REVIVE: Regional Visual Representation Matters in Knowledge-Based Visual Question Answering

Yuanze Lin[♣*] **Yujia Xie**[♠] **Dongdong Chen**[♠]
Yichong Xu[♠] **Chenguang Zhu**[♠] **Lu Yuan**[♠]
[♣] University of Washington [♠]Microsoft
yuanze@uw.edu
{yujiaxie, dochen, yicxu, chezhu, luyuan}@microsoft.com

Abstract

This paper revisits visual representation in knowledge-based visual question answering (VQA) and demonstrates that using regional information in a better way can significantly improve the performance. While visual representation is extensively studied in traditional VQA, it is under-explored in knowledge-based VQA even though these two tasks share the common spirit, i.e., rely on visual input to answer the question. Specifically, we observe that in most state-of-the-art knowledge-based VQA methods: 1) visual features are extracted either from the whole image or in a sliding window manner for retrieving knowledge, and the important relationship within/among object regions is neglected; 2) visual features are not well utilized in the final answering model, which is counter-intuitive to some extent. Based on these observations, we propose a new knowledge-based VQA method ***REVIVE***, which tries to utilize the explicit information of object regions not only in the knowledge retrieval stage but also in the answering model. The key motivation is that object regions and inherent relationships are important for knowledge-based VQA. We perform extensive experiments on the standard OK-VQA dataset and achieve new state-of-the-art performance, *i.e.*, **58.0%** accuracy, surpassing previous state-of-the-art method by a large margin (**+3.6%**). We also conduct detailed analysis and show the necessity of regional information in different framework components for knowledge-based VQA.

1 Introduction

Many vision-based decision making processes in our daily life go beyond perception and recognition. For example, if we see a salad bow in the deli bar, our decision on whether to buy it does not only depend on what is in the bow, but also the calories in each of the item. This motivates the knowledge-based Visual Question Answering (VQA) task [18], which extends traditional VQA task [1] to solve more complex problems, *i.e.*, where commonsense knowledge is required to answer the open-domain questions.

By definition, knowledge-based VQA takes three different information sources to predict the answer: input visual information (image), input question, and the external knowledge. While existing research on knowledge-based VQA mainly focuses on improving the incorporation of external knowledge, this paper focuses on improving the object-centric visual representation and presents a comprehensive empirical study to demonstrate that visual features matter in this task.

Intuitively, visual information should be well used for both knowledge retrieval and final answering. However, we find existing state-of-the-art (SOTA) methods [32, 5] have not fully utilized it. On the

[*] Work done during an internship at Microsoft Redmond.

Preprint. Under review.

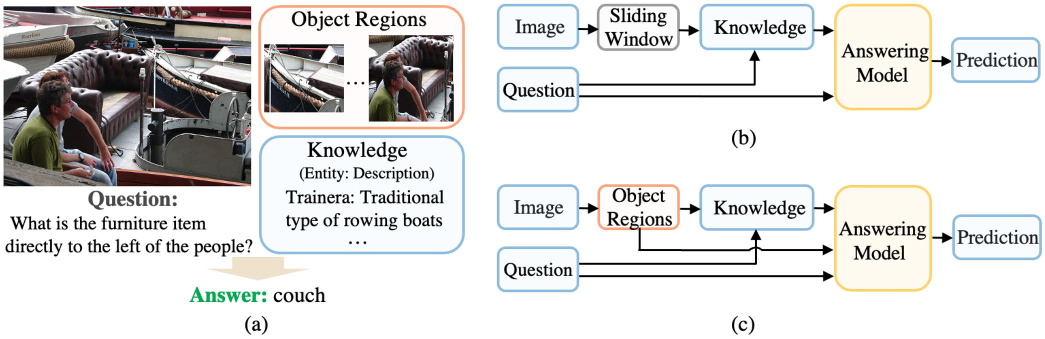

Figure 1: (a) An example from OK-VQA dataset, our method utilizes the retrieved knowledge and object-centric regions to solve the question. (b) The pipeline of previous state-of-the-art method KAT [5]. (c) The pipeline of our proposed *REVIVE*.

one hand, they simply use either the whole image or a sliding window on the image to retrieve the external knowledge. On the other hand, they only use visual information for knowledge retrieval but ignore it in the final answering model. In other words, they fuse only the retrieved knowledge and the question as a pure natural language processing (NLP) model to obtain the answer, a typical method [5] is illustrated in Figure 1 (b).

In this paper, we revisit visual representation in knowledge-based VQA, and argue that the information of object regions and their relationship should be considered and used in a dedicated way. The underlying motivation is shown in Figure 1 (a), which demonstrates that understanding the objects and their relationship is necessary. To this end, we propose **REVIVE** to better utilize **RE**gional **VI**sual Representation for knowledge-based **V**isual qu**E**stion answering. It not only exploits the detailed regional information for better knowledge retrieval, but also fuses the regional visual representation into the final answering model. Specifically, we first use the object detector GLIP [12] to locate the objects, and then use the cropped region proposals to retrieve different types of external knowledge. Finally, we integrate the knowledge together with the regional visual features into a unified transformer based answering model for final answer generation.

We perform extensive experiments on the OK-VQA dataset [18], and the proposed *REVIVE* achieves the SOTA performance of **58.0%** accuracy, a **3.6%** absolute improvement from the results of previous SOTA method [5].

We summarize our contribution as follows:

(a) We systematically explore how to better exploit the visual feature to retrieve knowledge. The empirical results suggest the region-based approach performs the best, compared to whole image-based and sliding window-based approaches.

(b) We integrate the regional visual representation, retrieved external and implicit knowledge into a transformer-based question answering model, which can effectively leverage the three information sources for solving knowledge-based VQA.

(c) Our proposed REVIVE achieves the state-of-the-art performance on OK-VQA dataset, *i.e.*, **58.0%** accuracy, surpassing the previous methods by a large margin.

