Co-VQA : Answering by Interactive Sub Question Sequence

Ruonan Wang[†], **Yuxi Qian**[†], **Fangxiang Feng**[†], **Xiaojie Wang**[†*], **Huixing Jiang**[‡]
[†]Beijing University of Posts and Telecommunications
[‡]Meituan-Dianping Group
[†]{wangrn, qianyuxi, fxfeng, xjwang}@bupt.edu.cn
[‡]jianghuixing@meituan.com

Abstract

Most existing approaches to Visual Question Answering (VQA) answer questions directly, however, people usually decompose a complex question into a sequence of simple sub questions and finally obtain the answer to the original question after answering the sub question sequence(**SQS**). By simulating the process, this paper proposes a conversation-based VQA (**Co-VQA**) framework, which consists of three components: Questioner, Oracle, and Answerer. Questioner raises the sub questions using an extending HRED model, and Oracle answers them one-by-one. An **Adaptive Chain Visual Reasoning Model (ACVRM)** for Answerer is also proposed, where the question-answer pair is used to update the visual representation sequentially. To perform supervised learning for each model, we introduce a well-designed method to build a SQS for each question on VQA 2.0 and VQA-CP v2 datasets. Experimental results show that our method achieves state-of-the-art on VQA-CP v2. Further analyses show that SQSs help build direct semantic connections between questions and images, provide question-adaptive variable-length reasoning chains, and with explicit interpretability as well as error traceability.

1 Introduction

Visual Question Answering (Agrawal et al., 2015) requires to answer questions about images. It has to process visual and language information simultaneously, which is a basic ability of advanced agents. Therefore, it has attracted more and more attention (Anderson et al., 2018; Lu et al., 2016; Goyal et al., 2017b; Agrawal et al., 2018). The conventional approach (Agrawal et al., 2015) for Visual Question Answering (VQA) is to encode image and question separately and incorporate the representation of each modality into a joint representation. Recently, with the proposal of Transformer (Vaswani et al., 2017), based on previous dense co-attention models (Kim et al., 2018; Nguyen and Okatani, 2018), some methods (Yu et al., 2019; Gao et al., 2019) further adopt self-attention mechanism to exploit the fine-grained information in both visual and textual modalities. Meanwhile, to enrich indicative information about the image contained in the visual representation, some researchers (Cadène et al., 2019; Li et al., 2019) have explored different methods of relational reasoning to capture the relationship between objects.

Figure 1: An illustrative example. After a sequence of four sub questions and their answers {(q1,a1),(q2,a2),(q3,a3),(q4,a4)}, its easier to answer the original question.

Though above methods have achieved significantly improved performances on real datasets (Agrawal et al., 2015; Goyal et al., 2017b), there are still some issues unsolvable. Most existing approaches answer questions directly, however, it is often difficult, especially to answer complex questions. On the one hand, achieving holistic scene understanding in one round is pretty challenging. On the other hand, performing the whole Q&A process in one round lacks interpretability, and it is difficult to locate the errors when the model runs into wrong answers. To address the above difficulties, motivated by theory of mind (Leslie, 1987), as shown in Figure 1, we imagine an internal conversation for answering the original question, where a sub question sequence (**SQS**, which includes several simple sub questions, we use **SQ** to refer to sub

*Xiaojie Wang is the corresponding author.

Findings of the Association for Computational Linguistics: ACL 2022, pages 2396 - 2408

question later) is raised and answered one-by-one progressively. Finally, the answer to the original question is obtained by capturing joint information accumulated in the whole SQS. This way has several significant cognitive advantages: 1) SQSs with different numbers of sub questions will be automatically generated for different questions, resulting in question-adaptive variable-length reasoning chains, 2) a SQS gives a clear reasoning path, it therefore provides explicit interpretability and traceability of errors, 3) different questions are likely to contain the same SQs or SQSs, these common SQs even SQSs help improve the generalization ability of models, 4) SQs are usually more simple and directly related to the images, which help to strengthen the semantic connections between linguistic and visual information.

