20.4±0.4 | 19.0±0.5 | 75.6±0.3 | 63.4±0.3 |
| *FLARE* | 15.9±0.2 | 10.7±0.4 | 23.6±0.3 | 20.2±0.4 | 72.7±0.3 | 64.9±0.2 |
| *LATS* | 17.7±0.3 | 11.2±0.2 | 24.8±0.2 | 23.4±0.4 | 71.4±0.2 | 65.0±0.2 |
| *MORE* | 18.2±0.4 | 10.9±0.3 | 25.4±0.3 | 24.2±0.5 | 70.1±0.3 | 63.6±0.3 |
| *GPT-4o* | 20.4±0.2 | 10.7±0.3 | 25.0±0.1 | 20.8±0.4 | 73.2±0.2 | 68.1±0.2 |
| **Ours** | **28.8**±0.3 | **16.1**±0.2 | **33.2**±0.2 | **30.4**±0.4 | **85.6**±0.3 | **78.9**±0.2 |

They can measure the quality of answers from the perspectives of keyword matching and semantic consistency. For the *ROUGE* metric, we used $ROUGE_{-N}$ to measure the recall based on N-gram overlap, $ROUGE_{-L}$ to evaluate matching based on the *Longest Common Subsequence* (*LCS*). We repeated running 5 times and reported the average performance to reduce bias. Our experiments were conducted using *PyTorch* (Paszke et al., 2019) and were run on four *NVIDIA Tesla V100 GPUs*. Further implementation details and the selection of hyperparameters were provided in Appendix C.

### 3.3 Comparisons Against State-of-the-Arts

We compared our method against ten baselines, including: (1) *CogVLM* (Wang et al., 2023b) and *InstructBLIP* (Dai et al., 2023), typical models leveraging the powerful reasoning capabilities of *LLMs* for downstream applications such as *VQA*; (2) *CiVQA* (Özdemir and Akagündüz, 2024) and *LAMOC* (Du et al., 2023), caption-based models that deduced the answers based on the image description; (3) *IRCoT* (Trivedi et al., 2023) and *FLARE* (Jiang et al., 2023), chain-of-thought-based models that derived answers by iterative inference; (4) *Hypergraph Transformer* (*HT*) (Heo et al., 2022), a graph network-based model which had good performance on multi-hop *VQA*; (5) *LATS* (Zhou et al., 2023) and *MORE* (Cui et al., 2024), innovative models that employed tree structures for reasoning; (6) *GPT-4o* (Achiam et al., 2023), a strong multi-modal *LLM* tool.

As shown in Tab.(2), our model obtained the best performance. It was superior to the *LLMs* baselines. We inferred that while the generic large models have good matching ability, they were still poor at reasoning about knowledge in a certain domain. For other methods, our model significantly surpassed the baselines of *CiVQA* and *LAMOC* in terms of *ROUGE*. This improvement would be due to our usage of *RAG* for image analysis. That provided more accurate descriptions than caption-based approaches, and avoided reasoning on incorrect premises. In addition, our method improved the *METEOR* metric by approximately $11.4\%$ over *HT*. That indicated, compared to graph networks, our iterative retrieval approach could reduce noise to produce more suitable clues. In contrast to methods relying solely on linear reasoning (i.e., *IRCoT* and *FLARE*) or tree-based reasoning (i.e., *LATS* and *MORE*), our approach employed a multi-criteria strategy to prune the irrelevant clues. That gives the model better reasoning capabilities, and helps to avoid the wrong inference direction.

### 3.4 Ablation Studies

To better gain insight into the relative contributions of key components in our approach, we conducted ablation studies on six aspects, including (1) reinforcement learning, where the feedback on evidence quality was removed from influencing subsequent generated queries; (2) adaptive pruning, set the next round to proceed in all reasoning directions from the previous round; (3) multiple knowledge source retrieval, substituted it with separate retrieval from only the knowledge base or *LLM*; (4) multi-factor validation, replaced with only utilizing *LLM* validation; (5) iterative retrieval, replaced by one-time retrieval; and (6) image context analysis, substituted with a caption-based methods to describe the image. As shown in Tab.(3), the ablation on all evaluated components led to a significant performance drop. We found that without adaptive pruning and multi-factor validation, the computational cost was greatly increased and the reasoning accuracy was reduced. Moreover, conducting one-time retrieval resulted in performance degradation in terms of *ROUGE* metric by

more than $4.88\%$. That indicated the dynamic retrieval technique could provide more relevant external knowledge than one-time retrieval. When we changed the image analysis module, the performance declined. We inferred that the retrieval-based approach was more effective than the caption-based approach in understanding the image content, which could provide better semantic contexts for reasoning. Furthermore, we evaluated the performance impact of using different *LLMs* as the policy network. As shown in Tab.(4), the *LLMs* of *LLAMA3-8B* achieved an optimal balance between computational cost and prediction performance.

