International Journal of Computer Vision
https://doi.org/10.1007/s11263-023-01976-7

Learning by Asking Questions for Knowledge-Based Novel Object Recognition

Kohei Uehara[1] · Tatsuya Harada[1,2]

Received: 3 April 2023 / Accepted: 16 December 2023

Abstract

In real-world object recognition, there are numerous object classes to be recognized. Traditional image recognition methods based on supervised learning can only recognize object classes present in the training data, and have limited applicability in the real world. In contrast, humans can recognize novel objects by questioning and acquiring knowledge about them. Inspired by this, we propose a framework for acquiring external knowledge by generating questions that enable the model to instantly recognize novel objects. Our framework comprises three components: the object classifier (OC), which performs knowledge-based object recognition, the question generator (QG), which generates knowledge-aware questions to acquire novel knowledge, and the policy decision (PD) Model, which determines the "policy" of questions to be asked. The PD model utilizes two strategies, namely "confirmation" and "exploration"—the former confirms candidate knowledge while the latter explores completely new knowledge. Our experiments demonstrate that the proposed pipeline effectively acquires knowledge about novel objects compared to several baselines, and realizes novel object recognition utilizing the obtained knowledge. We also performed a real-world evaluation in which humans responded to the generated questions, and the model used the acquired knowledge to retrain the OC, which is a fundamental step toward a real-world human-in-the-loop learning-by-asking framework. We plan to release the dataset immediately upon acceptance of our work.

Keywords Visual question generation · Novel object recognition · Human-in-the-loop learning · Knowledge acquisition

1 Introduction

Object category recognition has long been a central topic in computer vision research. Traditionally, object recognition has been addressed by supervised learning using a large dataset of image-label pairs (Deng et al., 2009). However, with supervised approaches, the model can only recognize a frozen set of object classes, and is not suitable for real-world object recognition, where numerous object classes exist. Recently, image recognition methods based on contrastive learning using image-text pair datasets have emerged (Radford, 2021; Jia et al., 2021). By training with hundreds of millions of image-text pairs, these models have acquired remarkable zero-shot recognition capabilities for various objects. However, these models can recognize objects that commonly appear in the pre-training dataset but are not as effective for rare objects (Shen et al., 2022). Collecting new data and retraining the entire model to make these models recognize novel objects is impractical considering the cost of data collection and computation. Therefore, it is essential to develop a method that enables the model to recognize novel objects while maintaining low data collection costs and avoiding model retraining as much as possible.

Asking questions and explicitly acquiring knowledge are important skills when humans acquire knowledge about the world (Chouinard, 2007; Ronfard et al., 2018). Inspired by this, we explored methods to dynamically increase knowledge in image recognition by asking questions. This approach has several advantages over the traditional supervised learning method: (1) it requires only a small amount of data to acquire knowledge because the system acquires only the required knowledge, and (2) it has a low data collection cost because the system itself seeks the required data.

Communicated by Massimiliano Mancini.

✉ Kohei Uehara
uehara@mi.t.u-tokyo.ac.jp

Tatsuya Harada
harada@mi.t.u-tokyo.ac.jp

[1] The University of Tokyo, Tokyo, Japan

[2] RIKEN, Tokyo, Japan

Published online: 12 January 2024

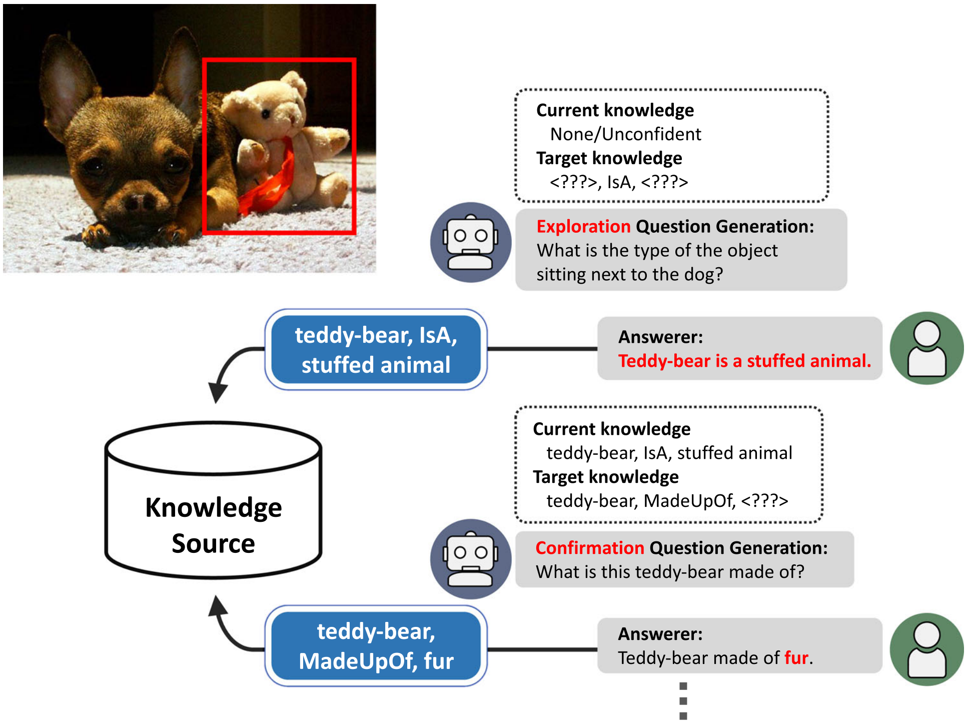

Current knowledge
None/Unconfident
Target knowledge
<???>, IsA, <???>
Exploration Question Generation:
What is the type of the object sitting next to the dog?
teddy-bear, IsA, stuffed animal
Answerer:
Teddy-bear is a stuffed animal.
Knowledge Source
Current knowledge
teddy-bear, IsA, stuffed animal
Target knowledge
teddy-bear, MadeUpOf, <???>
Confirmation Question Generation:
What is this teddy-bear made of?
teddy-bear, MadeUpOf, fur
Answerer:
Teddy-bear made of fur.

Fig. 1 Example of a knowledge acquisition process via visual-question asking. An intelligent agent observes an image and asks a question to an answerer to acquire knowledge. The answerer answers the question, the agent updates its knowledge, and then the agent asks another question to acquire more knowledge

We propose a pipeline comprising a knowledge-based object classifier (OC), a question generator (QG) for knowledge acquisition, and a Policy Decision (PD) model to determine the optimal questioning strategy. Following previous studies on structured knowledge (Ji et al., 2022), we represent knowledge as a *knowledge triplet*, that is, a list of three words or phrases: *head*, *relation*, and *tail*, such as ⟨dog, IsA, mammal⟩.

In anticipation of the ultimate goal of using the acquired knowledge to improve the object recognition performance, the OC is designed to perform object recognition while explicitly modeling the knowledge. This is realized by computing the image-knowledge similarity. The QG model then generates questions to add new knowledge to the knowledge source for novel object recognition. In the QG model, we use two modes in question generation: *confirmation* and *exploration*, as illustrated in Fig. 1.

Let's imagine the situation where our model encounters an image of a teddy bear for the very first time. Since it doesn't know anything about teddy bears, it enters the "Exploration" mode to generate questions. In this mode, the model asks broad questions without focusing on specific details, allowing it to learn entirely new knowledge. An example of such a question is: "What is the type of the object sitting next to the dog?"

When a human provides an answer, the model uses that answer to add the newly acquired knowledge to its knowledge source. For instance, it can learn something like "teddy-bear, IsA, stuffed animal" and store it to the knowledge source.

The second mode, named "Confirmation," is used when the model already has some knowledge about the object in question. Here, the model asks specific questions to confirm what it knows.

For example, if the model has already identified the object as a "teddy-bear" from previous question and answer, it might then inquire about the teddy bear's material. The model sets a target knowledge like ⟨`teddy-bear, MadeUpOf, [MASK]`⟩ and generates a question like "What is this teddy-bear made of?"

In the question-and-answer process, the model is tasked with determining the appropriate strategy to employ-either exploration or confirmation-and deciding the optimal moment to cease questioning. These determinations are made by the policy decision (PD) module. The PD module produces a policy for question generation by taking into account the current state of the object classifier (OC) model and the history of posed questions. The PD module gets trained using a reinforcement learning algorithm to maximize the expected performance of the OC model, allowing for optimal strategy selection in various scenarios.

Our contributions and findings are summarized as follows:

- We propose a novel pipeline to acquire knowledge about novel objects by asking questions. We designed the OC model based on CLIP (Radford, 2021) and the QG model as a Transformer (Vaswani et al., 2017) based text generation model.

- We collect a novel dataset—*Professional K-VQG* dataset, which contains knowledge-aware visual questions, annotated by experts. This dataset complements the existing K-VQG dataset, which was limited in terms of expert annotations. By merging our new dataset with the existing K-VQG dataset, we created an enriched resource—*K-VQG v2* dataset.
- We compare our proposed pipeline with several baselines and show that the knowledge acquired through question generation is effective for novel object recognition.
- We conducted an experiment with human-in-the-loop setting, where humans provide answers to the generated questions, and human-written answers are used to train the OC model. This experiment demonstrates the practicality of the proposed pipeline in real-world applications and validates its effectiveness by integrating human expertise into the learning process.

2 Related Work

2.1 Novel Object Recognition

Novel object recognition, which aims to increase the number of recognizable object classes, is a widely studied problem in the field of object recognition. A typical approach in novel object recognition involves training a model that computes the similarities between the visual and semantic features of objects. To compute the semantic features of a novel object, external knowledge about the object (e.g., attributes (Akata et al., 2016; Farhadi et al., 2009; Jayaraman & Grauman, 2014; Lampert et al., 2009; Li et al., 2021), class hierarchy (Rohrbach et al., 2011; Wang, 2018), or textual description (Ba et al., 2015; Qiao et al., 2016; Reed et al., 2016; Zareian, 2021)) is often employed. Recently proposed vision-and-language contrastive learning methods, such as CLIP (Radford, 2021) and ALIGN (Jia et al., 2021), leverage extremely large-scale image caption data to learn the relationship between images and their textual descriptions. With the help of the prefix-tuning technique, these models have demonstrated strong zero-shot recognition abilities. However, the aforementioned studies share a limitation in that they require either a well-prepared knowledge database on novel objects or a large number of image-text pair datasets and carefully designed prompts, both of which are labor-intensive tasks for humans. Our proposed method addresses this limitation by enabling the model to acquire the necessary knowledge dynamically through question generation, thereby reducing human effort.

2.2 Visual Question Generation (VQG)

Early studies on VQG employed simple methods that involved inputting image features into a text decoder and generating questions (Mostafazadeh et al., 2016). Recent studies have focused on improving the control over the content of the generated questions. Typically, this involves providing a text decoder with additional information along with image features to achieve better control. This was achieved by providing answers (Li et al., 2018; Liu et al., 2018), answer categories (Krishna et al., 2019; Uehara et al., 2018; Uppal et al., 2021), or by targeting knowledge that is expected to be acquired through questioning (Uehara & Harada, 2022). The latter study created a knowledge-aware VQG dataset (*K-VQG*) using Amazon Mechanical Turk (AMT) and employed UNITER (Chen et al., 2020), a state-of-the-art vision-and-language transformer, as an encoder for images and knowledge to successfully generate questions for knowledge acquisition. We designed our question generation model based on their work. In addition, we subsequently curated a new dataset, named Professional K-VQG. We followed the same format as their approach but with one significant difference—our annotations were performed exclusively by experts, not by workers on AMT.

We have summarized the key features of the existing VQG dataset and our dataset in Table 1. Our dataset is the first dataset with common-sense knowledge annotations and target bounding boxes, and annotated by humans.

2.3 Learning by Asking (LBA)

LBA is an approach that generates questions to collect additional data for model training. LBA has been studied in both natural language processing and vision-and-language domains. In the realm of NLP, various studies have harnessed the power of LBA for enhancing tasks like reading comprehension. For instance, Du et al. (2017) explored automatic question generation from text passages, leveraging attention-based sequence learning models. Yuan et al. (2017) employed LBA techniques to improve question-answering systems' performance, while Curiosity-driven Question Generation (Scialom & Staiano, 2020) took a novel approach to generate questions aimed at enriching existing knowledge or elucidating previous information for question answering task.