To achieve above advantages, we therefore propose a **Conversation-based VQA (Co-VQA)** framework which includes an internal conversation for VQA. It consists of three components: **Questioner**, **Oracle** and **Answerer**. As shown in Figure 1, once a question is raised, Questioner asks some SQs, and Oracle provides answers one-by-one. Their conversation brings a SQS and the corresponding answer sequence. When there is no more SQ to be generated, the internal conversation is finished and Answerer gives the final answer to the original question.

Questioner employs the hierarchical recurrent encoder-decoder architecture (Sordoni et al., 2015), and we adopt a representative VQA model (Anderson et al., 2018) as Oracle. For Answerer, we propose an **Adaptive Chain Visual Reasoning Model (ACVRM)** to accomplish an explicit progressive reasoning process based on SQS, where SQs are used to guide the update of visual features by an extended graph attention network (Velickovic et al., 2018) gradually. Meanwhile, the answers of SQs are utilized as additional supervision signals to guide the learning process. Further, to provide supervision information for the above three models during training, we propose a well-designed method to construct a SQS for each question which is based on linguistic rules and natural language processing technology. VQA-SQS and VQA-CP-SQS datasets are obtained after applying this method to VQA 2.0 (Goyal et al., 2017b) and VQA-CP v2 (Agrawal et al., 2018) datasets.

Our contributions can be concluded into threefold:

- We introduce a Conversation-based VQA (Co-VQA) framework, which consists of three components: Questioner, Oracle and Answerer. The frame is different from existing VQA methods in principle.
- An Adaptive Chain Visual Reasoning Model (ACVRM) for Answerer is proposed, where the question-answer pair is used to update visual representation sequentially.
- Co-VQA achieves the new state-of-the-art performance on the challanging VQA-CP v2 dataset. Moreover, SQSs help to build direct semantic connections between questions and images, they provide question-adaptive variable-length reasoning chains with explicit interpretability as well as error traceability.

2 Related Work

Visual Question Answering. The current dominant framework for VQA consists of an image encoder, a question encoder, multimodal fusion, and an answer predictor (Agrawal et al., 2015). To avoid the noises caused by global features, methods(Yang et al., 2016; Malinowski et al., 2018) introduce various image attention mechanisms into VQA. Instead of directly using visual features from CNN-based feature extractors, to improve the performance of model, BUTD(Anderson et al., 2018) adopts Faster R-CNN (Ren et al., 2015) to obtain candidate regional features while Pythia(Jiang et al., 2018) integrates the regional feature with grid-level features. Meanwhile, Lu et al. (2016); Nam et al. (2017) put more attention on learning better question representations. To merge information from different modalities sufficiently, MFB(Yu et al., 2017) and MUTAN(Ben-younes et al., 2017) explored higher-order fusion methods. Further, BAN(Kim et al., 2018) and DCN(Nguyen and Okatani, 2018) propose dense co-attention model which directly establish interaction between different modalities with word-level and regional features. Moreover, with the proposal of Transformer (Vaswani et al., 2017), MCAN (Yu et al., 2019) and DFAF (Gao et al., 2019) adopt self-attention mechanism to fully excavate the fine-grained information contained in text and image. Meanwhile, to fully cover the holistic scene in an image, MuREL (Cadène et al., 2019) and ReGAT (Li et al., 2019) explicitly incorporate relations between regions into the interaction process.

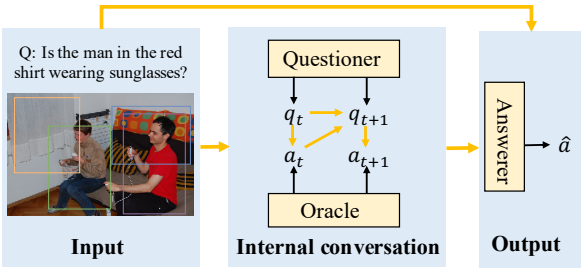

Figure 2: Overall illustration and data flow structure diagram of Co-VQA framework.