Table 3: Ablation studies with the metric of drop rates, where $R_{-N}$ refers to the *ROUGE*$_{-N}$ metric, $R_{-L}$ refers to the *ROUGE*$_{-L}$ metric, $M$ refers to the *METEOR* metric, and $B$ refers to the *BERTScore* metric.

| Method | $R_{-2}$ | $R_{-4}$ | $R_{-L}$ | M | B | F1 |
|---|---|---|---|---|---|---|
| w/o Reinforce Learning | ↓5.42 | ↓5.47 | ↓4.49 | ↓5.98 | ↓3.92 | ↓4.84 |
| w/o Image Retrieval | ↓5.19 | ↓5.08 | ↓4.83 | ↓5.59 | ↓3.53 | ↓4.30 |
| w/o Adaptive Pruning | ↓6.81 | ↓7.97 | ↓5.35 | ↓7.03 | ↓4.28 | ↓4.90 |
| w/o LLM Retrieval | ↓4.27 | ↓5.88 | ↓4.51 | ↓5.20 | ↓3.42 | ↓3.69 |
| w/o KnowledgeBase Retrieval | ↓5.33 | ↓6.40 | ↓4.20 | ↓4.83 | ↓3.63 | ↓3.82 |
| w/o Muti-factor Validation | ↓5.68 | ↓6.15 | ↓5.11 | ↓6.57 | ↓4.19 | ↓4.63 |
| w/o Iterative Retrieval | ↓6.22 | ↓7.17 | ↓4.88 | ↓6.49 | ↓3.89 | ↓4.57 |

Table 4: Comparisons on the evaluated *LLMs*.

| Method | $R_{-2}$ | $R_{-4}$ | $R_{-L}$ | M | B | F1 |
|---|---|---|---|---|---|---|
| *LLAMA3-8B* | 28.8 | 16.1 | 33.2 | 30.4 | 85.6 | 78.9 |
| *OPT-6.7B* | 25.3 | 14.8 | 31.4 | 28.8 | 84.5 | 77.2 |
| *T5-3B* | 24.4 | 13.9 | 30.2 | 28.1 | 84.0 | 76.3 |
| *BLOOM-3B* | 21.6 | 12.5 | 27.5 | 26.9 | 83.4 | 75.1 |
| *GPT-2 XL* | 17.7 | 10.7 | 25.3 | 23.1 | 81.9 | 72.6 |

### 3.5 Evaluations on the Trade-off Parameter

Moreover, we evaluated the trade-off parameters in our model by univariate analysis. For each parameter, we tuned it in intervals of 0.2 over the range [0, 1]. When adjusting one parameter, the others were fixed at their optimal values determined from the previous univariate analysis. Afterward, we plotted the performance curves in Fig.(3). We observed that the best result was achieved when the value loss coefficient $\lambda_v$ and reward weighting factor $\omega$ were 0.5, with the relevance score weight $\tilde{\alpha} = 0.3$. That indicated the balanced strategy in terms of strategy loss and value loss, clue relevance, and accuracy could achieve better performance.

### 3.6 Human Evaluations and Analysis

To fully evaluate answers, we conducted human evaluations. We recruited 10 participants with fluent English reading ability, and randomly sampled

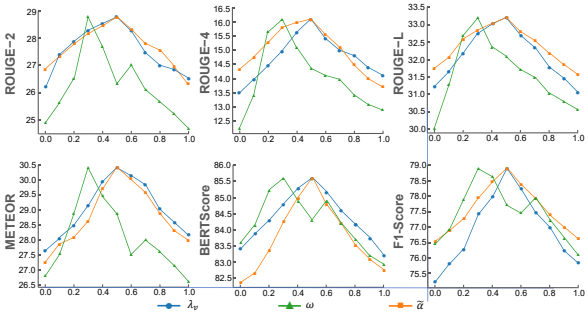


Figure 3: Evaluations of the trade-off parameters.

500 cases from the test results. We asked participants to rate the answers on a scale of 1-10 (with 10 being the best). Each sample was rated by at least 3 participants. The scoring indicators are '*Grammatical Correctness*', '*Language Fluency*', '*Answer Relevance*' and '*Factual Accuracy*'. As shown in Tab.(5), we compared our model with the best baseline. The results show that our model outperforms the best baselines, indicating that the generated answers have good naturalness and accuracy.

Table 5: Results of human evaluation. The best results are represented in **bold**, the second-best is in underlined.

| Model | Grammatical Correctness | Language Fluency |
|---|---|---|
| *CogVLM* | 7.20 | 7.79 |
| *InstructBLIP* | 7.25 | 7.53 |
| *Ours* | **7.81** | **8.56** |
| **Model** | **Answer Relevance** | **Factual Accuracy** |
| *CogVLM* | 6.04 | 6.85 |
| *InstructBLIP* | 6.33 | 7.06 |
| *Ours* | **7.49** | **8.11** |

### 3.7 Case Studies and Discussions

Furthermore, we conducted case studies to analyze the pros and cons of our model. As shown in Fig.(4), our model covered almost all key points. In contrast, *CogVLM*, *GPT-4o*, and *IRCoT* introduced reasoning biases due to incorrect or ambiguous location inference. *CiVQA* and *LAMOC* tended to introduce noise from irrelevant elements in the image, harming performance. *InstructBLIP*, *MORE*, and *LATS* obtained some information about the answers but selected the wrong cities. *FLARE* and *HT* mistakenly interpreted the area of the cities as the drainage area of the rivers. These results further indicated the effectiveness of our model. By analyzing our bad cases, mistakes mainly came from grammatical errors, such as word form and tenses. For instance, the word "*bridge*" in the predicted answer "*two bridge*" should be rectified to "*bridges*",


Figure 4: Case study.

the word "*is*" in sentence "*The building is built in 1900.*" should be rectified to "*was*". These challenges would be studied in future work.