In the vision-and-language domain, the work of Misra et al. (2018) applied LBA to the VQA task. Unlike traditional VQA methods, where questions are predefined during training, their model had the capability to generate its own questions and realized a more organic and interactive learning process. Further bridging vision and language through LBA, the study by Shen et al. (2019) showcased an agent

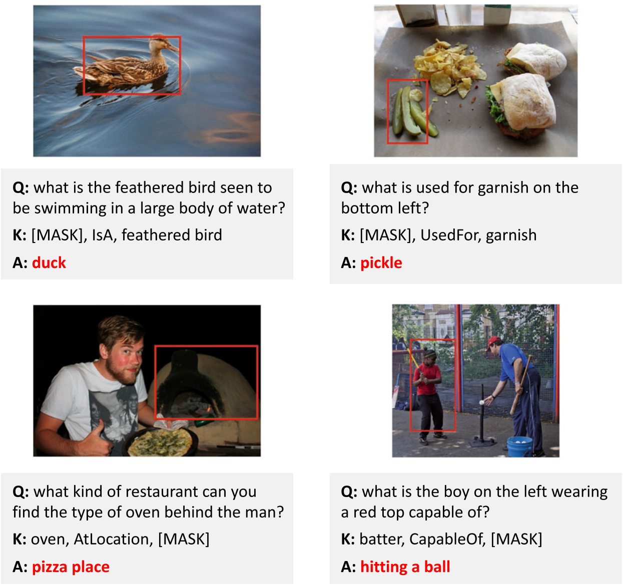
Table 1 This table provides a comprehensive comparison of major datasets used in VQG and Knowledge-aware VQA, emphasizing their features such as the number of questions, knowledge types, and annotation methodologies (Uehara & Harada, 2022)

	Num. of Q	Knowledge type?	Structured knowledge?	Target bounding box?	Manually annotated?
VQAv2 (Goyal & Khot, 2017)	1.1M	N/A	✗	✗	✓
VQG$_{\text{COCO, Flickr, Bing-5000}}$ (Mostafazadeh et al., 2016)	5000	N/A	✗	✗	✓
FVQA (Wang et al., 2017)	5,826	Common-sense	✓	✗	✓
OK-VQA (Marino et al., 2019)	14,055	Open knowledge	✗	✗	✓
K-VQA (Shah et al., 2019)	183,007	Named entities	✓	✗	✗
CRIC (Gao et al., 2019)	1.3M	Common-sense	✓	✓	✗
K-VQG (v2)	22,212	Common-sense	✓	✓	✓

The K-VQG dataset is the only dataset that is common-sense knowledge-aware, annotated by humans, and associated with target bounding boxes. This table is partly derived from

that actively learns by posing specific natural language questions to humans for the task of image captioning.

However, despite these advances, existing research has focused primarily on well-defined tasks, such as reading comprehension, standard VQA, or image captioning. In contrast to these approaches, we address the broad challenge of real-world object recognition by introducing a framework that dynamically recognizes novel objects through questioning.



Q: what is the feathered bird seen to be swimming in a large body of water?
K: [MASK], IsA, feathered bird
A: duck

Q: what is used for garnish on the bottom left?
K: [MASK], UsedFor, garnish
A: pickle

Q: what kind of restaurant can you find the type of oven behind the man?
K: oven, AtLocation, [MASK]
A: pizza place

Q: what is the boy on the left wearing a red top capable of?
K: batter, CapableOf, [MASK]
A: hitting a ball

Fig. 2 Examples of the Professional K-VQG dataset

3 Professional K-VQG Dataset

To address the limitations in existing datasets and further advance the field of knowledge-aware visual question generation, we developed a novel dataset, *Professional K-VQG*. This dataset comprises knowledge-aware visual questions related to objects, annotated by professional annotators. The images are sourced from the Visual Genome (Krishna et al., 2017), whereas knowledge is derived from ConceptNet (Speer et al., 2017) and ATOMIC$^{20}_{20}$ (Hwang et al., 2020). We identified 371 object classes common to both the Visual Genome dataset and the knowledge sources.

ConceptNet In ConceptNet, knowledge is structured as triplets in the format of ⟨head, relation, tail⟩. For instance, the triplet ⟨cat, AtLocation, sofa⟩ represents the concept that a cat can be found on a sofa. ConceptNet comprises approximately 34 million triplets and 37 relation types. However, some relations in ConceptNet, such as *DistinctFrom* or *MotivatedByGoal*, are unsuitable for generating questions about images. Therefore, we identified 15 relation types as appropriate targets for image-based question generation.

ATOMIC$^{20}_{20}$ ATOMIC$^{20}_{20}$ features over 1 million knowledge triplets related to physical-entity relations (e.g., ⟨bread, ObjectUse, make french toast⟩), event-centered relations (e.g., ⟨PersonX eats spinach, isAfter, PersonX makes dinner⟩), and social interactions (e.g., ⟨PersonX calls a friend, xIntent, to socialize with their friend⟩). For our dat aset construction, we exclusively used physical-entity relations since they are more relevant to images in Visual Genome.

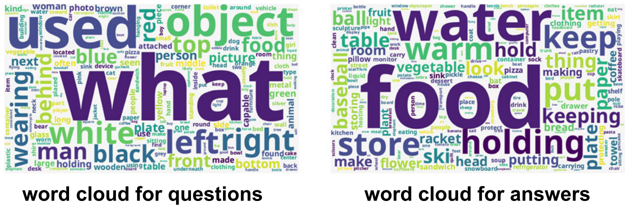


Fig. 3 Word clouds for the questions and answers in the Professional K-VQG dataset

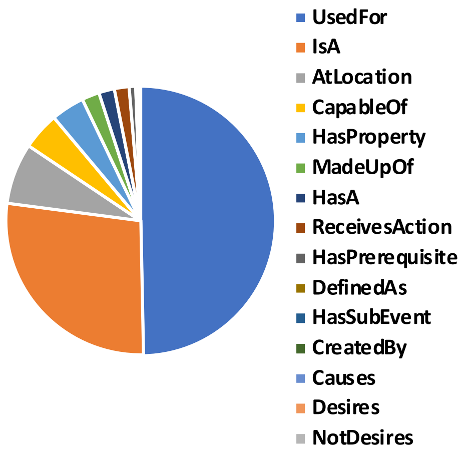


Fig. 4 Distribution of relations in the professional K-VQG dataset

During the annotation process, we first extracted the corresponding knowledge triplets for each candidate object in the image from the knowledge sources. Subsequently, annotators were instructed to create knowledge-aware questions with the head or tail of the knowledge triplet as the answer. Annotators were provided with a bounding box indicating the target object. In addition, they were provided with a list of candidate knowledge items related to the image and target object. After selecting a knowledge item, the annotators wrote questions and answers based on the selected knowledge.

A summary of the guidelines for formulating the questions and answers is as follows:

- The answer should be the head or the tail of the knowledge.
- Answers could be rephrased to ensure a natural flow of words and sentences.
- Questions should be framed in relation to other objects in the image or object's position within the image.
- Questions should not mention the presence of bounding boxes.

This process resulted in 10,431 questions for 9210 images, with 5242 unique knowledge.

Figure 2 displays sample questions from the dataset, demonstrating their relation to the images and target the masked part of the knowledge—e.g., if the target knowledge is ⟨`[MASK], IsA, feathered bird`⟩, the resulting question is actually about the category of the bird (*What is the feathered bird seen to be swimming in a large body of water?*)

The dataset's word clouds and relation distribution are illustrated in Figs. 3 and 4, which reveal a diverse range of questions encompassing topics such as food, clothing, and furniture. Although the most frequent relations, "UsedFor" and "IsA," constitute approximately 50 and 25% of the total, respectively, this apparent bias reflects the prevalence of these relations in the knowledge sources.

3.1 K-VQG v2 Dataset

To enhance the existing K-VQG dataset (Uehara & Harada, 2022) (referred to as K-VQG v1), we integrated it with the Professional K-VQG dataset, resulting in *K-VQG v2* dataset. Anticipating the integration of object recognition models in this study, we excluded samples in which the Faster R-CNN failed to detect the target regions as objects (i.e., the IoU between the detected bounding box and the target bounding box was less than 0.5).

The detailed statistics of the Professional K-VQG dataset and K-VQG v2 dataset are shown in Table 2. The K-VQG v2 dataset features 22,212 questions on 9210 images, which is a significant increase compared to previous versions. One

Table 2 Detailed statistics of the professional K-VQG and K-VQG v2 dataset

	K-VQG v1	Professional K-VQG	K-VQG v2
# Questions	16,098	10, 431	22,212
—Head answers	11,588	5047	13,916
—Tail answers	4510	5384	8296
# Images	13, 648	9494	9210
# Unique answers	2819	3687	4953
# Unique knowledge	6084	5242	7808
# Unique head	527	371	533
# Unique tail	4922	4257	6199

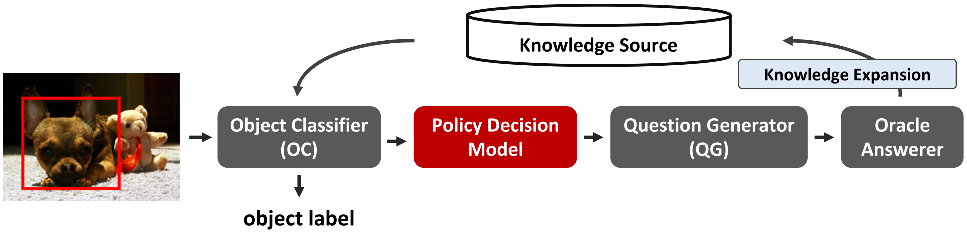


Fig. 5 Overall pipeline of our method. The OC model performs knowledge-based object recognition using knowledge sources. The QG model generates questions targeting the knowledge needed for novel object recognition. Answers to the questions are provided by the Oracle Answerer and added to the knowledge source. With the newly added knowledge, the OC model is able to recognize novel objects

of the significant features of the K-VQG v2 dataset is the increased number of unique answers and knowledge. It has 4953 unique answers and 7808 unique knowledge items, which represent a considerable improvement over the previous versions. This indicates that the K-VQG v2 dataset has an increased diversity of answers and knowledge, which can improve the generalization ability of the VQG model trained.

These statistics reveal that the K-VQG v2 dataset is not only larger but also more diverse and comprehensive than its predecessors, making it a valuable resource for research and development in the field of knowledge-aware VQG.

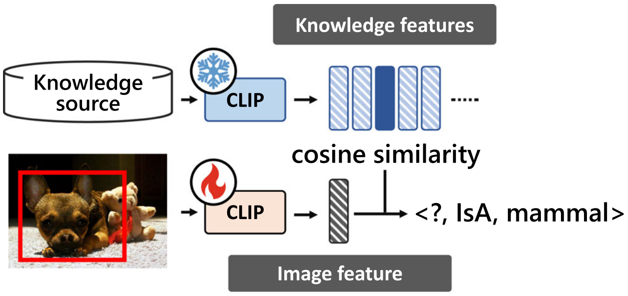


Fig. 6 Architecture of the OC model. Based on the knowledge encoded by BERT and the similarity calculation of the object image encoded by CLIP, the prediction of the knowledge required for object recognition is performed

4 Method

Our system is designed with three modules: the *O*bject *C*lassifier (OC), the *Q*uestion *G*enerator (QG), and the *P*olicy *D*ecision model (PD). In this section, we present an overview of the system pipeline, followed by a detailed description of each module. The entire pipeline is illustrated in Fig. 5.

4.1 Overview

Starting with the OC model, this module takes an object region extracted from an image, and predicts the most-reltaed knowledge to the object, and outputs the object label. More specifically, the OC model, with its knowledge-centric object recognition capatbility, retrieves the corresponding knowledge triplet $k = [h, r, t] \in \mathcal{K}$ from the knowledge source $\mathcal{K}$. Here, h denotes the head (e.g., the object label), r denotes the relation, and t denotes the tail (e.g., the object property or attribute). The OC's output, in essence, is the object label based on the matched knowledge from $\mathcal{K}$. The strength of this classifier is evident when novel knowledge is added; there is no need to retrain the whole model, just update the knowledge source.