Selvaraju et al. (2020) also proposed sub questions but with very different motivation and methods. They found consistency issues in current VQA models which answer the reasoning questions correctly but fail on associated low-level perception questions. They therefore construct independent perception questions that serve as SQs to answer the reasoning questions, and proposed SQuINT to force a VQA model to attend to the same regions when answering the reasoning questions and their associated Perception SQ. The dataset proposed in this paper is different from them because our model needs a sequence of SQs to form a visual dialogue.

Visual Dialogue. Different from VQA, Visual dialogue (VD) is a continuous conversation for images. Several VD tasks (Visual Dialog (Das et al., 2017), GuessWhich (Chattopadhyay et al., 2017), GuessWhat?! (de Vries et al., 2017), MMD (Saha et al., 2018)) have been proposed. GuessWhat?!, as a goal-directed dialogue task, requires both players to continuously clarify the reference object through dialogue. The Oracle provides the Questioner with relevant information about the target object by constantly answering yes/no questions raised by the Questioner, and the Guesser generates the final answer based on the historical dialogue. Following the setting, our Co-VQA framework consists of three components, in which Questioner raises SQs, and Oracle answers them one-by-one, finally, Answerer obtains the answer to the original question.

3 Approach

Figure 2 shows the overall structure and data flows of Co-VQA, where the Questioner, the Oracle, and the Answerer are three major components. Given an input image I and a question Q, Co-VQA aims to predict the correct answer from the candidate answer set A^*. Specifically, the Questioner generates a new SQ q_t for the next round by combining the information in Q, I and the dialogue history

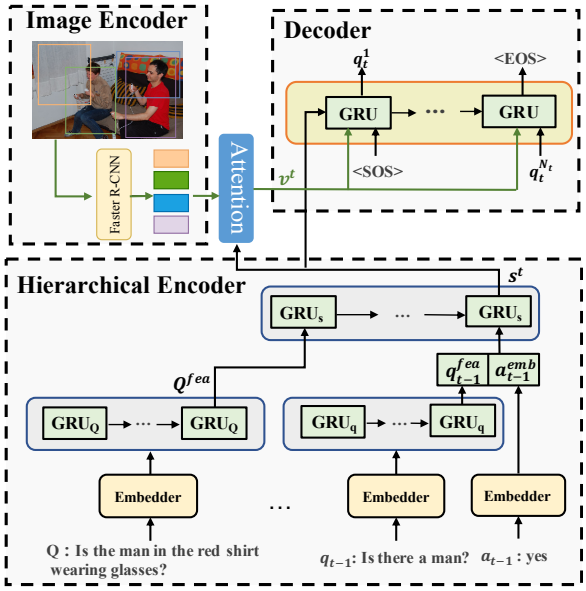

Figure 3: Overview of the **Questioner** model which is based on extending HRED model. There are three modules: Image Encoder, Hierarchical Encoder, Decoder.

$H_{t-1} = \{(q_1, a_1), \cdots, (q_{t-1}, a_{t-1})\}$. Then, Oracle produces appropriate answer a_t for q_t. After accomplishing the last round of sub question-answer pair, Answerer utilizes the historical information accumulated throughout the process to obtain the final answer. In this section, we will introduce the three components in Section 3.1-3.3.

3.1 Questioner

At round t, given an image I, a question Q and the dialogue history $H_{t-1} = \{(q_1, a_1), \cdots, (q_{t-1}, a_{t-1})\}$, Questioner aims to generate a new SQ q_t, which could be denoted as:

$$q_t \sim P_{\theta_Q}(q|Q, I, H_{t-1}), \quad (1)$$

where θ_Q denotes the parameters of Questioner. Generally, we build Questioner based on an extending hierarchical recurrent encoder decoder (HRED) architecture (Sordoni et al., 2015). The overall structure of Questioner is depicted in Figure 3.