## 4 Related Work

The task of geographic question answering has a wide range of applications. They can assist students in learning geography, be integrated into navigation systems to provide traffic guidance or help travelers explore attractions and local cultures (Mai et al., 2021). Early researches primarily focus on answering text-based geographic questions (Chen, 2014). They often first determine the question type, and then acquire predefined templates tailored to this type to extract answers from the *geographic information systems(GIS)* (Punjani et al., 2018). Later, Xu et al. (2020) and Contractor et al. (2021) introduced geo-analytic QA. It combined multiple knowledge sources to conduct spatial reasoning (Scheider et al., 2021) and was used to solve questions in applications such as education and tourism. Differently, we focus on *GeoQA* task rather than text QA, which requires conducting the cross-modality reasoning over the textual questions and visual image. Our work also relates to the retrieval augmentation technique that can support various tasks (Chen et al., 2017). For this technique, some studies (Yu et al., 2023) propose to optimize the retrieval queries (Jiang et al., 2023), with reinforcement learning (Deng et al., 2022) emerging as a trend in this area (Ma et al., 2023). Other studies (Xu et al., 2023) concentrate on filtering the retrieved clues (Zhao et al., 2023) to provide more accurate information, thereby enhancing the model's reasoning capabilities (Asai et al., 2023).

Other studies are also related to our research, such as knowledge-based multi-hop reasoning and multimodal reasoning, which can tackle some complex reasoning questions. To capture multi-modal context, traditional methods either converted inputs into uniform modality by caption-based methods (Özdemir and Akagündüz, 2024), or integrated multi-modal features (Hu et al., 2023) for fusion. To support multi-hop reasoning, they often used graph network (Sun et al., 2021) or hypergraph (Heo et al., 2022) methods, but that would face challenges in determining the number of nodes and incur high computational costs when dealing with open-domain QA (Wang et al., 2023a). Some work resort to large language models (*LLMs*) (Peng et al., 2023) with chain-of-through (*CoT*). *CoT* can generate intermediate reasoning steps to infer answer (Caffagni et al., 2024). Furthermore, some research (Zhou et al., 2023) employed a tree structure for reasoning (Yao et al., 2024), which can exclude incorrect inference directions and prevent the propagation of irrelevant clues.

## 5 Conclusion

This paper has proposed a new task called *GeoVQA*. It aimed to answer geographic reasoning questions based on the given image. We proposed a new approach with three steps. We first fully analyzed the image content and context based on the multi-modal retrieval augmentation technique. We then deduced from this content and context with respect to the question. To reduce noise, we broke down a complex reasoning process into multiple steps. We adaptively collected the necessary commonsense clues from external knowledge for each step and filtered the noise to derive a reasoning chain. Based on the chain, we decoded the correct answer. Moreover, we created a large-scale dataset *GVQA*, and conducted extensive evaluations on it. The results showed the effectiveness of our model.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (62276279, U22B2060, 62372483, 62306344), Guangdong Basic and Applied Basic Research Foundation (2024B1515020032, 2024A1515010253), Key-Area Research and Development Program of Guangdong Province (2024B0101050005), and Research Foundation of Science and Technology Plan Project of Guangzhou City (2023B01J0001, 2024B01W0004).

## Limitations

The *GeoVQA* task is challenging. It requires not only understanding the geographic context within the image and utilizing multiple knowledge clues for inference, but also generating a fluent and accurate answer. For this task, we proposed an effective framework that adaptively generates answers, but there remains room for further improvement. By analyzing the bad cases in our experimental results, we found that the model could accurately generate the key segments of the answer, while the sections beyond them occasionally exhibit flaws, such as incorrect tense, word forms, or redundancy. For example, when users ask, "*Starting in March, what climate conditions will Tokyo experience over the next month*", the answer should use the future tense. Sometimes it also had mistakes like incorrect use of adverb forms. One way to tackle this problem is to apply a grammar corrector in the post-processing stage to refine the answers more precisely, and we will tackle these issues in future work.

## Ethics Statement

The technique presented in this paper can be applied to provide guidance for questions in travel and navigation domains, thereby facilitating users' lives. Unlike existing methods, our model enhances the accuracy of answers by breaking down complex reasoning processes into multiple steps. That can progressively collect necessary commonsense clues from external knowledge, and eliminate erroneous inference directions. Excluding misuse scenarios, this technology has minimal or no ethical issues. However, as a question-answering technique, the answers it provides could be misused. For instance, malicious users might exploit the technique to obtain real-time location information or travel plans from users for tracking or planning criminal activities. This issue can be mitigated by reminding users to conceal necessary privacy information when posing questions.

## 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*.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Satanjeev Banerjee and Alon Lavie. 2004. Meteor: an automatic metric for mt evaluation with high levels of correlation with human judgments. *Proceedings of the Association for Computational Linguistics-Workshop on Statistical Machine Translation*, pages 65–72.

Davide Caffagni, Federico Cocchi, Nicholas Moratelli, Sara Sarto, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. 2024. Wiki-llava: Hierarchical retrieval-augmented generation for multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1818–1826.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2017. Neural natural language inference models enhanced with external knowledge. *arXiv preprint arXiv:1711.04289*.

Wei Chen. 2014. Parameterized spatial sql translation for geographic question answering. In *IEEE International Conference on Semantic Computing*, pages 23–27, Newport Beach, USA.

Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. 2023. Can pre-trained vision and language models answer visual information-seeking questions? *arXiv preprint arXiv:2302.11713*.

Lin Chin-Yew. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out*.

Danish Contractor, Shashank Goel, Mausam, and Parag Singla. 2021. Joint spatio-textual reasoning for answering tourism questions. In *Proceedings of the Web Conference*, pages 1978–1989, Ljubljana, Slovenia.

Wanqing Cui, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2024. More: Multi-modal retrieval augmented generative commonsense reasoning. *arXiv preprint arXiv:2402.13625*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang,

Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P Xing, and Zhiting Hu. 2022. Rlprompt: Optimizing discrete text prompts with reinforcement learning. *arXiv preprint arXiv:2205.12548*.

Yifan Du, Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Zero-shot visual question answering with language model feedback. *arXiv preprint arXiv:2305.17006*.

Han Guo, Bowen Tan, Zhengzhong Liu, Eric P Xing, and Zhiting Hu. 2021. Efficient (soft) q-learning for text generation with limited good data. *arXiv preprint arXiv:2106.07704*.

Ehsan Hamzei, Martin Tomko, and Stephan Winter. 2022. Translating place-related questions to geosparql queries. *Proceedings of the ACM Web Conference*.

Yu-Jung Heo, Eun-Sol Kim, Woo Suk Choi, and Byoung-Tak Zhang. 2022. Hypergraph transformer: Weakly-supervised multi-hop reasoning for knowledge-based visual question answering. *arXiv preprint arXiv:2204.10448*.

Ziniu Hu, Ahmet Iscen, Chen Sun, Zirui Wang, Kai-Wei Chang, Yizhou Sun, Cordelia Schmid, David A Ross, and Alireza Fathi. 2023. Reveal: Retrieval-augmented visual-language pre-training with multi-source multimodal knowledge memory. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23369–23379.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models. *arXiv preprint arXiv:2305.14283*.

Gengchen Mai, Krzysztof Janowicz, Rui Zhu, Ling Cai, and Ni Lao. 2021. Geographic question answering: challenges, uniqueness, classification, and future directions. *AGILE: GIScience Series*, 2:8.

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, Long Beach, United States.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pages 1659–1666, Portorož, Slovenia.

Övgü Özdemir and Erdem Akagündüz. 2024. Enhancing visual question answering through question-driven image captions as prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1562–1571, Shanghai, China.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. 2023. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*.

Dharmen Punjani, Kuldeep Singh, Andreas Both, Manolis Koubarakis, Iosif Angelidis, Konstantina Bereta, Themis Beris, Dimitris Bilidas, Theofilos Ioannidis, Nikolaos Karalis, et al. 2018. Template-based question answering over linked geospatial data. In *Proceedings of the 12th Workshop on Geographic Information Retrieval*, pages 1–10.

Simon Scheider, Enkhbold Nyamsuren, Han Kruiger, and Haiqi Xu. 2021. Geo-analytical question-answering with gis. *International Journal of Digital Earth*, 14(1):1–14.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

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*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, Toronto, Canada.

Jinyuan Wang, Junlong Li, and Hai Zhao. 2023a. Self-prompted chain-of-thought on large language models for open-domain multi-hop reasoning. *arXiv preprint arXiv:2310.13552*.

Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton Van Den Hengel. 2017. Fvqa: Fact-based visual question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(10):2413–2427.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. 2023b. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*.

Haiqi Xu, Ehsan Hamzei, Enkhbold Nyamsuren, Han Kruiger, Stephan Winter, Martin Tomko, and Simon Scheider. 2020. Extracting interrogative intents and concepts from geo-analytic questions. *AGILE: GI-Science Series*, 1:23.

Yan Xu, Mahdi Namazifar, Devamanyu Hazarika, Aishwarya Padmakumar, Yang Liu, and Dilek Hakkani-Tür. 2023. Kilm: Knowledge injection into encoder-decoder language models. *arXiv preprint arXiv:2302.09170*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. 2023. Augmentation-adapted retriever improves generalization of language models as generic plug-in. *arXiv preprint arXiv:2305.17331*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yufeng Zhang, Meng-Xiang Wang, and Jianxing Yu. 2023. Answering subjective induction questions on products by summarizing multi-sources multi-viewpoints knowledge. In *IEEE International Conference on Data Mining*, pages 848–857, Shanghai, China.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. *arXiv preprint arXiv:2305.03268*.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2023. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*.