Note that in our study, we provide predefined object regions in images for recognition. We chose not to incorporate object detection to maintain a focused and less complex architecture, as our primary aim is to learn novel object concepts. Existing detectors mainly recognize trained objects, making them unsuitable for our research involving unknown objects. While recent advances tackle more generalized object detection (Gu et al., 2021; Kirillov et al., 2023), we've deferred such considerations for future research.

The QG model, taking its cue from the OC's output knowledge and the region-of-interest, generates questions regarding the objects in the image to obtain relevant knowledge that is useful for novel object recognition. Specifically, the QG accepts a partially masked knowledge triplet (e.g., ⟨`[MASK], IsA, mammal`⟩) as input, which is taken from the output of the OC model. This approach encourages the model to generate questions that help acquire the most effective knowledge for object recognition.

The PD model plays an instrumental role in dictating the sequence of questions. Taking as input the current state of the OC model (i.e., the distribution of the knowledge similarity score) and the region image, it outputs the most appropriate next question to ask. In essense, this module decides the policy of question asking. Without this module, the model might gather incorrect knowledge or redundantly acquire information it already possesses. For instance, consider an unknown object that, based on the OC model's prediction, is identified to be "MadeUpOf, fur". In this case, posing the question "What is the object made up of?" becomes redundant. Hence, the Policy Decision Model is crucial for guiding the model to ask more insightful questions that can yield new knowledge.

Upon obtaining answers to the generated questions, the acquired knowledge $\mathcal{K}'$ is added to the model's original knowledge source $\mathcal{K}$. The OC's knowledge source is thereafter updated as $\mathcal{K}^+ = \mathcal{K} \cup \mathcal{K}'$. In the subsequent inference phase, the OC refers to the updated knowledge source $\mathcal{K}^+$ used to make predictions regarding novel objects.

4.2 Object Classifier

The OC model, as illustrated in Fig. 6, is designed to predict the object label while utilizing object-related knowledge by leveraging the similarity between the object feature $\boldsymbol{f}_o \in \mathbb{R}^d$ and knowledge feature $\boldsymbol{f}_k \in \mathbb{R}^d$ of the associated knowledge. Specifically, the similarity is computed as $p(k) = \mathrm{sim}(\boldsymbol{f}_o, \boldsymbol{f}_k)$, where d denotes the dimensions of the object and the knowledge features.

To effectively predict object knowledge, we decided to base our OC model on the state-of-the-art visual recognition model, CLIP (Radford, 2021), which comprises an image encoder and a text encoder that calculate the similarity between images and text. The image encoder of the CLIP, f_θ, accepts a cropped image I_{crop} as the input, and outputs the visual feature $\boldsymbol{f}_o$. The knowledge features $\boldsymbol{f}_k$ are computed using the pre-trained CLIP text encoder f_ϕ. Prior to feeding knowledge into the text encoder, we convert the triplet representation (e.g., ⟨`cat, IsA, mammal`⟩) into a single sentence with a masked head (e.g., ⟨`[MASK] is a mammal`⟩), allowing the model to focus on object-related knowledge instead of the object label itself.

The cosine similarity is employed to measure the similarity between the object and knowledge features as follows:

$$\boldsymbol{f}_o = f_\theta(I_{\mathrm{crop}}), \quad \boldsymbol{f}_k = f_\phi(k) \tag{1}$$

$$\mathrm{sim}(\boldsymbol{f}_o, \boldsymbol{f}_k) = \frac{\boldsymbol{f}_o^\top \boldsymbol{f}_k}{\|\boldsymbol{f}_o\| \|\boldsymbol{f}_k\|} \tag{2}$$

The OC model is trained to minimize the binary cross-entropy loss as follows:

$$\begin{aligned} L_{\mathrm{OC}} = -\sum_i^{|\mathcal{K}|} \Big(& y_i \cdot \log \sigma(\mathrm{sim}(\boldsymbol{f}_o, \boldsymbol{f}_{k_i})) \\ & + (1 - y_i) \cdot \log(1 - \sigma(\mathrm{sim}(\boldsymbol{f}_o, \boldsymbol{f}_{k_i}))) \Big) \end{aligned} \tag{3}$$

where $y_i \in \{0, 1\}$ indicates the ground-truth label for the i-th knowledge.

Upon successful knowledge prediction, the OC model can identify the relation and tail of the object's knowledge. To infer labels from the predicted knowledge $\hat{k}$, we search for a knowledge source $\mathcal{K}$ that satisfies the predicted relation and the tail conditions. The corresponding head of the matching knowledge serves as the predicted label. This process allows the OC model to recognize and classify objects effectively based on acquired knowledge.

4.3 Question Generator

In our question generation model, we employed a vision-and-knowledge encoder based on the state-of-the-art vision-and-language model ViLT (Kim et al., 2021) as the encoder and GPT-2 (Radford et al., 2019) as the decoder. The overall architecture is shown in Fig. 7. The motivation for using these models is their proven performance in handling both visual and textual data, which is essential for generating meaningful and knowledge-aware questions.

The ViLT encoder $\mathrm{Enc}(\cdot)$ takes two inputs: (1) the input image I and the masked region image I_R and (2) knowledge triplets k in sentence form, such as ⟨`[MASK] is a mammal`⟩. A masked region image is created by setting the pixel value outside the target region to zero.

In the knowledge encoder, each word in the masked knowledge is embedded into the knowledge embedding space $\boldsymbol{k}_i \in \mathbb{R}^D$, where i denotes the word index and D denotes the dimension of the embedding space. The knowledge embedding vector is thereafter summed with the modal-type embedding $\boldsymbol{k}_{\mathrm{type}} \in \mathbb{R}^D$ and the positional embedding $\boldsymbol{k}_{\mathrm{pos}} \in \mathbb{R}^D$.

The visual encoder of ViLT processes the input image $I \in \mathbb{R}^{C \times H \times W}$ by dividing it into patches of size $P \times P$ and flattening them into two-dimensional patches $V_p \in \mathbb{R}^{N_p \times (C \times P^2)}$. Here, N_p denotes the number of patches, calculated as $N_p = HW/P^2$. The visual embedding layer embeds the patches in the visual embedding space $\boldsymbol{v} \in \mathbb{R}^{N_p \times D}$. The visual embedding vectors are summed with learnable positional embeddings $\boldsymbol{v}_{\mathrm{pos}} \in \mathbb{R}^{N_p \times D}$ and learnable modal-type embeddings $\boldsymbol{v}_{\mathrm{type}} \in \mathbb{R}^{N_p \times D}$.

For the masked region image I_R, the same embedding layer is used, with the only difference being the use of a different modal-type embedding vector.

Once the visual and knowledge embeddings are obtained, they are concatenated and fed into stacked transformer layers to produce the contextualized embedding vector $\boldsymbol{z}$.

The GPT-2 based decoder, which comprises stacked transformer layers, uses the encoder output $\boldsymbol{z}$ as its initial input. It predicts the next token $\hat{y}_t$ at time step t using the previous word sequences $\boldsymbol{y}_{<t}$ and context vector $\boldsymbol{z}$.

The model is trained to minimize the following loss function:

$$L = -\sum_t^{|y|} \log P(y_t \mid y_{<t}, \mathrm{Enc}([I, I_M], k)) \tag{4}$$

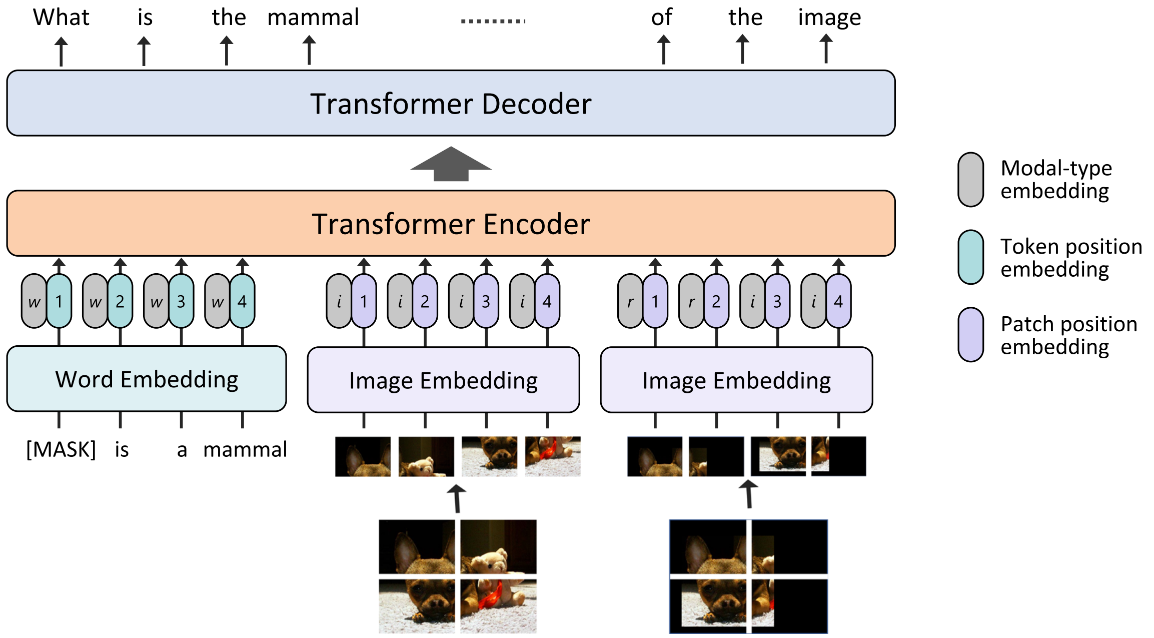


Fig. 7 Overview of our VQG model. The input to the model is the masked target knowledge, the entire image, and the masked image that indicate the region of the target object. The masked target knowledge is embedded into the knowledge embedding space, and the entire and masked images are split into patches and embedded into the visual embedding space. The embedding vectors are concatenated with modal-type embeddings ($\boldsymbol{t}_w$, $\boldsymbol{t}_i$, and $\boldsymbol{t}_r$) and summed with the positional embeddings

where y_t denotes the t-th word of the question, and $\mathrm{Enc}(\cdot)$ represents the ViLT encoder, which is responsible for producing fused visual and textual knowledge features.

4.4 Policy Decision Model

Since the VQG module outputs question q for target knowledge k, the PD module determines the target knowledge k to be used as an input to the VQG module.

First, we explain how the target knowledge is determined. In knowledge acquisition, it is important to acquire knowledge that is "appropriate" and "useful" for recognition, that is, to acquire correct knowledge at the lowest possible cost. Here, "low cost" implies that retraining the OC model should be avoided as much as possible. Therefore, we propose using two different modes of question generation: "confirmation" and "exploration." As described in Sect. 1, the "confirmation" mode is used when the unknown object is relatively close to a known object category, whereas the "exploration" mode is used when the unknown object is far from the existing object category. The target knowledge k for each case is defined as follows:

$$k = \begin{cases} [\texttt{MASK},\ \hat{r},\ \hat{t}\,] & \text{(confirmation)} \\ [\texttt{MASK},\ r^*,\ \texttt{MASK}] & \text{(exploration)} \end{cases} \tag{5}$$

where $\hat{r}$ and $\hat{t}$ denote the predicted relation and tail, respectively, and r^* is an arbitrarily selected relation based on its frequency in the data.

We propose two approaches for the PD module: a naive greedy model and a reinforcement learning based model.

4.4.1 Greedy Model

In the greedy model, we control for the mode selection policy based on the expected value of utility that the model can obtain from the answer. We define the policy selection function π, which takes the value of one for the confirmation mode and zero for the exploration mode.

We thereafter adopt a policy that maximizes the expected utility of the model using the utility function u_θ for the training data $\mathcal{X}$:

$$\frac{1}{|\mathcal{X}|}\sum(\pi\, u_\theta(\boldsymbol{f}_o) + (1-\pi)\, u_\theta(\boldsymbol{f}_o)) \tag{6}$$

We define the utility function as the sum of the "correctness" and "informativeness" of the expected answer. The "correctness" represents the estimated correctness of the knowledge expected to be acquired by the answer. For simplicity, we assume that the oracle answer should be correct and suppose that the expected correctness is 1.0 when the mode is "exploration." In contrast, when the mode is "confirmation," the expected correctness depends on the confidence of the model $\mathrm{conf}(\hat{k})$; thus, we set the expected correctness as the predicted score output by the OC model.