2 Related Work

Knowledge-Based VQA. Knowledge-based VQA [18] aims to predict answers for general questions by leveraging external knowledge beyond image content. Early works [29, 28] introduce external knowledge to solve visual question answering (VQA) tasks. OK-VQA dataset [18] is the first large-scale dataset with questions that need be answered using external knowledge instead of a provided fixed knowledge base [28]. Recent studies [29, 28, 20, 19, 35, 31, 17, 32, 5] integrate different knowledge from various external knowledge resources, *e.g.*, ConceptNet [23], Wikipedia [26], *etc*, for solving knowledge-based VQA. Later, PICa [32] regards large language models, e.g., GPT-3 [2] as an implicit knowledge source and employs it [2] to get answer prediction based on textual

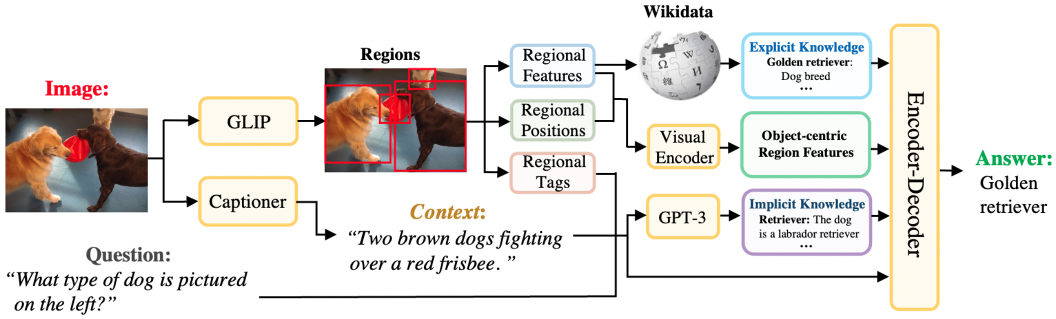

Figure 2: **The illustration of *REVIVE*.** It exploits regional information (*i.e.*, features, positions and tags), question and context to retrieve different types of knowledge. In addition, it also incorporates learned object-centric region features with retrieved knowledge for answer generation.

prompts. Inspired by the recent success of knowledge-retrieved methods [7, 6] that leverage external knowledge retrieval with language generative models for open-domain question answering, KAT [5] exploits the FiD reader [7] to perform knowledge reasoning over retrieved implicit and explicit knowledge. Our work instead emphasizes revisiting the visual representation for knowledge retrieval, *i.e.*, resorting to regional visual representation. In addition, we propose to incorporate object-centric regional visual representation together with retrieved knowledge in the answer generative model.

Vision-Language Models. Recent years have witnessed the rapid development of vision-language models [25, 8, 33, 24, 13, 34, 27, 30]. Those works usually first pre-train a neural network on a large-scale image-text dataset and then finetune the models for solving specific vision-language tasks. Among them, VinVL [34] aims to learn the object-centric representation. CLIP [21] pre-trains the models with large-scale text-image pairs by contrastive learning. GLIP [33] reformulates the pre-training process by unifying object detection and phrase grounding. Our method uses the three models as sub-modules to identify object-centric regions and retrieve knowledge for knowledge-based VQA task.

3 Proposed Method

Knowledge-based VQA task [18] seeks to answer questions based on external knowledge beyond images. Specifically, let us denote a knowledge-based VQA dataset as $\mathcal{D} = \{(I_i, Q_i, A_i)\}_{i=1}^{N}$, where I_i, Q_i and A_i denote the input image, question and answer of the i-th sample respectively, and N is the number of total samples. Given the dataset, the goal is to train a model with parameter θ to generate the answer A_i with input I_i and Q_i.

In this section, we introduce our method *REVIVE*. Figure 2 shows an overview of the method. We leverage the detected regions of the input image to obtain the object-centric region features and retrieve explicit knowledge, meanwhile, we prompt GPT-3 [2] by regional tags, question and context to retrieve implicit knowledge. After that, the regional visual features, retrieved knowledge, and the text prompt consists of regional tags, question and context will then be fused into a encoder-decoder module to generate the answer. We explain more details in Section 3.1, 3.2 and 3.3.

3.1 Regional Feature Extraction Module

Given an image I, we first adopt a object detector to give us the positions of region proposals,

$$\mathcal{B} = \{b_j\}_{j=1}^{M} = D(I), \tag{1}$$

where $\mathcal{B} = \{b_j\}_{j=1}^{M}$ is the set of bouding boxes, M is the number of detected boxes, and $D(\cdot)$ is the object detector.

Here, we adopt $D(\cdot)$ as the visual grounding model GLIP [12]. We use the text prompt "`Detect : person, bicycle, car, ..., toothbrush`", which contains all object categories of MSCOCO dataset [14]. In this way, the model can provide us with all bounding boxes associated with those categories.

After we get the bounding boxes $\mathcal{B}$ of interested objects from GLIP, We crop the image I according to $\mathcal{B}$ to obtain region proposals $\mathcal{R} = \{r_j\}_{j=1}^{M}$. We then extract the object-centric visual features from the proposals: $v_j = E(r_j)$, where $v_j \in \mathbb{R}^S$ is the visual embedding of the j-th proposal, S is the embedding dimension and $E(\cdot)$ represents the image encoder. Inspired by the strong transferring capability of recent contrastively trained vision-language models, we adopt the visual encoder of CLIP [21] as our image encoder $E(\cdot)$. We use the encoding of `[CLS]` token as the final embedding.

To understand the relationship between/among the objects, we find it also important to introduce the position information $\mathcal{B}$ along with its regional visual features.