Image Encoder. Following common practice(Anderson et al., 2018), we extract regional visual features from I in a bottom-up manner by using Faster R-CNN model(Ren et al., 2015). Each image will be encoded as a series of M regional visual features $R \in \mathbb{R}^{M \times 2048}$ with their bounding box $b = [x, y, w, h] \in \mathbb{R}^{M \times 4}$ ($M \in [10, 100]$ in our experiments).

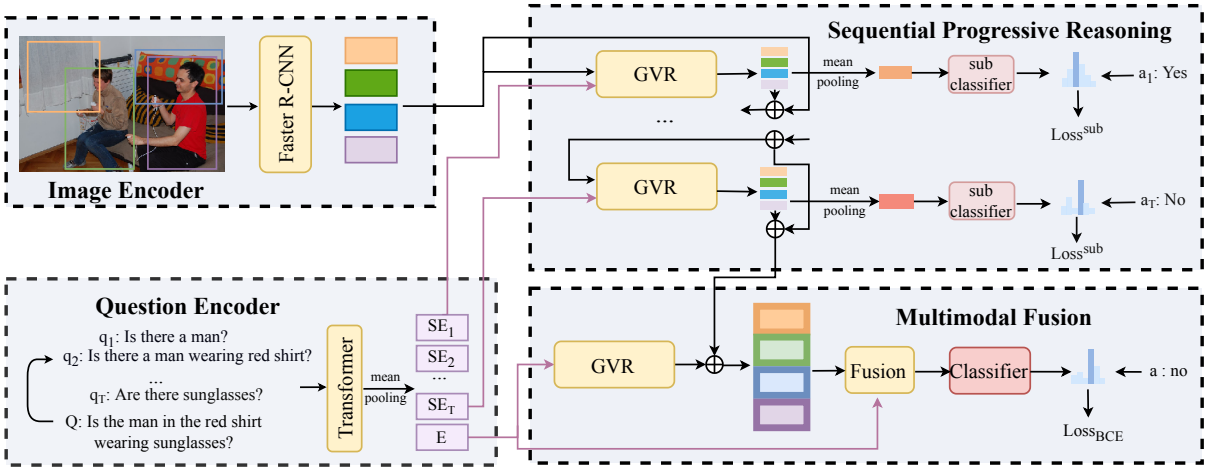

Figure 4: Model architecture of the proposed **ACVRM** for Answerer. There are four functional modules: Image Encoder, Question Encoder, Sequential Progressive Reasoning and Multimodal Fusion.

Hierarchical Encoder. Embedding matrix Embedder is adopted to map Q and each pair (q_i, a_i) in H_{t-1} to Q^{emb} and (q_i^{emb}, a_i^{emb}) respectively. Then, two question-level encoder GRU, GRU_Q and GRU_q, are deployed to obtain corresponding question feature Q^{fea} and q_i^{fea} for Q and q_i.

Q^{fea} is utilized as the first step input of session-level encoder GRU, GRU_s to grasp global information of original question. q_i^{fea} and a_i^{emb} are concatenated as qa_i^{fea}, which is regarded as representation for sub question-answer pair. Meanwhile, it is treated as the i+1-th step input of GRU_s to obtain context feature s_{i+1}:

$$s_{i+1} = GRU_s([q_i^{fea} \,||\, a_i^{emb}], s_i), \qquad (2)$$

where $||$ represents concatenation. After encoding H_{t-1}, we obtain current context representation s_t.

Decoder. At decoding q_t, we employ an extra one-layer GRU as decoder, which is initialized by s_t. Then a question-guided attention is deployed to regional features R to obtain the weighted visual feature v_t. Further, we fuse v_t with $Embedder(q_t^i)$ as the input of decoder at every time step i.