The "informativeness" is the value representing the usefulness of the acquired knowledge to the model. For the "exploration" mode, we estimate the informativeness using the similarity between the input image and target knowledge features $\mathrm{sim}(\boldsymbol{f}_o,\ \boldsymbol{f}_{\hat{k}})$. For the "confirmation" mode, we use the expected value of the similarity based on the mean similarity of the training data, i.e., $\mathbf{E}[I] \simeq \frac{1}{|\mathcal{X}|}\sum \mathrm{sim}(\boldsymbol{f}_o,\ \boldsymbol{f}_{\hat{k}})$.

The utility function is expressed as follows:

$$u_\theta(\boldsymbol{f}_o) = \begin{cases} \mathrm{conf}(\hat{k}) + \mathrm{sim}(\boldsymbol{f}_o,\ \boldsymbol{f}_{\hat{k}}) & (\mathrm{conf.}) \\ 1 + \frac{1}{|\mathcal{X}|}\sum \mathrm{sim}(\boldsymbol{f}_o,\ \boldsymbol{f}_{\hat{k}}) & (\mathrm{exp.}) \end{cases} \tag{7}$$

Once the input knowledge k is determined, question generation is performed using it as the input.

4.4.2 RL-Based Policy Decision Model

In addition to the greedy model, we consider an RL-based model as an improved approach. We construct this RL-based model using a recurrent neural network with four inputs: the region image feature $\boldsymbol{f}_{I_r}$ and current prediction scores $\boldsymbol{f}_{\mathrm{score}}$. We formulate the PD model as follows:

$$a_t = \mathrm{PD}(\boldsymbol{f}_{I_r},\ \boldsymbol{f}_{\mathrm{score}},\ h_{t-1}) \tag{8}$$

where a_t denotes the action at time t and $\boldsymbol{f}_{\mathrm{score}}$ denotes the current prediction score. h_{t-1} denotes the hidden state of the previous time step. We extract the image region feature $\boldsymbol{f}_{I_r}$ using a pre-trained CLIP feature extractor (Radford, 2021), which is the same as that used in the OC module. We use a two-layer LSTM (Hochreiter & Schmidhuber, 1997) for the recurrent neural network.

This PD model is trained to maximize the expected cumulative reward r. The reward consists of the following values: *Target region consistency* r_R This reward is given when the generated question is actually related to the region of the target object. To compute this value, we first calculate the question-to-region grounding score by UNITER-grounding model Chen et al. (2020). The UNITER-grounding model takes questions q and the image I as inputs, and outputs the probability of the image region the question is related to. We thereafter calculate the Intersection over Bounding Box (IoBB) score between the target region and the region with the highest probability.

The reward is computed as follows:

$$r_R = \begin{cases} 1.0 & \text{if IoBB} > \theta \\ 0.0 & \text{otherwise} \end{cases} \tag{9}$$

The threshold θ is set to 0.4.

Informativeness r_I This value implies how informative the question is, i.e., how much recognition performance of the object recognition model can be improved by adding the knowledge obtained by the generated questions. To compute this value, we use the Oracle Answerer model to provide the answer to the generated question. The Oracle Answerer model takes question q and image I as inputs, and outputs answer k_a. The details of the Oracle Answerer model are described in the following section. We thereafter calculate the recognition performance of the object recognition model before and after adding the knowledge obtained from the generated questions. We use the difference between the recognition performance before and after adding knowledge as a reward. The computation of this reward is as follows:

$$k_a = \mathrm{OA}(\hat{q},\ I) \tag{10}$$

$$K_y^+ = K_y \cup \{k_a\} \tag{11}$$

$$r_I = \mathrm{score}(y \mid \boldsymbol{f}_o,\ K_y^+) - \mathrm{score}(y \mid \boldsymbol{f}_o,\ K_y) \tag{12}$$

Consequently, the expected cumulative reward r is computed as follows:

$$r = r_R \cdot r_I \tag{13}$$

In addition, we set certain constraints on the action selection. First, the model was not allowed to select the confirmation mode multiple times. This is because the target knowledge of the confirmation mode relies purely on the initially predicted knowledge; thus, the question target never changes throughout the time steps. Second, if the model outputs the *no-question* mode, it is not allowed to select any other mode for the remaining time steps. This is because the model has already decided that it has completed the gathering of the necessary knowledge; thus, it does not need to ask questions.

We train the PD model using the REINFORCE (Williams et al., 1992) algorithm. The gradient of the PD model is calculated as follows:

$$\nabla_\theta J(\theta) = \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t \mid \boldsymbol{f}_{I_r},\ \boldsymbol{f}_{\mathrm{score}},\ h_{t-1}) \cdot \sum_{t'=t}^{T} \gamma^{t'-t} \exp(r_{t'}) \tag{14}$$

where θ denotes the parameter of the PD model, $\pi_\theta(a_t \mid \boldsymbol{f}_{I_r},\ \boldsymbol{f}_{\mathrm{score}},\ h_{t-1})$ is the probability of the action a_t given the region image $\boldsymbol{f}_{I_r}$, the current prediction scores $\boldsymbol{f}_{\mathrm{score}}$, and the hidden state h_{t-1}, and T denotes the number of time steps. We set the discount factor γ to 0.99.

In addition to the policy gradient loss, we train the PD model to minimize entropy loss, which is calculated as the

Shanon's entropy of the action distribution. The entropy loss is calculated as follows:

$$L_{\text{entropy}} = -\sum_{t=1}^{T} \sum_{a_t} \left(\pi_\theta(a_t \mid f_{I_r},\ f_{\text{score}},\ h_{t-1}) \cdot \log \pi_\theta(a_t \mid f_{I_r},\ f_{\text{score}},\ h_{t-1})\right) \tag{15}$$

This entropy loss is used to encourage the model to explore various actions and avoid becoming stuck in a specific action.

The entire loss function is calculated as the sum of the policy gradient and entropy loss as follows:

$$L = L_{\text{policy}} + \alpha\, L_{\text{entropy}} \tag{16}$$

The balancing factor α to 0.01.

4.5 Oracle Answerer

Given an image and generated question, Oracle Answerer predicts the answer knowledge for the question. We implement this module as a composition of three submodules: (1) Head classifier, (2) Relation classifier, and (3) Region classifier. Each module checks whether the generated question is "valid," and if all modules agree that the question is "valid," Oracle Answerer searches the oracle knowledge source and outputs the knowledge that matches the targeted head and relation. Oracle knowledge source is a knowledge source that merges ConceptNet (Speer et al., 2017) and ATOMIC$^{20}_{20}$ (Hwang et al., 2020) . The overall architecture of the oracle answerer is illustrated in Fig. 8.

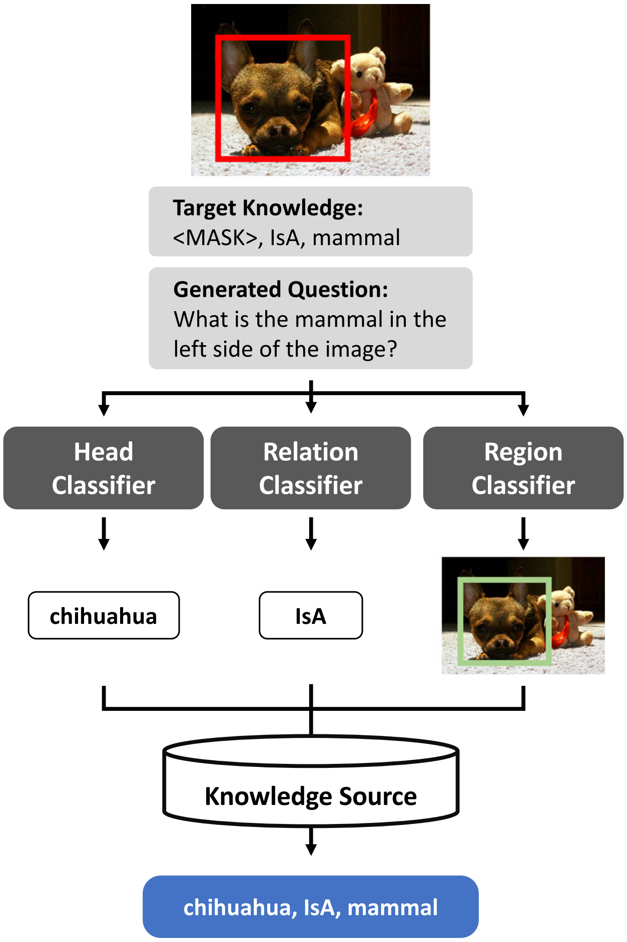


Fig. 8 Architecture of the oracle answerer

Head classifier The head classifier $\mathcal{H}$ predicts the head of the target knowledge from the generated question, that is $h = \mathcal{H}(I, Q)$. We implement this module following the standard VQA methodology, that is, as a multi-class classification problem that outputs the proper entity given an image and question. For this module, we fine-tuned pre-trained ViLT-VQA (Kim et al., 2021) model. This module returns "valid" if the predicted head is equal to the object in the target region.

Relation classifier The relation classifier $\mathcal{R}$ predicts the relation of the target knowledge from the generated question, that is, $r = \mathcal{R}(Q)$. Since this problem can be formulated as a sentence classification problem, we use the fine-tuned Distil-BERT Sanh et al. (2019) as the relation classifier. This module returns "valid" if the predicted relation matches the target relation (r in Eq. 5).

Region classifier The region classifier $\mathcal{G}$ predicts the target region, that is, $g = \mathcal{G}(I, Q)$. We design this module as a model that outputs the region most relevant to the question, given a question and a set of candidate regions. The problem setup is similar to that of the Referring Expression Comprehension (RE Comprehension) (Yu et al., 2016). Therefore, we used a fine-tuned version of the UNITER grounding model (Chen et al., 2020), which achieved high performance in the RE Comprehension task. This module returns "valid" if the predicted region is sufficiently close to the target region. We calculated the IoBB (Intersection over Bounding Box) between the predicted and target regions and considered two regions sufficiently close if the value was greater than 0.4.

Oracle Knowledge Source The Oracle Knowledge Source is used to provide the answer knowledge to the generated question. To build such knowledge source, it is important to collect as much correct knowledge as possible. Therefore, we extend the original knowledge source in the dataset. The extension of the knowledge source is performed in the following steps:

1. Collect the knowledge from the train and validation datasets.
2. Add all the knowledge from the original knowledge source, ConceptNet (Speer et al., 2017) and ATOMIC$^{20}_{20}$ (Hwang et al., 2020), whose head entity is already contained in the dataset.

```
'stool, UsedFor, use as a step',
'pillow, UsedFor, Keep head comfortable
    ',
'cab, UsedFor, ride to their home',
'bread, UsedFor, offer someone else',
'briefcase, MadeUpOf, plastic',
'window, UsedFor, lock when she leaves
    ',
'cardigan, UsedFor, hide themselves',
'sack, HasA, strap',
'knife, UsedFor, outdoors living',
'jump suit, UsedFor, buy',
'ottoman, IsA, chair',
'octopus, IsA, animal',
'coffee cup, AtLocation, restaurant',
'sausage, UsedFor, pass around',
'ruler, UsedFor, measure the distance',
'segway, UsedFor, travel to the venue',
'container, CapableOf, amount water',
'bath mat, UsedFor, wash face',
'coffee cup, AtLocation, airport',
'soccer ball, IsA, burgoise'
```

Listing 1 The list of randomly sampled oracle knowledge.

3. For each knowledge in the collected knowledge by the previous step, we add the knowledge whose head entity is a synonym of the head entity of the original knowledge. To determine whether the head entity is a synonym of the head entity of the original knowledge, we use the pre-trained word embeddings from ConceptNet (Speer et al., 2017). We calculate the similarity using the cosine similarity between the word embeddings of the head entity and all candidate head entities in the data. If the similarity is higher than 0.5, we add the knowledge of the candidate head entity.