In addition to the embeddings, explicitly obtaining the description of each region proposal in the textual format is also helpful for knowledge retrieval. For the contrastively trained vision-language models, the training loss explicitly encourages inner product between the image embedding and the text embedding to be larger if the image and the text are well-aligned. Therefore, such a model is capable of selecting the tags that describe the image from a set of customized tags $\bar{\mathcal{T}}$ by computing the inner product. Denote the language encoder of CLIP as $T(\cdot)$. Given a set of tags $\bar{\mathcal{T}} = \{t_i\}_{i=1}^{N_1}$, N_1 is the number of total tags, we compute the inner product between the region proposals and all tags, and adopt the tags with the top-P similarities as the description of the region proposals,

$$\mathcal{H} = \{h_p\}_{p=1}^{P} = \arg\underset{t_i \in \bar{\mathcal{T}}}{\mathrm{TopP}} \langle E(r_j), T(t_i)\rangle, \quad j = 1, \cdots, M, \tag{2}$$

where $\langle \cdot, \cdot \rangle$ is the inner product, P denotes the number of the obtained regional tags and $\mathcal{H}$ means the retrieved regional tags.

In complement to the localized textual description $\mathcal{H}$, we adopt a caption model to explicitly describe the relationships between the major objects and provide more context,

$$c = C(I), \tag{3}$$

where $C(\cdot)$ is the caption model. For example, in Figure 2, the context "`Two brown dogs fighting over a red frisbee`" provides us with the essential relationships between the objects, e.g., fighting *over* a red frisbee. Here, we adopt Vinvl [34] as the caption model $C(\cdot)$.

In summary, we extract regional visual and positional information as $\{v_j\}_{j=1}^{M}$ and $\{b_j\}_{j=1}^{M}$, and textual descriptions for the objects and the relationship between the objects as $\mathcal{H}$ and c. In the next section, we will elaborate on how we use these regional information sources to retrieve external knowledge.

3.2 Object-Centric Knowledge Retrieval Module

Inspired by KAT [5], we consider both the explicit knowledge and implicit knowledge.

3.2.1 Explicit Knowledge

Since the questions from knowledge-based VQA [18] are general and open-ended, introducing external knowledge is important for model to generate accurate answers by providing extra and complementary knowledge beyond visual contents of input images.

External Knowledge Base. We construct a external knowledge base $\mathcal{Q}$ by constructing a subset from Wikidata [26] following KAT [5]. Specifically, we extract 8 commonly appeared categories, ***i.e.***, `Role`, `Point of interest`, `Tool`, `Vehicle`, `Animal`, `Clothing`, `Company`, `Sport`, to form the subset $\mathcal{Q}$. Each item in $\mathcal{Q}$ consists of an entity and a corresponding description, ***e.g.***, one entity and its description can be "`pegboard`" and "`board wall covering with regularly-spaced holes for insertion of pegs or hooks`" respectively.

Regional Knowledge Retrieval. As mentioned earlier, vision-language models like CLIP are capable of selecting the most relevant text from a set of texts. We reformat the entries in knowledge base $\mathcal{Q}$ as "{entity} `is a` {description}", and denote the reformatted text set as $\mathcal{T}$. We retrieve the top-K most relevant knowledge entries among all the regional proposals as explicit knowledge $\mathcal{E}$,

$$\mathcal{E} = \{e_k\}_{k=1}^{K} = \arg\underset{d_i \in \mathcal{T}}{\mathrm{TopK}} \langle E(r_j), T(d_i)\rangle, \quad j = 1, \cdots, M, \tag{4}$$

where K denotes the number of retrieved explicit knowledge samples. In our implementation, we use FAISS [9] to speed up the computation of Equation (2) and (4).

3.2.2 Implicit Knowledge

Large language models, *e.g.*, GPT-3 [2], not only excel in many language tasks, but also memorize lots of commonsense knowledge from its training corpus [32]. Therefore, we exploit GPT-3 [2] as our implicit knowledge base by reformulating the task as open-domain question answering.

Context-Aware Prompt with Regional Descriptions. We design the textual prompt based on question Q, caption c, and regional tags $\mathcal{H}$. Specifically, we adopt the prompt X to be “context: {caption} + {tags}. question: {question}”. In this way, the language model is also supplemented with regional visual information.

Implicit Knowledge Retrieval. Finally, we query GPT-3 model [2] which takes the reformulated prompt X as input, and obtain predictive answer. Since some of the questions may have ambiguity, we follow the prompt tuning procedure of PICa [32] and get answer candidates $\{o_u\}_{u=1}^{U}$. In addition to answer prediction, we also aim for acquiring corresponding explanation e_u from GPT-3 model to obtain more context information. To be more specific, the corresponding explanation is acquired by feeding the text prompt “{question} {answer candidate}. This is because" into GPT-3. Note that “{question}" and “{answer candidate} are input question Q and GPT-3's answer o_u for image I respectively. The final retrieved implicit knowledge can be denoted as $\mathcal{I} = \{(o_u, e_u)\}_{u=1}^{U}$.

3.3 Encoder-Decoder Module

Once we've retrieved the explicit and implicit knowledge and the regional information, we utilize the FiD network structure [7] to encode and decode retrieved knowledge and regional information.

Knowledge Encoder. For the explicit knowledge, we reformat the input text as “entity: {entity} description: {description}”, where the entity and the description is from the entries in the retrieved explicit knowledge $\mathcal{E}$. We denote this text as h_k, where $k = 1, \cdots, K$.

For implicit knowledge, we adopt input format as “candidate: {answer} evidence: {explanation}”, where answer is the retrieved answer o_u and explanation is e_u. Here, $u = 1, \cdots, U$, where U is the number of answers provided by GPT-3. We denote the input text as s_u.

We then encode the knowledge in textual format by the FiD's encoder [25], which is denoted as F_{e},

$$\alpha_k = F_{\text{e}}(h_k), \quad \beta_u = F_{\text{e}}(s_u), \tag{5}$$

in which $\alpha_k \in \mathbb{R}^D$, $\beta_u \in \mathbb{R}^D$ and D means the embedding dimension.