The negative log-likelihood loss is used for training, where T is the maximum round of dialogues:

$$L(\theta_Q) = -\sum_{t=1}^{T} logP(q_t|Q, I, H_{t-1}). \qquad (3)$$

3.2 Oracle

The Oracle aims to constantly answer SQs raised by Questioner. Specifically, at round t, Oracle supplies the answer a_t for SQ q_t, based on the image I and SQ q_t. We regard Oracle as a conventional VQA task and adopt the BUTD (Anderson et al., 2018), which is a representative VQA method, as our Oracle.

3.3 Answerer

Given a question Q, an image I and a complete dialogue history $H_T = \{q_1, a_1, ..., q_T, a_T\}$, the assignment of Answerer is to find out the most accurate $\hat{a}$ in the candidate answer set A^*, which could be denoted as:

$$\hat{a} = \underset{a \in A^*}{argmax}\, P_\theta(a|I, Q, H_T), \qquad (4)$$

where θ denotes the parameters of Answerer. To accomplish this task, we propose an **Adaptive Chain Visual Reasoning Model (ACVRM)**, which consists of four components: Image Encoder, Question Encoder, Sequential Progressive Reasoning, and Multimodal Fusion. The overall structure of ACVRM is illustrated as Figure 4.

3.3.1 Image and Question Encoder

Feature extraction modules are shown in the left part of Figure 4. Image encoder is the same as Questioner. For question encoder, we adopt a bidirectional Transformer (Vaswani et al., 2017). Q and each SQ in H_T will be padded to a maximum length and be encoded by bidirectional Transformer with random initialization, at last the corresponding question features $E \in \mathbb{R}^{d_q}, \{SE_i\}_{i=1}^{T} \in \mathbb{R}^{T \times d_q}$ are obtained after mean pooling. To align the feature dimensions, we linearly map image feature R to $V_0 \in \mathbb{R}^{M \times d_v}$. We set $d_q = d_v = 768$.

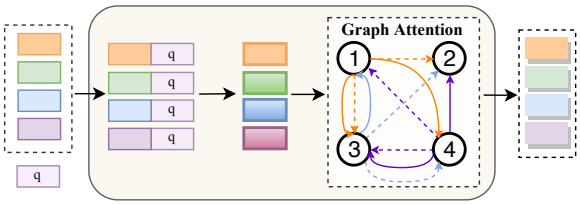

Figure 5: Flowchart of the **GVR**, including two parts: multimodal fusion based on concatenation and relation reasonsing based graph attention network.

3.3.2 Sequential Progressive Reasoning (SPR)

Overall. To realize progressive visual reasoning under the guidance of SQS, we utilize **Graph Visual Reasoning (GVR)** module, which will be introduced later, to gradually guide the update of visual features. Specifically, for Q containing T SQs, the t-th step of SPR can be expressed as:

$$V_t^R = GVR(V_{t-1}, SE_t; \theta_G), \tag{5}$$

where V_t^R represents the t-th step visual feature, and θ_G denotes parameters for GVR. Then, the residual connection is deployed in each round to preserve historical information and avoid vanishing gradients. Therefore, the updated visual feature for the t-th round can further be depicted as:

$$V_t = V_{t-1} + V_t^R. \tag{6}$$

Furthermore, each q_t has a corresponding answer a_t, which supplies an additional supervision signal for training. For each step t, we adopt a shared two-layer MLP as the sub classifier and then utilize average V_t^R as input. A cross-entropy loss is used for classification, which is denoted as $Loss_t^{sub}$.

Graph Visual Reasoning. Inspired by ReGAT (Li et al., 2019), we utilize an extended Graph Attention Network (Velickovic et al., 2018) to learn relations between objects. An overall illustration of GVR is shown in Figure 5. The whole reasoning process is abbreviated as $V^R = GVR(V, q)$, which consists of two parts: feature fusion and relational reasoning.