Using these procedures, a large amount of knowledge that is related to the dataset was collected. The original knowledge source in the training and validation datasets contain 8585 knowledge, while the extended knowledge source contains 124,326 knowledge. Examples of additional knowledge in the extended knowledge source are listed in List 1.

To obtain the answer knowledge, we search the oracle knowledge source for the knowledge whose head entity is the same as the target head and the predicted relation. If there is no knowledge that satisfies the condition, the oracle answerer returns the answer as "invalid."

4.6 Knowledge Expansion

When an answer knowledge k' is obtained for a generated question q by the model, it is added to the model's knowledge source $\mathcal{K}$, that is, $\mathcal{K}^{+} = \mathcal{K} \cup \{k'\}_{i=1}^{M}$, where M denotes the number of newly acquired knowledge.

Avoiding redundancy To avoid asking redundant questions, we use different types of QG methods: neural-based QG (as described above) and rule-based QG. The rule-based QG method uses simple rules to generate questions for the input knowledge, e.g., ⟨`[MASK]`, `UsedFor`, `[MASK]`⟩ → "What is the object *used for?*" or ⟨`[MASK]`, `MadeUpOf`, `[MASK]`⟩ → "What is the object *made of?*"

Neural-QG is better at generating questions that reflect the image content and target knowledge in detail, and at generating questions that allow answerers to clearly identify the target object. However, when considering its use in multi-turn questions, once a question that can clearly identify the target object is generated, there is no need for further information to identify the target object in subsequent questions. For instance, if the first question is "What is the object sitting next to the dog?," the answerer can easily identify the target object as a *teddy-bear*. Therefore, in subsequent questions, it is not necessary to include spatial information, such as *next to the dog*.

We make the decision of which QG method to use based on the Region Classifier model, which can identify the region of the image referred to in the question. We calculate IoBB between the ground-truth target region and the predicted region for each question If the IoBB up to the present question is greater than the threshold, we use the rule-based QG method. Otherwise, we use the neural-based QG method.

$$q = \begin{cases} \text{Neural-VQG}(I,\ a) & \text{if IoBB}_{<t} < \theta \\ \text{Rule-VQG}(a) & \text{otherwise} \end{cases} \tag{17}$$

where I denotes the input image, a denotes the action determined by the PD model, t denotes the current turn, and θ denotes the threshold.

5 Experiments

5.1 Training

We used the same text encoder as CLIP (Radford, 2021) and ViT-B/32 (Dosovitskiy et al., 2021) as the visual encoder in the OC model. The OC model is trained from a pre-trained checkpoint of CLIP.[1] The number of training epochs was 200, with a cosine learning rate scheduler and a warmup ratio of 0.2. We used the Adafactor optimizer (Shazeer & Stern, 2018) with learning rate 8e−5 and weight decay 0.01. The training of the OC model required about 12 h on 8×Tesla A100 GPUs with a batch size of 512.

[1] https://huggingface.co/openai/clip-vit-base-patch32.

We tested all methods in two settings: zero-shot and fine-tuned. In the zero-shot setting, we did not conduct any fine-tuning on the OC model with the knowledge acquired by the QG model. In the fine-tuned setting, we fine-tuned the OC model using the knowledge obtained. To maintain the performance on known classes in a fine-tuned setting, we adopted simple replaying methods in which the same number of samples as the newly acquired data were randomly sampled from the training set and input to the model along with the newly acquired knowledge. For fine-tuning, we trained the OC model for 40 epochs with a learning rate of 8e−5 and weight decay of 0.2, clipping the gradient norm to 0.1.

In the VQG model, we used the pre-trained ViLT Kim and Son (2021) encoder[2] as the multi-modal encoder and the pre-trained GPT-2 Radford et al. (2019) decoder[3] as the decoder.

5.2 Baselines

We compared our approach to four baselines: *CLIP-Ret.* In this setting, no knowledge acquisition is performed using the QG model and the performance of the OC model trained using only the training set is evaluated. *All Exp./All Conf.* In these settings, the question generation policy is fixed to "exploration" and "confirmation," respectively. *Random Policy.* The question generation policy is selected randomly. This method was tested three times using different random seeds.

It is important to note, as detailed in Sect. 2, that none of the previous methods are designed to generate questions targeting knowledge or specific regions within an image. Consequently, these methods could not be adopted as baseline approaches for our study. Even if these methods were utilized, due to the outlined limitations, they would fail to generate questions that accurately target the correct knowledge or regions. This shortcoming is expected to result in a significant reduction in the quality of the generated questions, leading to an overall decrease in performance when compared to the proposed method and other baselines.

Furthermore, we conducted an ablation study concerning the algorithm of the model. Specifically, within the PD model, we tested versions that did not utilize region consistency (*w/o region cons.*) and informativeness (*w/o informativeness*) for reward calculation. These were implemented by setting r_R and r_I to 1.0 in Eq. (13) respectively.

5.3 Evaluation Metrics

Following previous studies on multi-label object recognition (Huynh & Elhamifar, 2020; Ben-Cohen et al., 2021), we evaluated the performance of the proposed model using the mean average precision (mAP). We computed the average precision (AP) for each class c as follows:

$$\mathrm{AP}(c) = \frac{1}{N_c} \sum_{k=1}^{N} \mathrm{Precision}(k,\ c) \tag{18}$$

where N_c denotes the number of examples with label c, $\mathrm{Precition}(k,\ c)$ denotes the precision at the k-th ranked prediction.

We calculate the mAP for known and novel classes separately.

To calculate the AP for each class, we considered labels that satisfied the following conditions as ground-truth labels: First, we considered the ground-truth labels in the original dataset for the target region as the initial set of ground-truth labels for the given target region R. Second, we added objects to the overlapping region of R. The overlapping region was defined as the region in which the IoBB is greater than 0.4. Finally, we added the synonyms of the labels to the set of ground-truth labels. We used the same synonym list as Oracle Answerer.

5.4 Results and Discussion

The main results are shown in Table 3.

We compare the performance of the baseline (CLIP-Ret.), single-turn methods, and five-turn methods, as well as the zero-shot and fine-tuning settings.

When comparing the baseline CLIP-Ret. to other methods, the baseline is inferior in all metrics. This highlights the effectiveness of knowledge acquisition through question generation for improving object recognition performance, particularly for novel classes, which are more challenging to recognize without additional information.

For single-turn settings, our Greedy method outperforms both All Conf. and All Exp. in all metrics, achieving the highest overall mAP, known class mAP, and novel class mAP. This demonstrates the effectiveness of our Greedy approach in acquiring useful knowledge for object recognition with just one question generation turn.

In the five-turn settings, our RL Policy method attains the best performance among all metrics, showing substantial improvement over the All Exp. and Random methods. Moreover, the standard deviations of our RL Policy method are relatively small, indicating the stability of our approach across multiple runs.

When comparing single-turn and five-turn methods, we observe that the five-turn methods generally yield better performance, particularly in the fine-tuning setting. This improvement is most prominent in novel class mAP, which supports the notion that our model successfully learns to

[2] https://github.com/dandelin/ViLT.

[3] https://github.com/openai/gpt-2.

Table 3 The results of the object recognition model after obtaining the knowledge by asking questions

	Overall		Known		Novel	
	Zero-shot	Fine-tune	Zero-shot	Fine-tune	Zero-shot	fine-tune
Single						
Baseline (CLIP-Ret.)	12.10	–	12.26	–	6.86	–
All Conf	12.92	12.94	12.59	12.61	24.02	23.87
5-turn						
All Exp	12.97	13.26	12.56	12.81	27.14	28.40
Greedy	12.99	13.32	12.59	12.92	26.64	26.97
All Exp	13.48	14.69	13.02	14.13	29.12	33.85
Random	13.52 ± 0.15	14.62 ± 0.01	**13.06 ± 0.13**	13.99 ± 0.01	29.12 ± 0.55	35.98 ± 0.15
RL	**13.54 ± 0.08**	**15.32 ± 0.1**	**13.06 ± 0.09**	**14.48 ± 0.12**	**29.59 ± 0.23**	**43.52 ± 0.42**
—w/o region cons	13.17	14.81	12.70	14.00	29.17	42.41
—w/o informativeness	13.38	14.48	12.91	13.86	29.22	35.72

Bold values indicate the best results

The results are shown in terms of mAP for overall classes, known classes, and novel classes. The baseline (CLIP-Ret.) refers to the performance without additional question generation or knowledge acquisition. The middle group of rows shows the results when question generation is performed for only one turn. The bottom group presents examples of the outcomes when question generation is conducted for multiple turns (five turns) per target

Table 4 Performance variations of VQG resulting from the replacement of individual components with different structures

	BLEU-4	METEOR	CIDEr	Mean IoU
Confirmation				
UNITER + BART	16.95	**22.71**	**113.39**	0.45
ViLT + GPT-2	**17.15**	21.94	97.81	**0.49**
—w/o image	14.22	20.10	85.90	0.44
—w/o region	15.57	20.85	86.47	0.45
—w/o knowledge	8.9	14.59	20.31	0.36
Exploration				
UNITER + BART	**9.62**	**16.82**	**39.17**	0.28
ViLT + GPT-2	9.51	16.57	38.09	**0.40**
—w/o image	7.83	14.19	14.08	0.23
—w/o region	8.00	15.18	26.45	0.26
—w/o knowledge	6.52	13.96	20.72	0.32

Bold values indicate the best results

We used two types of the encoder: UNITER or ViLT, and two types of the decoder: BART or GPT-2

select a policy that generates questions and acquires useful knowledge for recognizing novel objects.

From the results of the ablation study, it is evident that both region consistency and informativeness in reward calculation effectively contribute to acquiring novel information through question generation. Notably, the recognition performance for novel objects during fine-tuning exhibited a significant drop under the setting without informativeness. This can be attributed to the fact that, without considering informativeness during reward computation, questions tend to acquire redundant knowledge. Specifically, they pose questions with low information target, hindering the acquisition of diverse knowledge regarding novel objects.

Table 5 Performance variations of VQG from different training dataset

	BLEU-4	METEOR	CIDEr	Mean IoU
Confirmation				
CRIC	2.09	12.16	15.23	0.35
K-VQG v1	13.75	19.62	67.57	0.45
K-VQG v2	**17.15**	**21.94**	**97.81**	**0.49**
Exploration				
CRIC	0.87	10.98	8.43	0.32
K-VQG v1	6.60	14.06	19.83	0.31
K-VQG v2	**9.51**	**16.57**	**38.09**	**0.40**

Bold values indicate the best results

We used ViLT + GPT type model for this experiments

5.5 Model Component Variations

Here, we conduct experiments to see the performance changes when varying the structures of individual components and provide a detailed analysis of the results. In the main result, we used the pre-trained ViLT (Kim et al., 2021) based model as the encoder, and the GPT-2 (Radford et al., 2019) based model as the decoder. Here, we experimented with counterpart models, one using the pre-trained UNITER (Chen et al., 2020) as the encoder and the BART (Lewis et al., 2020) as the decoder. The UNITER model is one of the large-scale pre-trained multi-modal encoder, and the BART model is an encoder-decoder pre-trained text generation model.

In addition, we conducted an ablation study to investigate the question generation performance when altering the model's input components. Specifically, we evaluated scenarios where each of the three inputs—the entire image, the region image, and the target knowledge—was individually omitted from the model's input. This ablation study is done for ViLT + GPT-2 model, which is used in our primary experiments.

For all models, we report the results of "confirmation" setting and "exploration" setting. As described in Sect. 4.3, in former setting, the model is given the head-masked target knowledge as the input. In the latter setting, the model is given the target knowledge in which the head and tail are masked.

As the evaluation metric, we used BLEU-4 (Papineni et al., 2002), METEOR (Denkowski & Lavie, 2014), CIDEr (Vedantam et al., 2015), and Mean IoU. The BLEU, METEOR, and CIDEr scores are the metrics to evaluate the quality of the generated questions compared to the ground-truth questions. The Mean IoU (Intersection over Union) is a metric that evaluates whether the question is about the correct region in the image. We compute the IoU between the predicted region of the generated question and the ground-truth question. To predict the target region of the question, we used region grounding model $G(I_r \mid q, \ I)$, which predicts the target region of the question I_r from the question q. We built the grounding model based on UNITER grounding model (Chen et al., 2020), same as the region classifier in the Oracle Answerer model.