Visual Encoder. We introduce a visual encoder for the visual embeddings $\{v_j\}_{j=1}^{M}$ and positional coordinates $\{b_j\}_{j=1}^{M}$. We feed v_j and b_j into two different fully connected layers, stack the outputs into a sequence of vectors, and then feed them into a transformer encoder F_{v},

$$f = F_{\text{v}}(\text{Concat}(\text{FC}_1(v_1), \text{FC}_2(b_1), \cdots, \text{FC}_1(v_M), \text{FC}_2(b_M))), \tag{6}$$

where $f \in \mathbb{R}^{(2M)\times D}$, $\text{FC}_1(\cdot)$ and $\text{FC}_2(\cdot)$ are two different fully-connected layers, $\text{Concat}(\cdot)$ is the concatenation operation along a new dimension.

Context-aware Question Encoder. To better leverage the context information, we replace the input question Q by the context-aware prompt X, we then encode it by the same transformer encoder $F_{\text{e}}(\cdot)$,

$$q = F_{\text{e}}(X), \tag{7}$$

where $q \in \mathbb{R}^D$ and q means encoded context-aware question.

Generative Decoder. We have obtained the knowledge encoding $\{\alpha_k\}_{k=1}^{K}$ and $\{\beta_u\}_{u=1}^{U}$, visual encoding f, and context-aware question encoding q. Note that as the outputs of the encoder F_{e}, they are all sequences of vectors. We then concatenate these vectors along the first dimension, and feed them into the FiD's decoder F_{d},

$$y = F_{\text{d}}(\text{Concat}(\alpha_1, \cdots, \alpha_K, \beta_1, \cdots, \beta_U, f, q)), \tag{8}$$

where y means the generated answer. The cross entropy loss function is adopted to train the model,

$$\mathcal{L} = -\sum\nolimits_{\ell=1}^{L} \log p_\theta(\bar{y}_\ell | y_{<\ell}), \tag{9}$$

in which L is the length of the ground truth answer text, $\bar{y}_\ell$ is ground truth text at the position ℓ and θ is the model parameters.

Model Ensemble. To generate more accurate answers, one promising method is to leverage multiple trained models, *i.e.*, model ensemble. In our experiments, we just train three models whose initialized seeds are different, and then the most frequent result among the generated results from these three models is selected as final answer prediction for each sample.

3.4 Relationship to Existing Works

Motivated by KAT [5], the proposed *REVIVE* also retrieves two types of knowledge, *i.e.*, implicit and explicit knowledge, for solving knowledge-based VQA. However, we pay more attention to revisiting the visual representation for retrieving knowledge. In addition, we integrate the visual representation of object regions with retrieved knowledge in the answer generative model. The pipeline differences between KAT [5] and our method can be explained in Figure 1.

There're two works [31, 17] that leverage visual regions for knowledge-based VQA as well. However, MAVEx [31] considers object regions as a kind of knowledge without using their visual representation to retrieve other knowledge, KRISP [17] utilizes object regions to learn implicit knowledge by a transformer-based model and retrieve external knowledge by the text symbols of these regions, while our proposed *REVIVE* explores how to better leverage visual representation to retrieve knowledge and integrate their visual features with retrieved knowledge into the answering model.

4 Experiments

4.1 Experimental Setup

Dataset. OK-VQA dataset [18] is selected for evaluation, which is currently the largest knowledge-based VQA dataset. OK-VQA dataset includes 14055 questions associated with 14031 images from MSCOCO dataset [14]. Its questions cover a variety of knowledge categories, and are annotated by Amazon Mechanical Turkers. The training and testing split consist of 9009 and 5046 samples respectively. Each data sample is made up of one question, one corresponding image and 10 ground-truth answers. To construct the general domain tag set $\bar{\mathcal{T}}$, we collect the most frequently searched 400K queries in Bing Search as the tags.

Pre-processing. We utilize the pre-trained visual grounding model GLIP [12] to detect object-centric region proposals by using its default prompt "`Detect: person, bicycle, car, ..., toothbrush`", which contains all object categories of MS-COCO dataset [14]. The captions of images are obtained by the pre-trained Vinvl-Large model [34]. For explicit knowledge and regional tag retrieval, we choose CLIP model (ViT-B/16 variant) [21]. In our experiments, we adopt U, K, M and P as 5, 40, 36 and 30 respectively. Note that the models of CLIP, GLIP, Vinvl and GPT-3 are all frozen during usage.

Implementation Details. We use 4 × NVIDIA V100 32Gb to train models for 10K steps, with a batch size of 8. The learning rate is $8\mathrm{e}^{-5}$ and AdamW [15] is chosen as optimizer. The warm-up steps are 1K and the trained models are evaluated every 500 steps. We initialize our model with the pre-trained T5 model [22], *i.e.*, T5-large, following KAT [5]. The encoder F_v in Equation (6) consists of 9 transformer layers [25]. Note that we evaluate the prediction results after normalization, and the normalization process mainly includes removing articles, punctuation and duplicated whitespace and lowercasing [3, 10].

Evaluation Metric. In our experiments, we choose the soft accuracy of VQAv2 [1] as evaluation metric for comparison.

4.2 Comparison with State-of-the-art Methods

As shown in Table 1, we can see that previous works (*e.g.*, KRISP [17], Visual Retriever-Reader [16] and MAVEx [31]) achieve similar performances, about 38.4% to 39.4% accuracy. Until recently, PICa [32] is the first one that exploits the pre-trained language model GPT-3 [2] as knowledge base for knowledge-based VQA task and KAT [5] further introduces Wikidata [26] as an external knowledge resource, these two works obtain significant performances compared with previous ones.

Table 1: Results comparison with existing methods on OK-VQA dataset [18], the evaluation metric (*i.e.*, accuracy) is in %.