# A Details of Our Approach

## A.1 Implementation Details

For the image analysis module with the multimodal retrieval enhancement, we extracted a short summary of each image page and three image-related keywords, such as facts, events, participants, locations, etc. In the dynamic iterative retrieval inference module, we set $\gamma = 0.98$ and use the *Adam* optimizer with $\beta_{policy} = 2e^{-5}, \beta_{value} = 1e^{-4}$. For the hyperparameters in policy optimization, we set the entropy regularization coefficient $\alpha = 0.05$ and the *PPO* clipping coefficient $\epsilon = 0.2$. The *Bias-variance* trade-off parameter $\lambda$ in *GAE* is set to 0.95. The loss weight $\lambda_v$ of the value function is set to 0.5. To better filter the relevant clues, we set the $\tilde{\alpha}$ of the balance to 0.3. The threshold $\tilde{\epsilon}$ is set to 0.95. Reward weighting factor $\omega = 0.5$. In order to leverage *LLM* to better retrieve external knowledge, we first decomposed the question step by step into sub-questions that are easier to retrieve and reason about. Meanwhile, we designed useful prompts to ensure the retrieved clues could meet the following criteria: (1) Only contain the clues that are directly relevant to the question, without adding irrelevant information. (2) Accurate information is provided. (3) Responses should be expressed in a concise, clear, and understandable manner that highlights the main message. The *GPT-4o*-based prompts are shown in Fig.(5) and Fig.(6), respectively.

```
prompt = """
You are an intelligent assistant capable of breaking down complex questions
into simple, logically coherent sub-questions. Please follow these guidelines:
1. Each sub-question should require only one step of reasoning.
2. Each sub-question should depend on the previous question and its
potential answer, ensuring there is clear logical progression between them.
You can simulate the answers to each sub-question to ensure they are
logically consistent and help lead to the final answer.
3. Ensure all necessary information is included while keeping the language
concise and accurate.
4. List the sub-questions in numbered order without additional introductory
phrases.
5. Verify the feasibility of each sub-question, ensuring that each one can be
independently answered and supports the reasoning for subsequent sub-
questions.

Question: {question}
Answer:
"""
```

Figure 5: Prompt for the question decomposition.

To verify clues, we designed a prompt based on *GPT-4o* by considering five aspects: validity, accuracy, completeness, logical clarity, and objectivity. The clues with scores above the threshold of 8 were viewed as good. The specific evaluation criteria and related scores were shown in Fig.(7).

```
prompt = """ You are an intelligent assistant with access to a wide range of
knowledge. Based on external information: {retrieved clues} and the
question: {question}, integrate these external information to form a clear,
concise and accurate response.
Answer:
"""


response = openai.ChatCompletion.create(
        model="gpt-4o",
        messages=[
            {"role": "system", "content": "You are a knowledgeable assistant."},
            {"role": "user", "content": prompt.format(retrieved_clues=clues,
                question=question)}
        ],
        max_tokens=500,
        temperature=0.4,
)
```

Figure 6: Prompt for the question context retrieval.

```
prompt = """
You are an intelligent assistant specialized in fact-checking and validating information.
Your task is to assess the correctness and reliability of the following knowledge clue based
on the given question.

**Question:** {question}
**Knowledge Clue:** {knowledge_clue}

**Instructions:**
1. **Effectiveness:** Determine if the knowledge clue directly answers the question
without evading or misinterpreting the question.
2. **Accuracy:** Verify the factual correctness of the information provided in the
knowledge clue.
3. **Completeness:** Assess whether the knowledge clue provides a comprehensive
answer or if it lacks essential details.
4. **Clarity:** Ensure the knowledge clue is clearly written and logically structured.
5. **Objectivity:** Check that the information is presented objectively without undue bias.

**Scoring Criteria (1-10):**
- **1-2:** The knowledge clue is mostly incorrect, irrelevant, and lacks clarity.
- **3-4:** The knowledge clue has significant inaccuracies or is only partially relevant.
- **5-6:** The knowledge clue is somewhat relevant and mostly accurate but lacks
completeness or clarity.
- **7-8:** The knowledge clue is relevant, accurate, and mostly complete with minor
omissions.
- **9-10:** The knowledge clue is highly relevant, completely accurate, thorough, and
clearly presented.

**Response Format:**
- **Score:** [1-10]
"""
response = openai.ChatCompletion.create(
        model="gpt-4o",
        messages=[
            {"role": "system", "content": "You are a fact-checking assistant."},
            {"role": "user", "content": validation_prompt.format(question=question,
knowledge_clue=knowledge_clue)}
        ],
        max_tokens=400,
        temperature=0.3,
    )
```

Figure 7: Prompt for clue verification.

## A.2 Details of Query Templates

Following previous work of Hamzei et al. (2022), *GeoSPARQL* queries consist of several components, including prefixes, *ASK/SELECT* statements, and the *WHERE* clause. The prefixes utilize a predefined set of prefixes that define the namespaces for accessing the knowledge base, ontology, and implementation functions. The *ASK/SELECT* statements determine the query output, and the *WHERE* clause captures the conditions specified in the query. In detail, the conversion of logical statements into *GeoSPARQL* queries involves three sequential steps: (1) The overall query structure (e.g., *ASK/SELECT* query) is determined based on the extracted intent. In the case of a *SELECT* query, the intent specifies which variables to retrieve; (2) The *WHERE* clause is dynamically generated through logical connections; and (3) Ordering and aggregation (*ORDER BY* and *GROUP BY* clauses) are generated for queries that require them. Given the Question 2 in Appendix Fig.(9) as an example, Fig.(8) is the predefined rule templates for generating queries, and Fig.(9) shows an example of the target query.