We summarize the results in Table 4. In terms of question quality, as measured by BLEU-4, METEOR, and CIDEr, the differences between the primary models, UNITER + BART and ViLT + GPT-2, are minimal. However, the distinction becomes more evident when examining target region correctness, with the Mean IoU scores indicating notable differences between these architectures.

When assessing the influence of individual inputs, the omission of the image input leads to a pronounced reduction in performance metrics across both modes. This highlights the importance of the image context in achieving high-quality question generation. The noticeable drop in performance when knowledge input is removed underscores its critical role in generating coherent and contextually appropriate questions.

While the distinction between the ViLT + GPT-2 and UNITER + BART architectures does not significantly influence the overarching quality of questions, it does impact the precision of region targeting. More significantly, the alteration in key inputs (image, region, or knowledge) seems to have more impact on performance. It implies that the high-level model structures we proposed, such as the encoding of region information and the introduction of knowledge embeddings, contribute significantly to the performance.

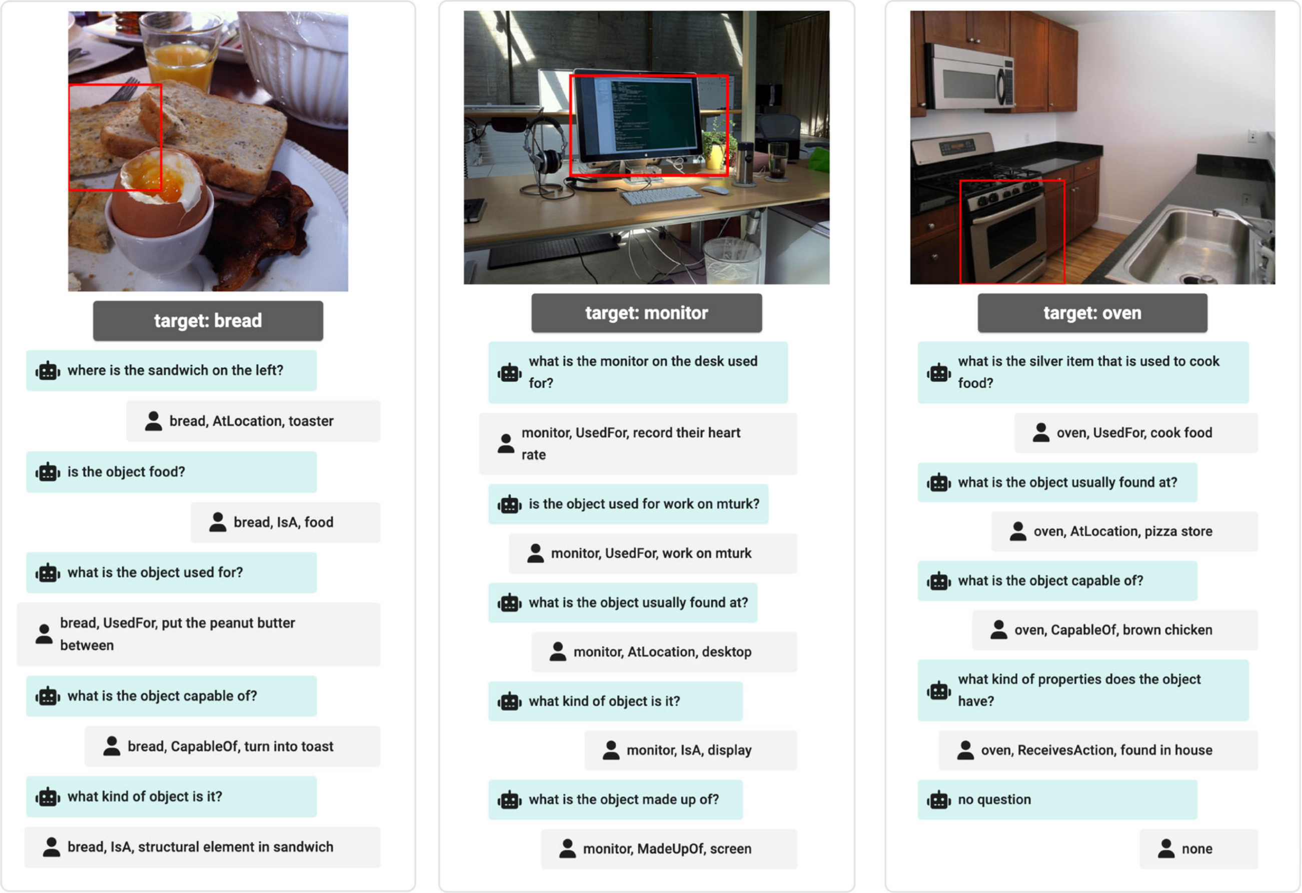
5.6 Dataset Variations

This section presents the comparative outcomes of VQG using diverse datasets. We summarize the results in Table 5. As highlighted in Sect. 2, datasets fulfilling all required criteria such as being manually created, containing region bounding boxes, and targeting knowledge acquisition are scarcely available. To demonstrate the efficacy of the dataset curated for this study, we conducted experiments using the newly constructed K-VQG v2 dataset, the smaller-scale K-VQG v1 dataset annotated via crowdsourcing, and the CRIC dataset, which is generated based on a rule-based algorithm rather than manual annotation.

We used the same architecture and training settings as the main experiments, i.e., ViLT + GPT-2. To evaluate the result under constant criteria, evaluations were conducted using the validation split from the K-VQG v2 dataset. The evaluation metrics adopted were consistent with Sect. 5.5, including BLEU-4, METEOR, and CIDEr for assessing the quality of the generated question, along with Mean IoU to measure how well the generated questions corresponded to the target regions.

The results indicate that using the K-VQG v2 dataset resulted in superior quality of the generated questions and a higher degree of alignment with the target regions compared to the other datasets. This superior performance is believed to be influenced by both the quantity and quality of the data. For instance, the K-VQG v2 dataset is approximately 1.5 times larger than K-VQG v1. Moreover, it is presumed that the K-VQG v2, written by human annotators, contains a more diverse and natural questions compared to the rule-based CRIC dataset.

These results underscore the suitability of our K-VQG v2 dataset for constructing models for the task of generating visual questions that acquire knowledge about target objects, as required for our research.



target: bread

where is the sandwich on the left?

bread, AtLocation, toaster

is the object food?

bread, IsA, food

what is the object used for?

bread, UsedFor, put the peanut butter between

what is the object capable of?

bread, CapableOf, turn into toast

what kind of object is it?

bread, IsA, structural element in sandwich

target: monitor

what is the monitor on the desk used for?

monitor, UsedFor, record their heart rate

is the object used for work on mturk?

monitor, UsedFor, work on mturk

what is the object usually found at?

monitor, AtLocation, desktop

what kind of object is it?

monitor, IsA, display

what is the object made up of?

monitor, MadeUpOf, screen

target: oven

what is the silver item that is used to cook food?

oven, UsedFor, cook food

what is the object usually found at?

oven, AtLocation, pizza store

what is the object capable of?

oven, CapableOf, brown chicken

what kind of properties does the object have?

oven, ReceivesAction, found in house

no question

none

Fig. 9 Qualitative examples of the multi-turn question generation

5.7 Qualitative Examples

We present qualitative examples of our model with RL policy in Figs. 9 and 10.

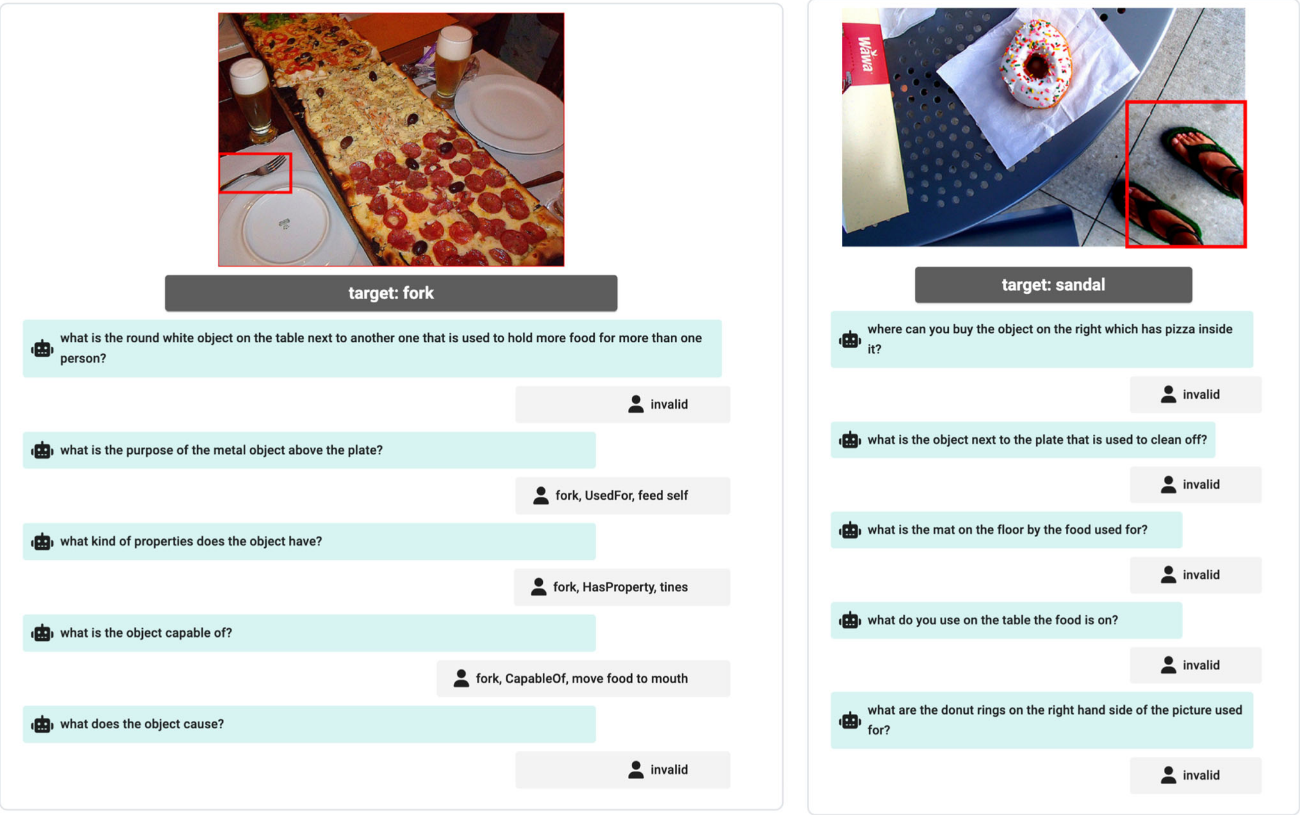
In the leftmost example of Fig. 9, the target object is "bread," which is a novel class. The model first asks a question in exploration mode, that is, the target knowledge is ⟨[MASK], AtLocation, [MASK]⟩. Since the first question is deemed as valid, the model asks a second question in confirmation mode, that is, the target knowledge is ⟨[MASK], IsA, food⟩, using a Rule-VQG model.

In the middle example, the target object is "monitor," which is also a novel class. In this case, the model first asks a question in the exploration mode in which the target knowledge is ⟨[MASK], UsedFor, [MASK]⟩. Since the question is deemed as valid, the next question is asked in the confirmation mode; the target knowledge is ⟨[MASK], UsedFor, work on mturk⟩, and the subsequent questions are in the exploration mode.

In the rightmost example, in the fifth turn, the model decides to discontinue the question generation ("no question"). As shown in this example, our model can discontinue question generation when it has obtained sufficient knowledge to recognize the target object.

In Fig. 10, we present examples in which the model failed to generate valid questions. In the left example, the first question "what is the round white object on the table next to another one that is used to hold more food for more than one person?" was considered invalid by Oracle Answerer. In this case, the generated question seems to incorrectly target "plate" in the image, while the correct target object is "fork." The second question, "what is the purpose of the metal object above the plate?," is correctly targeted to the fork. Thus, the model can obtain knowledge ⟨fork, UsedFor, feed self⟩.