Method	Knowledge Resources	Accuracy (%)
Q only [18]	-	14.9
MLP [18]	-	20.7
BAN [18]	-	25.1
BAN+AN [18]	Wikipedia	25.6
MUTAN [18]	-	26.4
BAN+KG-AUG [11]	Wikipedia+ConceptNet	26.7
MUTAN+AN [18]	Wikipedia	27.8
ConceptBERT [4]	ConceptNet	33.7
KRISP [17]	Wikipedia + ConceptNet	38.4
Visual Retriever-Reader [16]	Google Search	39.2
MAVEx [31]	Wikipedia+ConceptNet+Google Images	39.4
PICa-Base [32]	Frozen GPT-3 (175B)	43.3
PICa-Full [32]	Frozen GPT-3 (175B)	48.0
KAT (Single) [5]	Wikidata+Frozen GPT-3 (175B)	53.1
KAT (Ensemble) [5]	Wikidata+Frozen GPT-3 (175B)	54.4
REVIVE (Single)	Wikidata+Frozen GPT-3 (175B)	**56.6**
REVIVE (Ensemble)	Wikidata+Frozen GPT-3 (175B)	**58.0**

The proposed *REVIVE* can outperform all existing methods by large margins. Specifically, even using the same knowledge resources (*i.e.*, Wikidata [26] and GPT-3 [2]), our single model can achieve **56.6%** accuracy versus previous state-of-the-art method KAT's **53.1%** accuracy, when using model ensemble, our method can achieve **58.0%** accuracy compared with KAT's **54.4%** accuracy. These results demonstrate the effectiveness of the proposed approach.

4.3 Ablation Study

Next, we conduct extensive ablation studies on the single model to figure out the influence of each component of *REVIVE*.

Effect of Region Proposal Number. We perform the ablation study to figure out the effect of using different region proposal numbers. The results are displayed in Table 2. It can be observed that when the region proposal number is 36, the model achieves optimal performance. We conjecture that when the number of region proposals is too large, there are some meaningless and noisy region proposals, while if the number of region proposals is too small, many essential object-centric regions are ignored, which both hurt the model's performance.

Different Knowledge Retrieval Methods. The way of utilizing visual representation for retrieving knowledge plays an important role in knowledge-based VQA. We show the results of using three kinds of knowledge retrieval methods, *i.e.*, image-based, sliding window-based and region-based, in Table 5. Note that *sliding window-based* approach follows KAT [5]. Specifically, we first resizes input images to 384 × 384 and then crop the images with a sliding window whose size is 256 × 256 and stride size is 128. We can observe that the proposed *region-based* approach achieves best performance and surpasses *sliding window-based* method by **1.8%** points, which can validate the effectiveness of exploiting region-based visual representation for retrieving knowledge.

Effect of Regional Tag Number. In order to introduce more semantics into contexts, we propose to add region-aware descriptions (*i.e.*, regional tags) behind given contexts. We report the results of using different regional tag number for text prompt X in Table 3. The results show that when the number of regional tags is 30, it achieves optimal performances. In fact, when the number of regional tags is too large, we'll retrieve relatively irrelevant object tags, sacrificing the model's performance.

Effect of Positional Coordinates. In addition to incorporating visual representation of object-centric region proposals into the model, we also adopt the position information (*i.e.*, positional coordinates). The results of whether using positional coordinates are reported in Table 4. Introducing regional coordinates can improve the performance by **0.8%** points.

Table 2: Experimental results of using different region proposal number.

# of region proposals	Accuracy (%)
5	54.7
18	55.8
36	**56.6**
50	56.2

Table 3: Experimental results of using different regional tag number.

# of regional tags	Accuracy (%)
8	56.2
24	56.4
30	**56.6**
50	56.3

Table 4: Results of whether utilizing bounding box coordinates.

Positional coordinates	Accuracy (%)
✗	55.8
✓	**56.6**

Table 5: Experimental results of leveraging different methods for retrieving knowledge.

Method	Accuracy (%)
Image-based	53.2
Sliding window-based	54.8
Region-based	**56.6**

Effect of Each Component. In order to better show the effect of each component of *REVIVE*, we report the experimental results in Table 6. Note that we choose regional-based manner to retrieve different categories of knowledge. We can observe that introduced components can consistently improve the model's performance. Especially, the explicit knowledge and the object-centric region features can bring **1.6%** and **1.4%** points improvement respectively, feeding context-aware questions, which can be denoted as prompt "`context:` {caption}. `question:` {question}", into the answer generative model can have **0.5%** points gain, further introducing regional descriptions (*i.e.*., regional tags) into contexts, *i.e.*, prompt X, can have **0.7%** points improvement, and model ensemble can bring **1.4%** performance gain. These results can validate the efficiencies of proposed components.

Table 6: Experimental results of different components of *REVIVE*. Note that "*Implicit*", "*Explicit*" and "*Visual*" represent implicit knowledge , explicit knowledge and object-centric region features respectively. "*Context*" represents introducing the contexts into input questions. "*Tag*" represents introducing regional tags into contexts.

Implicit	Explicit	Visual	Context	Tag	Ensemble	Accuracy (%)
✓						52.4
✓	✓					54.0
✓	✓	✓				55.4
✓	✓	✓	✓			55.9
✓	✓	✓	✓	✓		56.6
✓	✓	✓	✓	✓	✓	**58.0**

4.4 Quantitative Result Analysis

Finally, we present the quantitative results and provide analysis for error cases, so that we can have a clear insight into the proposed approach.

Visualizing Results. The success cases of our approach are shown in Figure 3. We can observe that our approach can accurately retrieve implicit and explicit knowledge, which corresponds to the detected object regions, and deal with the relationship among these object areas. For example, in the left example of Figure 3, our method can handle the spatial relationship and recognize the referred object (*i.e.*, the right mouse), thus generating the correct answer, in the right example, our method can retrieve important knowledge (*e.g.*, brazilian terrier) to answer the breed of the referred dog.

Failure Cases Analysis. We showcase the failure examples in Figure 4. As shown in the left example, even though the prediction result *Cabin* doesn't appear in the ground truth answers, the generated answer of our approach is still reasonable for such scenario. For the right example, our predicted result is wrong due to the difficulty of answering such a general question. From Figure 4, we can also observe that our method can generate useful object-centric regions and accurately retrieve corresponding knowledge, especially explicit knowledge, which can demonstrate the potential of the proposed method.