Table 6: Categories of the question components.

| Code | Name | Explanation | Example |
|---|---|---|---|
| *N* | place name | a direct reference to a geographic place. | New York |
| *T* | place type | a generic reference to a category of a taxonomy that captures places with similar functional, spatial, and physical properties. | mountain |
| *P* | properties | describe diverse characteristics of places and a place may be described by a set of criteria imposed on these properties. | population |
| *A* | activities | afforded by places, and places may be queried for their affordances. | [a place] to buy hardware |
| *S* | situations | another way to describe places by reference to what is available or can be experienced there, instead of what one can do there. | [a place] to see birds |
| *Q* | qualities | quality of an activity, a situation, or a property of place that narrows down the search domain for identifying relevant places. | the most populated city, the old building, the best cafe |
| *R* | spatial relations | describe how places are located in relative space and include a diverse set of topological, directional, and metric relations. | inside, north of, within 200 meters |

**Place definition template using name**

```
VALUES ?<PI> {<URIS>}.
 ?<PI> geosparql : hasGeometry ?<PI>G .
<PI>G geosparql : asWKT ?<PI>GEOM .
```

**Place definition template using type**

```
?<PI> rdf : type ?<PI>TYPE;
      geosparql : hasGeometry ?<PI>G .
?<PI>G geosparql : asWKT ?<PI>GEOM .
VALUES ?<PI>TYPE {<URIS>} .
```

**Attribute relation template**

```
VALUES ?<ATTRIBUTE> {<URIS>} .
?<PI> ?<ATTRIBUTE> ?<PROPERTY> .
```

**Distance relation template**

```
FILTER(geof : distance(?<PI1>GEOM, ?<PI2>GEOM, <UNIT>) < <DISTANCE>) .
```

**Aggregation template**

```
GROUP BY <VARIABLE>
HAVING (COUNT(*) > <LIMIT>)
```

**Sorting template**

```
ORDER BY <ASC/DESC> (<VARIABLES>) LIMIT <LIMIT> .
```

Figure 8: Predefined templates for queries.

## A.3 Details of the Warm-up Dataset

Since the data in *OSM* cannot be directly used for retrieval, we use its interlinked *Geospatial*

*Gold Standard* dataset[2] to assist in constructing our warm-up dataset. It can interlink classes with the same or very similar labels in *DBpedia* and *OSM*, extract valid information from *OSM* in RDF format, and then construct *GeoSPARQL* queries for each geographic question. Considering the amount of data contained in the *Geospatial Gold Standard* dataset may be insufficient, it may not meet the requirements of our task. We use it as the warm-up dataset for initial training. Later, the valid results after verification will be added to enlarge it. The specific standard query samples in the warm-up dataset are shown in Fig.(9).

**Question1: Which pubs in Dublin are near Guinness Brewery?**

```
SELECT DISTINCT ?pub
WHERE {
  ?pub a osm:Pub ;
     geo:hasGeometry ?pubGeo .
  ?pubGeo geo:asWKT ?pubWKT .

  ?dublin osm:has_name ?l2 ;
        geo:hasGeometry ?dublinGeo .
  ?dublinGeo geo:asWKT ?dublinWKT .

  ?guiness osm:has_name ?l ;
        geo:hasGeometry ?guinessGeo .
  ?guinessGeo geo:asWKT ?guinessWKT .

  FILTER(STR(?l2) = "Dublin") .
  FILTER(STR(?l) = "Guinness Brewery") .
  FILTER(geof:distance(?guinessWKT, ?pubWKT, uom:metre) < 1000) .
  FILTER(geof:sfContains(?dublinWKT, ?pubWKT)) .
}
```

**Question2: Is there a church at most 2km from the University of Westminster?**

```
ASK {
  ?s2 linkedeodata:has_name "University of Westminster"^^xsd:string .
  ?s2 rdf:type osm:University .
  ?s2 geo:hasGeometry ?geo2 .
  ?geo2 geo:asWKT ?w2 .

  ?s3 geo:hasGeometry ?geo3 .
  ?s3 linkedeodata:has_name ?name .
  ?geo3 geo:asWKT ?w3 .
  ?s3 rdf:type osm:Church .

  FILTER(geof:distance(?w3, ?w2, uom:metre) < 2000)
}
```

Figure 9: Standard query samples of warm-up dataset.

## B Details of Our *GVQA* Dataset

We constructed a large-scale dataset *GVQA* tailored for our task. We extracted a total of 29,871 images from multiple search engines, which were publicly available on the Web. Then we provided several geographic QA pairs with corresponding reasoning evidence for each image. The content involves geographic knowledge related to landmarks and human activities, such as location, climate characteristics, terrain, national culture, customs, etc. These samples all need to understand the geographic background of the image to reason, and cannot be simply solved by identification and matching alone. Some of these can be answered by extracting information from the corresponding knowledge base, other examples require a deeper understanding to deduce abstractive answers.

[2] https://geoqa.di.uoa.gr/

## C Settings of All Evaluated Methods

We outlined the parameter settings for the methods utilized in our evaluations.

**Settings of Our Model:** Our experiments were conducted using *PyTorch* (Paszke et al., 2019) and were run on four *NVIDIA Tesla V100* GPUs. We leveraged the *BERT*-base model to initialize the word embeddings and employed the transformer-based *GPT-2* medium as the decoder. We generated 3 queries for each sub-question, and each query retained 3 clues threads after filtering. Accordingly, set the sliding window size $K = 500$. We used a linear learning rate scheduler with 3,000 warm-up steps. Gradients were clipped if their norm exceeded 1.0, and the weight decay for all bias-free parameters was set to 0.0001. We trained for up to 6,000 steps, and validated every 200 steps, with early stopping after one round of no improvement in validation loss. It takes about 11 hours to train the model and 6 minutes to test 1,000 samples in the test set.