In the example on the right, the model failed to generate valid questions for all five turns. In this case, the model continually asks questions about the objects around the donut, which is placed in the middle of the image, while the correct target object "sandal" is placed in the right bottom area of the image. This is attributed to the VQG model's limited ability to correctly localize the target object in the image.



target: fork

what is the round white object on the table next to another one that is used to hold more food for more than one person?

invalid

what is the purpose of the metal object above the plate?

fork, UsedFor, feed self

what kind of properties does the object have?

fork, HasProperty, tines

what is the object capable of?

fork, CapableOf, move food to mouth

what does the object cause?

invalid

target: sandal

where can you buy the object on the right which has pizza inside it?

invalid

what is the object next to the plate that is used to clean off?

invalid

what is the mat on the floor by the food used for?

invalid

what do you use on the table the food is on?

invalid

what are the donut rings on the right hand side of the picture used for?

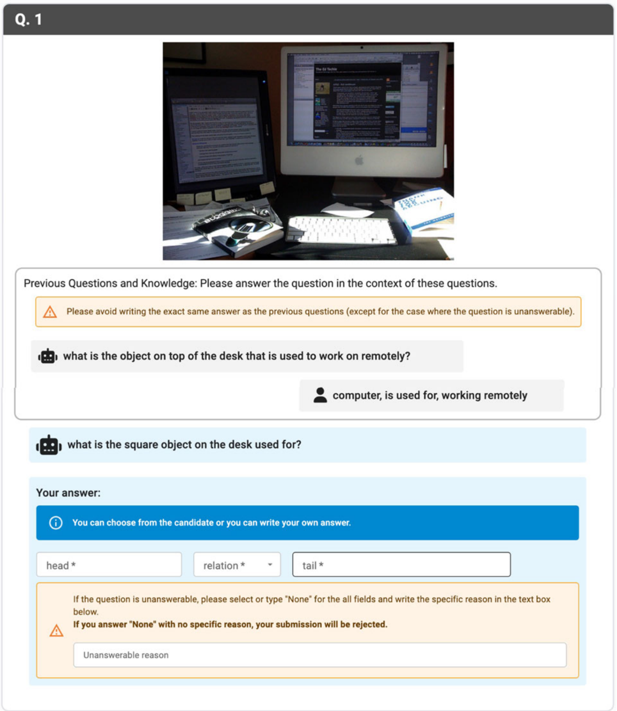
invalid

Fig. 10 Qualitative examples of the multi-turn question generation in which the model failed to generate valid questions

6 Human Evaluation

We employed human evaluations to assess the usefulness of the questions generated by our model for recognizing novel classes. To accomplish this, we used AMT as the evaluation platform. Since real-time question generation by the model is difficult to achieve, we used the following procedure. The initial question pertaining to the image was generated in advance on a local server utilizing the pre-trained model. Subsequently, the generated questions were submitted to AMT and workers were asked to provide the appropriate knowledge as answers. Once the answers to the initial question were collected, the initial question and workers' answers were fed into the trained model to generate the second question. The second question, along with the history of previous interactions (initial question and answer), was thereafter presented to the worker, who was prompted to provide an answer to the new question. This process was repeated for up to five questions. In Fig. 11, we present an example of a user interface for workers based on human evaluations.

We performed human evaluations for the object "monitor." We established three criteria for selecting AMT workers to ensure the highest possible data quality. First, the hit approval rate for all requesters' hits must be greater than 95%, which is considered to be a high bar for requesters. Second, the



Q. 1

Previous Questions and Knowledge: Please answer the question in the context of these questions.

⚠ Please avoid writing the exact same answer as the previous questions (except for the case where the question is unanswerable).

what is the object on top of the desk that is used to work on remotely?

computer, is used for, working remotely

what is the square object on the desk used for?

Your answer:

ⓘ You can choose from the candidate or you can write your own answer.

head * | relation * | tail *

⚠ If the question is unanswerable, please select or type "None" for the all fields and write the specific reason in the text box below.
If you answer "None" with no specific reason, your submission will be rejected.

Unanswerable reason

Fig. 11 User interface for human evaluation

Table 6 Performance of the object recognition using the acquired knowledge from human evaluation

	Top-1 Acc	Top-5 Acc	Mean Rank
Ours (zero-shot)	0.0	84.8	5.3
Ours (fine-tuned)	**60.9**	**91.3**	**2.4**

Bold values indicate the best results
The performance was assessed under two settings: without fine-tune (zero-shot) and with fine-tune. Note that the lower the mean rank, the better the performance

workers had to be located in Canada, the United Kingdom, or the United States. Finally, we only considered workers who had been granted "Masters" status, which AMT awards to workers who have consistently demonstrated a high level of performance.

We obtained 225 responses (45 images, five questions per image). This resulted in 176 new knowledge obtained. Of these, knowledge with the head "monitor" was the most common, 35 new knowledge. The next most common head in the obtained knowledge were "desk" (22) and "laptop" (17). However, 22 questions were deemed invalid.

The performance of the object recognition using the acquired knowledge was thereafter assessed under two settings: without fine-tuning (zero-shot) and with fine-tuning. We evaluated the performance using accuracy and mean rank of "monitor" and the results are summarized in Table 6. Note that the metrics are calculated with the data that have "monitor" as the ground truth, as we only gathered knowledge for "monitor."

In the zero-shot setting, the accuracy for "monitor" was 0.0 and its rank was 5.33, while after fine-tuning was performed, the accuracy for "monitor" increased to 60.87 and its rank improved to 2.40. This indicates that the knowledge acquired from the human evaluation was not able to raise the prediction score of "monitor" to the point where it was predicted to be the top among the other classes without fine-tuning.

Notably, the mean rank of "monitor" was not extremely bad, considering that the number of all classes was 598. After fine-tuning, the accuracy and mean rank of "monitor" were significantly improved. From these results, we can conclude that the knowledge acquired from human evaluations is useful for novel object recognition.

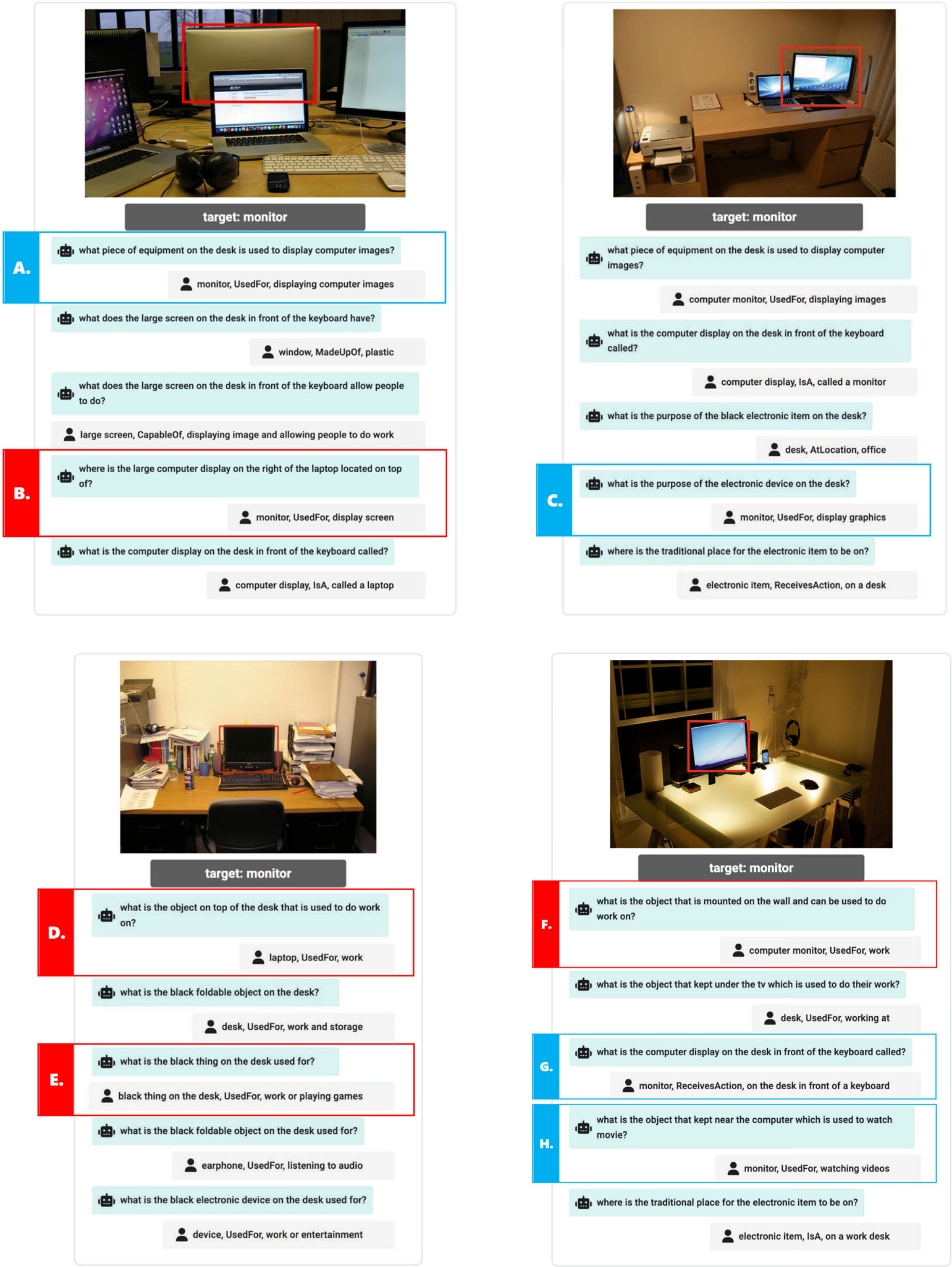
Examples of questions and answers are shown in Fig. 12. We highlight some of the questions and answers in the figure (A $\sim$ H). In answers (A, C, G, and H), the workers provided correct knowledge about the object *monitor*, such as ⟨`monitor, UsedFor, displaying computer images`⟩ (A), and ⟨`monitor, UsedFor, display graphics`⟩ (C). In these cases, the questions are concrete and easy to understand. For instance, from the question of A, "what piece of equipment on the desk is used to display computer images?," we can easily understand the question is about the monitor on the desk, and the required knowledge is whether the object is used to display computer images. In contrast, B, D, E, and F are examples of failed questions and answers. For B, the question seemed to be about the monitor and its typical location, but the answer was about the usage of the monitor (⟨`monitor, UsedFor, display screen`⟩). This indicates that the given task should be performed with caution, as there is a significant chance of misunderstanding or lack of seriousness among the workers. The case of E is similar to that of B; it is probable that the worker misunderstood the instruction, resulting in knowledge having a head of the *black thing on the desk*, which is a phrase from the original question, instead of an entity name, such as *monitor*, as it should have been.

In D and F, the workers provided knowledge about incorrect, but similar, or near-located objects (e.g., laptop or computer monitor). This was attributed to a lack of clarity in the questions. For instance, in D, the question is "what is the object on top of the desk that is used to do work on?," and the *monitor* and *laptop* are both located on the desk and used to do work on.

From these examples, we found that it is essential to ensure that the questions are clear and that the workers fully understand their task before beginning, or to provide a training session for the workers.

In addition, we present example of the knowledge obtained from human answerers in Fig. 13. By our method, the model successfully acquired various knowledge, i.e., various relations and tails for the head "monitor," such as ⟨`monitor, AtLocation, desk`⟩ or ⟨`monitor, CapableOf, display images`⟩. We observe that the knowledge corresponding to the relation "UsedFor" and "IsA" tends to be collected more than other relations. This is the same tendency as in the previous section and can also be explained by the imbalance of relations in the relying dataset. We believe that the model can acquire more knowledge of rare relations in the future when more data for rare relations are collected or when the model is trained to generate more questions for rare relations.

We observed certain tails that, though not exact matches, are semantically analogous (e.g., "displaying computer images" and "displaying images" or "playing computer games" and "playing games"). This is not surprising in the current context because semantically equivalent tails may be expressed differently in natural languages. However, from the perspective of computational complexity, it is desirable to avoid adding different, yet semantically analogous, tails to the knowledge source. This finding indicates the need for further exploration of how a knowledge base may be structured to store vast amounts of knowledge efficiently, while compressing semantically similar tails in the most compact manner.