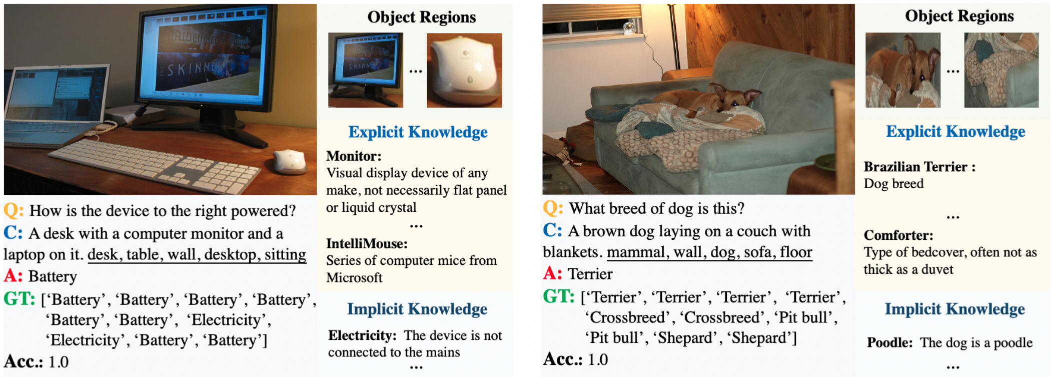
Object Regions
Explicit Knowledge
Monitor:
Visual display device of any make, not necessarily flat panel or liquid crystal
...
IntelliMouse:
Series of computer mice from Microsoft
Implicit Knowledge
Electricity: The device is not connected to the mains
...
Q: How is the device to the right powered?
C: A desk with a computer monitor and a laptop on it. desk, table, wall, desktop, sitting
A: Battery
GT: ['Battery', 'Battery', 'Battery', 'Battery', 'Battery', 'Battery', 'Electricity', 'Electricity', 'Battery', 'Battery']
Acc.: 1.0
Object Regions
Explicit Knowledge
Brazilian Terrier :
Dog breed
...
Comforter:
Type of bedcover, often not as thick as a duvet
Implicit Knowledge
Poodle: The dog is a poodle
...
Q: What breed of dog is this?
C: A brown dog laying on a couch with blankets. mammal, wall, dog, sofa, floor
A: Terrier
GT: ['Terrier', 'Terrier', 'Terrier', 'Terrier', 'Crossbreed', 'Crossbreed', 'Pit bull', 'Pit bull', 'Shepard', 'Shepard']
Acc.: 1.0

Figure 3: Representative success cases of the proposed *REVIVE* on OK-VQA dataset [18]. *"Q"*, *"C"*, *"A"* and *"GT"* denote question, context, predictive answer, ground truth answers respectively. Note that the underlined text represents regional tags and five tags are selected for illustration. We rescale all the object regions to the same size for a clearer view. *"Acc."* means accuracy.

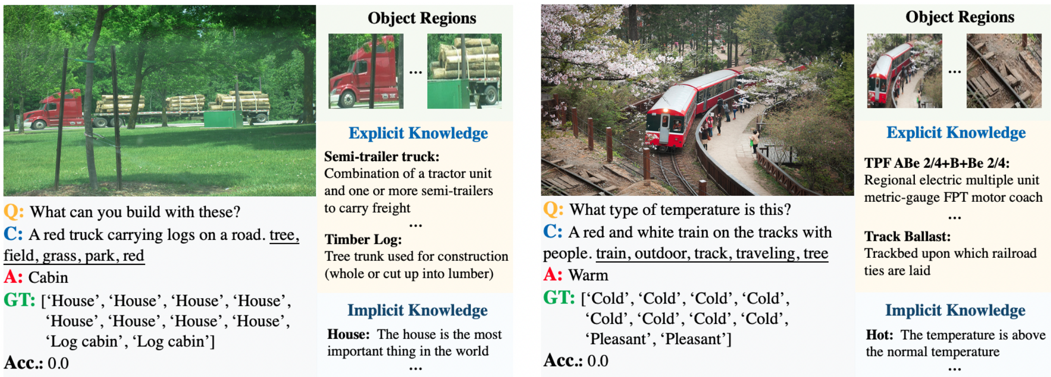
Object Regions
Explicit Knowledge
Semi-trailer truck:
Combination of a tractor unit and one or more semi-trailers to carry freight
...
Timber Log:
Tree trunk used for construction (whole or cut up into lumber)
Implicit Knowledge
House: The house is the most important thing in the world
...
Q: What can you build with these?
C: A red truck carrying logs on a road. tree, field, grass, park, red
A: Cabin
GT: ['House', 'House', 'House', 'House', 'House', 'House', 'House', 'House', 'Log cabin', 'Log cabin']
Acc.: 0.0
Object Regions
Explicit Knowledge
TPF ABe 2/4+B+Be 2/4:
Regional electric multiple unit metric-gauge FPT motor coach
...
Track Ballast:
Trackbed upon which railroad ties are laid
Implicit Knowledge
Hot: The temperature is above the normal temperature
...
Q: What type of temperature is this?
C: A red and white train on the tracks with people. train, outdoor, track, traveling, tree
A: Warm
GT: ['Cold', 'Cold', 'Cold', 'Cold', 'Cold', 'Cold', 'Cold', 'Cold', 'Pleasant', 'Pleasant']
Acc.: 0.0

Figure 4: Representative failure cases of the proposed *REVIVE* on OK-VQA dataset [18].

5 Conclusion

In this paper, we propose a method called *REVIVE*, which revisits regional visual representation for knowledge-based VQA task. Specifically, we conduct comprehensive experiments to show the effects of different components in the proposed approach, which can demonstrate the efficiency of the proposed region-based knowledge retrieval method. In addition, *REVIVE* incorporates object-centric regional visual features and two kinds of knowledge, *i.e.*, implicit and explicit knowledge, into the answer generative model for prediction. Our method achieves the state-of-the-art performance on OK-VQA dataset.