**Settings of *CogVLM*:** The model was trained for a total of 6000 steps. The batch size was set to 32 per GPU and the number of gradient accumulation steps was set to 8. The input resolution was set to 384×384, and dropout was applied with a probability of 0.1. The model parameters were optimized using *AdamW* with a learning rate of $1e^{-5}$, $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\varepsilon = 1 \times 10^{-8}$, and a weight decay of 0.1.

**Settings of *InstructBLIP*:** We froze the image encoder and the large language model on the backend, only updating the *Q-Former* module that connected images to text. The vocabulary size was set to 30,522 and the hidden size was set to 768. The *Transformer* encoder consisted of 12 hidden layers and 12 attention heads, with an intermediate layer size of 3,072. The hidden activation function was `gelu`, and both the hidden and attention dropout probabilities were set to 0.1. The maximum number of position embeddings was 512, and the initializer range was 0.02. The layer normalization epsilon was set to $1e^{-12}$, and the padding

token ID was 0. The position embedding type was chosen as `absolute`, while a cross-attention frequency of 2 was employed, and the encoder hidden size was 1,408. The image resolution remained at 384×384.

**Settings of *CiVQA*:** The maximum number of generated tokens was set to 5, and the temperature parameter was set to 0.2.

**Settings of *LAMOC*:** The officially released *BLIP* captioning checkpoint[3] was used for model initialization. Both captioning adaptation and reinforcement learning were conducted with the following hyperparameters: the learning rate was set to $2 \times 10^{-6}$, and warmup was performed for 600 steps. A weight decay of 0.05 was applied, and the batch size was set to 8. The balance factor $\alpha$ was configured to 0.9. Model training lasted for 10 epochs.

**Settings of *IRCoT*:** Training used a batch size of 8 per GPU with *AdamW* optimization at a learning rate of $2 \times 10^{-6}$, warm-up steps of 600, and a weight decay of 0.05. Mixed precision (FP16) was applied, gradient clipping was set to $[-5, 5]$, and inference employed a beam size of 6.

**Settings of *FLARE:*** The training configuration entails a batch size of 16, a learning rate of $1e^{-5}$, a beam size of 5, a maximum sequence length of 512, and an image resolution of 384 for ten epochs.

**Settings of *Hypergraph Transformer*:** The model was trained using stochastic gradient descent (*SGD*) as the optimizer, with a batch size of 64. The word embedding dimension was set to 128. The learning rate was fixed at 0.0001, with gradient clipping applied at a threshold of 10. A dropout rate of 0.05 was used to prevent overfitting.

**Settings of *LATS*:** We used *Adam* as the optimizer and the global batch size was set to 16. We used a learning rate of $1e^{-5}$ and utilized a cosine annealing learning strategy with an initial learning rate $1e^{-5}$ and a final learning rate of 0 after 150 epochs. The maximum sequence length was set to 512.

**Settings of *MORE*:** The model was trained using the *AdamW* optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 0.05. The batch size was selected from $\{64, 128\}$. Training was conducted for at most 20,000 steps, including a 1% warm-up period. For retrieval augmentation, an additional $T = 2000$ steps were used with query dropout, and the noisy retrieval augmentation input ratio $\hat{p}$ was set to 0.3. The learning rates for the task prompt and retrieval augmentation prompt were selected from $\{1e^{-4}, 5e^{-4}, 1e^{-3}\}$ and $\{1e^{-5}, 3e^{-5}\}$, respectively. During decoding, a beam search with a size of 5 was employed.

[3] https://storage.googleapis.com/sfr-vision-language-research/BLIP/models/model_large_caption.pth

**Prompt for answering the first step questions**

"role": "system", "content": "You are a expert in geography knowledgeable."

"role": "user", "content": f"Provide the answer to the following question based on the information about {landmark_name} located in {location}: {Question1}. Be concise and direct. Provide only the answer, and keep it as concise as possible without additional sentences."

**Prompt for asking the second step questions**

```
"role": "system",
      "content": (
        "You are an assistant that helps generate a second subquestion (Q2) based on a given initial subquestion (Q1) and its answer (A1). "
        "Your goal is to create Q2 such that it:\n"
        "1. Derives logically from the given Q1 and A1.\n"
        "2. Maintains logical progression: Q2 should be a natural follow-up to Q1.\n"
        "3. Involves only one step of reasoning, without requiring multiple inference steps.\n"
        "4. Uses concise language, focusing on essential logical aspects from Q1 and A1.\n"
        "5. Avoids repeating unnecessary known details or re-stating trivial facts.\n"
        "6. Q2 should not introduce unrelated information.\n"
        "\nIn other words, given Q1 and its answer A1, use the key entity or main concept from A1 to form a second question (Q2) that logically follows Q1 and A1, but does not repeat already established facts.\n"
    )

"role": "user",
      "content": (
        "Q1: '{Q1}'\n"
        "A1: '{A1}'\n\n"
        "Now, based on Q1 and A1, please generate a concise Q2. "
        "Make sure Q2 logically follows from A1, focuses on a key entity or concept mentioned, and requires only one reasoning step."
    )
```

**Prompt for answering the second step questions**

"role": "system", "content": "You are a helpful geography assistant who creates follow-up questions based on the provided question and answer."

"role": "user", "content": f"Given the known information about {landmark_name} located in {location}, and the current reasoning chain question '{Q1}' and its answer '{A1}', answer the question '{Q2}' : Be concise and direct. Provide only the answer, and keep it as concise as possible without additional sentences."