🤖 what piece of equipment on the desk is used to display computer images?

👤 monitor, UsedFor, displaying computer images

🤖 what does the large screen on the desk in front of the keyboard have?

👤 window, MadeUpOf, plastic

🤖 what does the large screen on the desk in front of the keyboard allow people to do?

👤 large screen, CapableOf, displaying image and allowing people to do work

🤖 where is the large computer display on the right of the laptop located on top of?

👤 monitor, UsedFor, display screen

🤖 what is the computer display on the desk in front of the keyboard called?

👤 computer display, IsA, called a laptop

🤖 what piece of equipment on the desk is used to display computer images?

👤 computer monitor, UsedFor, displaying images

🤖 what is the computer display on the desk in front of the keyboard called?

👤 computer display, IsA, called a monitor

🤖 what is the purpose of the black electronic item on the desk?

👤 desk, AtLocation, office

🤖 what is the purpose of the electronic device on the desk?

👤 monitor, UsedFor, display graphics

🤖 where is the traditional place for the electronic item to be on?

👤 electronic item, ReceivesAction, on a desk

🤖 what is the object on top of the desk that is used to do work on?

👤 laptop, UsedFor, work

🤖 what is the black foldable object on the desk?

👤 desk, UsedFor, work and storage

🤖 what is the black thing on the desk used for?

👤 black thing on the desk, UsedFor, work or playing games

🤖 what is the black foldable object on the desk used for?

👤 earphone, UsedFor, listening to audio

🤖 what is the black electronic device on the desk used for?

👤 device, UsedFor, work or entertainment

🤖 what is the object that is mounted on the wall and can be used to do work on?

👤 computer monitor, UsedFor, work

🤖 what is the object that kept under the tv which is used to do their work?

👤 desk, UsedFor, working at

🤖 what is the computer display on the desk in front of the keyboard called?

👤 monitor, ReceivesAction, on the desk in front of a keyboard

🤖 what is the object that kept near the computer which is used to watch movie?

👤 monitor, UsedFor, watching videos

🤖 where is the traditional place for the electronic item to be on?

👤 electronic item, IsA, on a work desk

Fig. 12 Examples of the questions and answers obtained from human evaluation. For the discussion, we highlighted some of the questions and answers in the figure (A $\sim$ H)

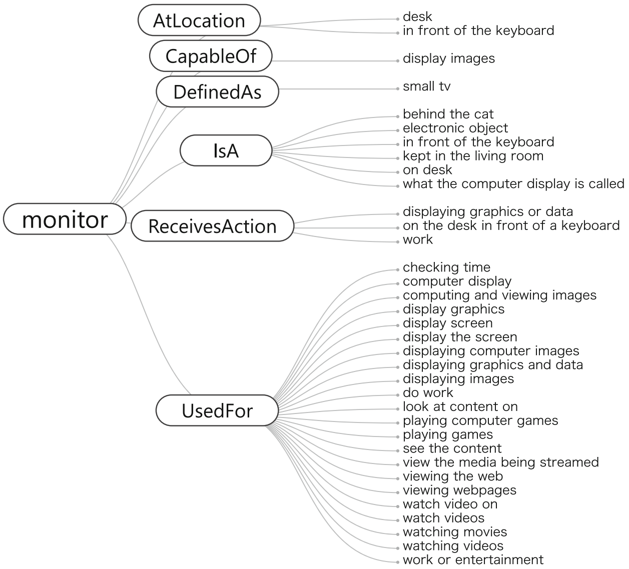


Fig. 13 Visualization of the knowledge acquired for the "monitor" from the human answers

7 Conclusion

In this study, we proposed a multi-turn question generation model that can generate questions for an object recognition model to recognize novel classes. We also proposed a policy network that can select the policy for each action, from the "confirmation" and "exploration" policies, and "no question" policy. We evaluated our model on the K-VQG v2 dataset and demonstrated that it can generate questions useful for recognizing novel classes. By adding newly obtained knowledge to the knowledge source, the model can recognize novel classes while maintaining the performance of known classes, which results in a significant improvement in mAP for novel classes, particularly after fine-tuning the model on the newly obtained knowledge. We also performed a human evaluation to investigate whether the questions generated by our model were useful for recognizing novel classes. From the human evaluation results, we confirmed that our model can generate questions that are useful for recognizing novel classes, even if the answerer is not an oracle VQA model but a human. Despite these successes, our method has a limitation in that the questions must be clear and concrete to enable workers to understand the tasks. Furthermore, we can include an answerer model that resembles the behavior of human answerers, such as the misunderstanding of the question or answering similar but incorrect knowledge. We believe that this limitation can be addressed by deploying this model in real-world applications and continuously collecting data on the behavior of human answerers.

Funding Open Access funding provided by The University of Tokyo. This work was partially supported by JST Moonshot R&D Grant Number JPMJPS2011, CREST Grant Number JPMJCR2015 and Basic Research Grant (Super AI) of Institute for AI and Beyond of the University of Tokyo.

Data Availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

References

Akata, Z., Malinowski, M., Fritz, M., & Schiele, B. (2016). Multi-cue zero-shot learning with strong supervision. In *CVPR*.

Ba, J., Swersky, K., Fidler, S., & Salakhutdinov, R. (2015). Predicting deep zero-shot convolutional neural networks using textual descriptions. In *ICCV*.

Ben-Cohen, A., Zamir, N., Ben-Baruch, E., Friedman, I., & Zelnik-Manor, L. (2021). Semantic diversity learning for zero-shot multi-label classification. In *ICCV*.

Chen, Y.-C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., & Liu, J. (2020). UNITER: Universal image-text representation learning. In *ECCV*.

Chouinard, M. M. (2007). Children's questions: A mechanism for cognitive development. *Monographs of the Society for Research in Child Development,* 72 1: vii–ix, 1–112.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *CVPR*.

Denkowski, M., & Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16 × 16 words: Transformers for image recognition at scale. In *ICLR*.

Du, X., Shao, J., & Cardie. C. (2017). Learning to ask: Neural question generation for reading comprehension. In *ACL*.

Farhadi, A., Endres, I., Hoiem, D., & Forsyth, D. A. (2009). Describing objects by their attributes. In *CVPR*.

Gao, D., Wang, R., Shan, S., & Chen, X. (2019). Cric: A VQA dataset for compositional reasoning on vision and commonsense. arXiv preprint arXiv:1908.02962,

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., & Parikh, D. (2017). Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *CVPR*.

Gu, X., Lin, T.-Y., Kuo, W., & Cui. Y. (2021). Open-vocabulary object detection via vision and language knowledge distillation. In *International conference on learning representations*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*, 1735–1780.

Huynh, D., & Elhamifar, E. (2020). A shared multi-attention framework for multi-label zero-shot learning. In *CVPR*.

Hwang, J. D., Bhagavatula, C., Le Bras, R., Da, J., Sakaguchi, K., Bosselut, A., & Choi, Y. (2020). Comet-atomic: On symbolic and neural commonsense knowledge graphs. In *AAAI*.

Jayaraman, D., & Grauman, K. (2014). Zero-shot recognition with unreliable attributes. In *NIPS*.

Ji, S., Pan, S., Cambria, E., Marttinen, P., & Yu, P. S. (2022). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems, 33*, 494–514.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q. V., Sung, Y.-H., Li, Z., & Duerig, T. (2021). Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*.

Kim, W., Son, B., & Kim, I. (2021). ViLT: Vision-and-language transformer without convolution or region supervision. In *ICML*.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023) Segment anything. arXiv:2304.02643

Krishna, R., Bernstein, M., & Fei-Fei, L. (2019). Information maximizing visual question generation. In *CVPR*.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., Bernstein, M. S., & Fei-Fei, L. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision, 123*, 32–73.

Lampert, C. H., Nickisch, H., & Harmeling, S. (2009). Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.

Li, S., Wang, L., Wang, S., Kong, D., & Yin, B. (2021). Zero-shot recognition with image attributes generation using hierarchical coupled dictionary learning. *ACM Multimedia Asia*.

Li, Y., Duan, N., Zhou, B., Chu, X., Ouyang, W., & Wang, X. (2018). Visual question generation as dual task of visual question answering. In *CVPR*.

Liu, F., Xiang, T., Hospedales, T. M., Yang, W., & Sun, C. (2018). IVQA: Inverse visual question answering. In *CVPR*.

Marino, K., Rastegari, M., Farhadi, A., & Mottaghi, R. (2019). OK-VQA: A visual question answering benchmark requiring external knowledge. In *CVPR*.

Misra, I., Girshick, R., Fergus, R., Hebert, M., Gupta, A., & van der Maaten, L. (2018). Learning by asking questions. In *CVPR*.

Mostafazadeh, N., Misra, I., Devlin, J., Mitchell, M., He, X., & Vanderwende, L. (2016) . Generating natural questions about an image. In *ACL*.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *ACL*.

Qiao, R., Liu, L., Shen, C., & van den Hengel, A. (2016). Less is more: Zero-shot learning from online textual documents with noise suppression. In *CVPR*.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., & Krueger, G. (2021). Learning transferable visual models from natural language supervision. In *ICML*

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners.

Reed, S., Akata, Z., Lee, H., & Schiele, B. (2016). Learning deep representations of fine-grained visual descriptions. In *CVPR*.

Rohrbach, M., Stark, M., & Schiele, B. (2011). Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *CVPR*.

Ronfard, S., Zambrana, I. M., Hermansen, T. K., & Kelemen, D. (2018). Question-asking in childhood: A review of the literature and a framework for understanding its development. *Developmental Review*.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. In *NeurIPS EMC2 Workshop*.

Scialom, T., & Staiano, J. (2020). Ask to learn: A study on curiosity-driven question generation. In *COLING*.

Shah, S., Mishra, A., Yadati, N., & Talukdar, P. P. (2019). KVQA: Knowledge-aware visual question answering. In *AAAI*.

Shazeer, N., & Stern, M. (2018). Adafactor: Adaptive learning rates with sublinear memory cost. In *ICML*.

Shen, S., Li, C., Hu, X., Xie, Y., Yang, J., Zhang, P., Gan, Z., Wang, L., Yuan, L., Liu, C., & Keutzer, K. (2022). K-LITE: Learning transferable visual models with external knowledge. In *NeurIPS*.

Shen, T., Kar, A., & Fidler, S. (2019). Learning to caption images through a lifetime by asking questions. In *ICCV*.

Speer, R., Chin, J., & Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*.

Uehara, K., & Harada, T. (2022). K-vqg: Knowledge-aware visual question generation for common-sense acquisition. arXiv:2203.07890

Uehara, K., Tejero-De-Pablos, A., Ushiku, Y., & Harada, T. (2018). Visual question generation for class acquisition of unknown objects. In *ECCV*.

Uppal, S., Madan, A., Bhagat, S., Yu, Y., & Shah, R. R. (2021). C3VQG: Category consistent cyclic visual question generation. In *ACM Multimedia in Asia*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., & Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *NeurIPS*.

Vedantam, R., Lawrence Zitnick, C., & Parikh, D. (2015). CIDEr: Consensus-based image description evaluation. In *CVPR*.

Wang, P., Wu, Q., Shen, C., Dick, A., & Van Den Hengel, A. (2017). FVQA: Fact-based visual question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*(10), 2413–2427.

Wang, X., Ye, Y., & Gupta, A. (2018). Zero-shot recognition via semantic embeddings and knowledge graphs. In *CVPR*.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning, 8*(3), 229–256.

Yu, L., Poirson, P., Yang, S., Berg, A. C., & Berg, T. L. (2016). Modeling context in referring expressions. In *ECCV*.

Yuan, X., Wang, T., Gülçehre, Ç., Sordoni, A., Bachman, P., Zhang, S., Subramanian, S., & Trischler, A. (2017). Machine comprehension by text-to-text neural question generation. In *Rep4NLP@ACL*.

Zareian, A., Rosa, K. D., Hu, D. H., & Chang, S. F. (2021). Open-vocabulary object detection using captions. In *CVPR*.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.