6 Limitations and Broader Impact

The quality of constructed Wikidata subset and designed textual prompt can influence final retrieved knowledge. In addition, the detector for obtaining region proposals also affect retrieved knowledge and visual features, all these factors affect the models' performances.

This paper proposes a novel approach *REVIVE* for knowledge-based VQA. *REVIVE* can help models to efficiently use visual and language information sources to answer open-domain questions. It can also generalize to real-life products, *e.g.*, dialogue robot. However, the failure cases of *REVIVE* will be negative to the society when using it as the educational technique. There may also exist certain forms of bias, *i.e.*, the model may predict biased answers if the training data is overwhelmed with certain scenarios.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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

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

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

[6] Gautier Izacard and Edouard Grave. Distilling knowledge from reader to retriever for question answering. *arXiv preprint arXiv:2012.04584*, 2020.

[7] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*, 2020.

[8] Haojun Jiang, Yuanze Lin, Dongchen Han, Shiji Song, and Gao Huang. Pseudo-q: Generating pseudo language queries for visual grounding. *arXiv preprint arXiv:2203.08481*, 2022.

[9] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[10] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*, 2019.

[11] Guohao Li, Xin Wang, and Wenwu Zhu. Boosting visual question answering with context-aware knowledge aggregation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1227–1235, 2020.

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

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

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[16] Man Luo, Yankai Zeng, Pratyay Banerjee, and Chitta Baral. Weakly-supervised visual-retriever-reader for knowledge-based question answering. *arXiv preprint arXiv:2109.04014*, 2021.

[17] Kenneth Marino, Xinlei Chen, Devi Parikh, Abhinav Gupta, and Marcus Rohrbach. 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, 2021.

[18] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 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, 2019.

[19] Medhini Narasimhan, Svetlana Lazebnik, and Alexander Schwing. Out of the box: Reasoning with graph convolution nets for factual visual question answering. *Advances in neural information processing systems*, 31, 2018.

[20] Medhini Narasimhan and Alexander G Schwing. Straight to the facts: Learning knowledge base retrieval for factual visual question answering. In *Proceedings of the European conference on computer vision (ECCV)*, pages 451–468, 2018.

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

[22] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[23] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[24] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

[27] Jianfeng Wang, Xiaowei Hu, Zhe Gan, Zhengyuan Yang, Xiyang Dai, Zicheng Liu, Yumao Lu, and Lijuan Wang. Ufo: A unified transformer for vision-language representation learning. *arXiv preprint arXiv:2111.10023*, 2021.

[28] Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton Van Den Hengel. Fvqa: Fact-based visual question answering. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2413–2427, 2017.

[29] Peng Wang, Qi Wu, Chunhua Shen, Anton van den Hengel, and Anthony Dick. Explicit knowledge-based reasoning for visual question answering. *arXiv preprint arXiv:1511.02570*, 2015.

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

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

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

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

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

[35] Zihao Zhu, Jing Yu, Yujing Wang, Yajing Sun, Yue Hu, and Qi Wu. Mucko: multi-layer cross-modal knowledge reasoning for fact-based visual question answering. *arXiv preprint arXiv:2006.09073*, 2020.

A Appendix

In the Appendix, we provide the following sections:

(a) Implementation details of implicit knowledge retrieval in Section A.1.

(b) Ablation study experiments in Section A.2.

(c) Visualization results in Section A.3.

A.1 Implementation Details of Implicit Knowledge Retrieval

We first describe more implementation details of implicit knowledge retrieval of the proposed *REVIVE*. Specifically, we explain how we extract multiple answer candidates.

Multiple Candidates. We retrieve multiple implicit knowledge candidates for each sample during training and inference stages to improve the robustness of answer generation. Specifically, we follow PICa [32], which proposes to use multi-query ensemble, *i.e.*, they prompt the GPT-3 [2] for k_1 times and choose the one with the highest probability as final answer prediction. Compared with PICa's multi-query ensemble approach, we take all these k_1 predictions from GPT-3 [2] as implicit knowledge candidates. Note that for each candidate, we also prompt the GPT-3 model to obtain its corresponding explanation. In our experiments, we just retrieve 5 (*i.e.*, $k_1 = 5$) implicit knowledge candidates and corresponding explanations.

A.2 Ablation Study

Next, we conduct more ablation study experiments to provide deeper insight into the components of our proposed *REVIVE*.

The effect of multiple implicit knowledge candidates. To validate the influence of the number of retrieved implicit knowledge candidate on the model's performance, we report the results in Table 7. When using only one implicit knowledge candidate, the model can achieve 55.8 % accuracy, after taking 5 implicit knowledge candidates, the performance can be improved to **56.6** % accuracy. However, when the retrieved candidate number is 8, we can see that the performance isn't the best, we conjecture that it's enough to include essential candidates when $k_1 = 5$. Due to certain incorrect answer predictions by GPT-3, larger k_1 may introduce incorrect and unnecessary candidates, thus hurting the model's performance by using too much noisy and misleading knowledge.

The effect of explicit knowledge number. Since the number of retrieved explicit knowledge samples can have an effect on the model's performance, we conduct the experiments and show the results in Table 8. We find the model can achieve optimal performance when $k_2 = 40$. It's reasonable to see that a too large k_2 (*i.e.*, $k_2 = 50$) cannot let the model achieve optimal performance, since when k_2 increases, there will exist certain retrieved explicit knowledge samples which have relatively low confidences, thus introducing unreliable knowledge and hurting the model's performance.

Table 7: Experimental results of using different implicit knowledge candidates. k_1 represents the number of retrieved implicit knowledge candidates.

k_1	Accuracy (%)
1	55.8
3	56.3
5	**56.6**
8	56.4

Table 8: Experimental results of different explicit knowledge numbers. k_2 represents the number of retrieved explicit knowledge samples.

k_2	Accuracy (%)
10	55.6
20	55.9
30	56.2
40	**56.6**
50	56.3

A.3 Visualization Results

Finally, we show more visualization cases in Figure 5 and 6. We can see that our proposed method can focus on important object-centric areas, and then retrieve relevant knowledge for corresponding regional areas, which can be used to generate accurate answers. These visualization results can demonstrate the effectiveness and potential of the proposed *REVIVE*.