**Prompt for integrating multi-hop questions**

```
"role": "system",
        "content": (
            "You are an assistant specialized in generating multi-hop questions for question-answering datasets. "
            "Given two subproblems, Q1 and Q2, your task is to seamlessly combine them into a single, coherent multi-hop question. "
            "Ensure that the combined question does not mention any specific landmark names, cities, or locations. "
            "Avoid making any assumptions beyond the provided subproblems. "
            "The generated multi-hop question should be purely interrogative, concise, logically structured, and focused on reasoning.\n\n"
            "Guidelines:\n"
            "1. Do not include any specific landmark names or location details that are not present in Q1 or Q2.\n"
            "2. Maintain the logical flow between the subquestions without introducing external information.\n"
            "3. Ensure clarity and conciseness in the phrasing of the multi-hop question.\n"
            "4. The question should naturally integrate the information from both subproblems.\n\n"
            "Examples for reference:\n"
            "Example 1:\n"
            "Q1: 'Is the type of trees in this area coniferous or broad-leaved?'\n"
            "Q2: 'What are some common examples of broad-leaved trees found in this area?'\n"
            "Combined Multi-hop Question: Are the trees in this area coniferous or broad-leaved? What are the trees in this area that belong to this type?\n\n"
            "Example 2:\n"
            "Q1: 'What types of public transportation are available in this city?'\n"
            "Q2: 'How extensive is the bus network in Ringsjön, Sweden, in terms of coverage and frequency of service?'\n"
            "Combined Multi-hop Question: What public transport options are available in the city, and what is the
```

Figure 10: Prompt designed for *GVQA* datasets. We show a sample prompt for generating 3-hop questions. For generating 4-hop questions, we need to set the inference steps of the sub-questions to 2 in the template for yielding the questions in the second step.

**Prompt for integrating multi-hop questions**

```
coverage and service frequency of this public transport?\n\n"
            "Example 3:\n"
            "Q1: 'What is the local time zone?'\n"
            "Q2: 'What are the typical differences in hours between Japan Standard Time and Coordinated Universal Time (UTC)?'\n"
            "Combined Multi-hop Question: What's the local time zone? What is the difference between this time zone and Coordinated Universal Time (UTC)?\n"
        )

"role": "user",
        "content": (
            "Now combine these subproblems: Q1: '{Q1}', Q2: '{Q2}' to generate a multi-hop question. "
            "Ensure that the entire content of the first question is preserved. For the subsequent questions, omit any information not covered in the first question and integrate them into a coherent multi-hop question. "
            "Note:\n"
            "1. Do not mention any specific landmark names or locations.\n"
            "2. Do not make any assumptions beyond the provided subproblems.\n"
            "3. The language should be concise, clear, and easy to understand."
        )
```

**Prompt for answering multi-hop questions and providing evidence**

```
"role": "system",
        "content": (
            "You are an assistant specialized in generating comprehensive answers and relevant evidence for multi-hop questions in a question-answering dataset. "
            "Given two subproblems, Q1 and Q2, along with their respective answers, A1 and A2, your task is to:\n"
            "1. Generate a clear and concise answer to the combined multi-hop question.\n"
            "2. Provide a list of evidence tuples derived from the subproblems and their answers.\n\n"
            "Guidelines:\n"
            "1. The answer should integrate information from both A1 and A2 to comprehensively address the multi-hop question.\n"
            "2. The evidence should be a list of tuples in the format (Entity, Aspect, Detail), where 'Entity' can be any relevant subject from the answers.\n"
            "3. Ensure that all evidence is solely based on the provided subanswers without introducing any new or assumed information.\n"
            "4. Focus on logical relationships that support the reasoning in the answer.\n"
            "5. Maintain clarity and conciseness in both the answer and the evidence.\n\n"
            "Example:\n"
            "Q1: 'Is this a man-made landscape or a natural landscape?'\n"
            "A1: The Twelve Apostles in Victoria, Australia, is a natural landscape.\n"
            "Q2: 'Can I see polar bears here?'\n"
            "A2: Polar bears are native to the Arctic region, in the Northern Hemisphere, and not found in Australia.\n"
            "Multi-hop Question: 'Is this a man-made landscape or a natural landscape, and can I see polar bears here?'\n"
            "Answer: The Twelve Apostles is a natural landscape, not man-made. Polar bears are native to the Arctic and not found in Australia, so you cannot see them here.\n"
            "Evidence: (The Twelve Apostles, Landscape Type, Natural), (Polar Bears, Native Location, Arctic Region), (Australia, Hemisphere, Southern Hemisphere)."
        )

"role": "user",
        "content": (
            f"Given the following subproblems and subanswers:\n\n"
            f"Q1: '{Q1}'\n"
            f"A1: '{A1}'\n\n"
            f"Q2: '{Q2}'\n"
            f"A2: '{A2}'\n\n"
            "Generate a comprehensive answer to the multi-hop question and provide relevant evidence.\n"
            "The evidence should be in the format (Entity, Aspect, Detail), where 'Entity' can be any relevant subject from the answers.\n"
            "Ensure that the answer is clear, concise, and integrates information from both subanswers.\n"
            "Focus on the logical relationships that support the reasoning in the answer."
        )
```

Figure 11: Prompt designed for *GVQA* datasets.