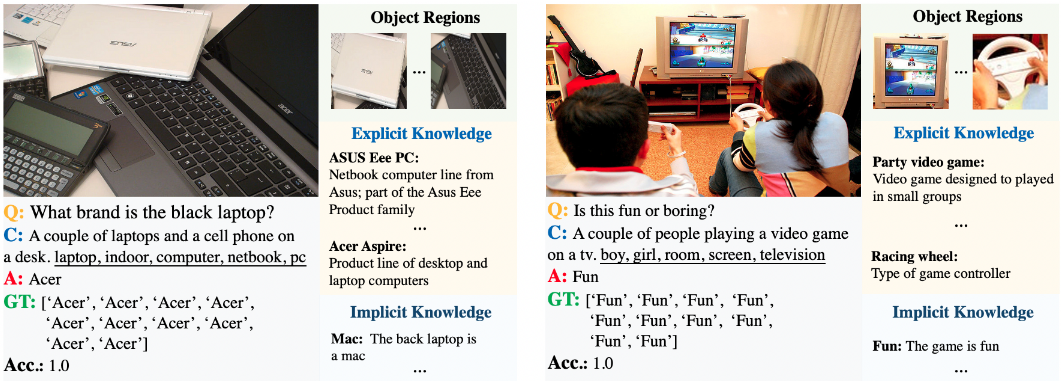
Object Regions
...
Explicit Knowledge
ASUS Eee PC:
Netbook computer line from Asus; part of the Asus Eee Product family
...
Acer Aspire:
Product line of desktop and laptop computers
Implicit Knowledge
Mac: The back laptop is a mac
...
Q: What brand is the black laptop?
C: A couple of laptops and a cell phone on a desk. laptop, indoor, computer, netbook, pc
A: Acer
GT: ['Acer', 'Acer', 'Acer', 'Acer', 'Acer', 'Acer', 'Acer', 'Acer', 'Acer', 'Acer']
Acc.: 1.0
Object Regions
...
Explicit Knowledge
Party video game:
Video game designed to played in small groups
...
Racing wheel:
Type of game controller
Implicit Knowledge
Fun: The game is fun
...
Q: Is this fun or boring?
C: A couple of people playing a video game on a tv. boy, girl, room, screen, television
A: Fun
GT: ['Fun', 'Fun', 'Fun', 'Fun', 'Fun', 'Fun', 'Fun', 'Fun', 'Fun', 'Fun']
Acc.: 1.0

Figure 5: Representative visualization cases of the proposed *REVIVE* on OK-VQA dataset [18]. *"Q"*, *"C"*, *"A"* and *"GT"* denote question, context, predictive answer, ground truth answers respectively. Note that the underlined text represents regional tags and five tags are selected for illustration. We rescale all the object regions to the same size for a clearer view. *"Acc."* means accuracy.

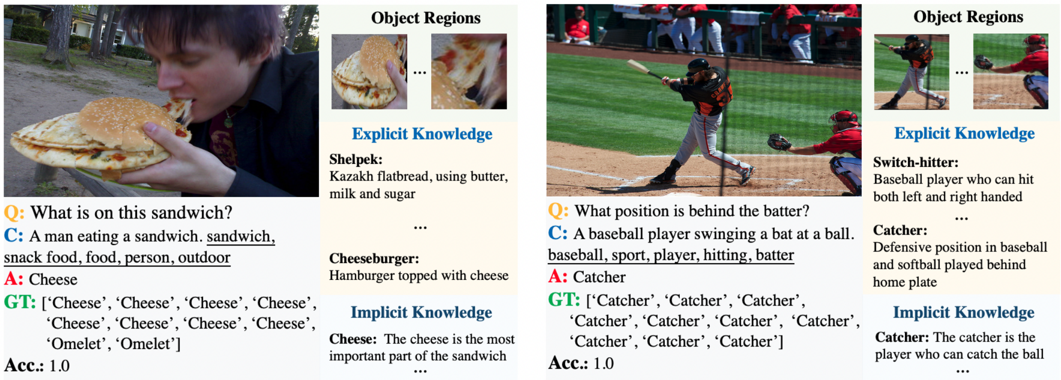
Object Regions
...
Explicit Knowledge
Shelpek:
Kazakh flatbread, using butter, milk and sugar
...
Cheeseburger:
Hamburger topped with cheese
Implicit Knowledge
Cheese: The cheese is the most important part of the sandwich
...
Q: What is on this sandwich?
C: A man eating a sandwich. sandwich, snack food, food, person, outdoor
A: Cheese
GT: ['Cheese', 'Cheese', 'Cheese', 'Cheese', 'Cheese', 'Cheese', 'Cheese', 'Cheese', 'Omelet', 'Omelet']
Acc.: 1.0
Object Regions
...
Explicit Knowledge
Switch-hitter:
Baseball player who can hit both left and right handed
...
Catcher:
Defensive position in baseball and softball played behind home plate
Implicit Knowledge
Catcher: The catcher is the player who can catch the ball
...
Q: What position is behind the batter?
C: A baseball player swinging a bat at a ball. baseball, sport, player, hitting, batter
A: Catcher
GT: ['Catcher', 'Catcher', 'Catcher', 'Catcher', 'Catcher', 'Catcher', 'Catcher', 'Catcher', 'Catcher', 'Catcher']
Acc.: 1.0

Figure 6: Representative visualization cases of the proposed *REVIVE* on OK-VQA dataset [18]. *"Q"*, *"C"*, *"A"* and *"GT"* denote question, context, predictive answer, ground truth answers respectively. Note that the underlined text represents regional tags and five tags are selected for illustration. We rescale all the object regions to the same size for a clearer view. *"Acc."* means accuracy.