At first, the question representation q is concatenated with each of the M visual features v_i, which we write as $[v_i \,||\, q]$, then we compute a joint embedding as:

$$v_i^{'} = W([v_i \,||\, q]) \quad for \;\; i = 1, ..., M, \tag{7}$$

where $W \in \mathbb{R}^{d_q \times (d_q + d_v)}$, and $v_i^{'} \in \mathbb{R}^{d_q}$ is conducted as the initial value of node in the graph $G(V, E)$, where e_{ij} denotes edges between nodes. Then, to reduce the interference caused by irrelevant information, we design a masked multi-head attention for relational reasoning. Specially, for each head, inspired by Hu et al. (2018), the attention weight not only depends on visual-feature weight $\alpha_{ij}^{h,v}$, but also bounding-box weight $\alpha_{ij}^{h,b}$, we formulate non-normalized attention weight e_{ij} as:

$$e_{ij}^h = \alpha_{ij}^{h,v} + \log(\alpha_{ij}^{h,b}), \tag{8}$$

$$\alpha_{ij}^{h,v} = \frac{(W_q^h v_i^{'})^T \cdot W_k^h v_j^{'}}{\sqrt{d_h}}, \tag{9}$$

$$\alpha_{ij}^{h,b} = \max\{0, w \cdot f_b(b_i, b_j)\}, \tag{10}$$

where $d_h = \frac{d_q}{H}$, H denotes the number of head and we set $H = 8$, $W_q^h \in \mathbb{R}^{d_h \times d_q}$, $W_k^h \in \mathbb{R}^{d_h \times d_q}$, $f_b(\cdot, \cdot)$ first computes relative geometry feature $(log(\frac{|x_i - x_j|}{w_i}), log(\frac{|y_i - y_j|}{h_i}), log(\frac{w_j}{w_i}), log(\frac{h_j}{h_i}))$, then embeds it into a d_h-dimensional feature by computing cosine and sine functions of different wavelengths, $w \in \mathbb{R}^{d_h}$. Furthermore, according to e_{ij}^h, to learn a sparse neighbourhood N_i^h for each node i, we adopt a ranking strategy as $N_i^h = top_K(e_{ij}^h)$, where top_K returns the indices of the K largest values of an input vector, and we set K=15.

By employing above mechanism, output features of each head are concatenated, where $W_v^h \in \mathbb{R}^{d_h \times d_q}$:

$$v_i^R = ||_{h=1}^H \sigma(\sum_{j \in N_i^h} softmax(e_{ij}^h) \cdot W_v^h v_j^{'}). \tag{11}$$

3.3.3 Fusion Module

SQ-aware visual features V_T are obtained after completing the whole process of SPR. To sufficiently integrate the information of two modalities, we utilize Q to convert V_T into final context-aware $\tilde{V}$ through GVR:

$$\tilde{V} = GVR(V_T, E). \tag{12}$$

Then, we employ the same multi-modal fusion strategy as Anderson et al. (2018) to obtain a joint representation H. For Answer Predictor, we adopt a two-layer multi-layer perceptron (MLP) as classifier, with H as the input. Binary cross entropy is used as the loss function. Thus, final loss can be formulated as:

$$Loss = Loss_{BCE} + \sum_{t=1}^{T} Loss_t^{sub}. \tag{13}$$

	Validation				Test-std
Model	All	Y/N	Num	Other	All
Bottom-Up	63.37	80.4	43.02	55.96	65.67
BAN	66.04	-	-	-	-
MuREL	65.14	-	-	-	68.41
ReGAT∗	67.18	-	-	-	70.58†
DFAF	66.66	-	-	-	70.34†
MCAN	67.2(67.14±0.04‡)	84.82‡	49.24‡	58.44‡	70.9†
MLIN	66.53	-	-	-	70.28†
Ours	**67.26**±0.02	84.71	**50.38**	58.44	70.39

Table 1: Performance on VQA 2.0 validation split and test-standard splits. "∗" means ensembling result. "†" means training with augmented VQA samples from Visual Genome. "‡" based on our re-implementations.

4 Experiments

4.1 Datasets

We evaluate our approach on two widely used datasets:

VQA 2.0 (Goyal et al., 2017b) is composed of real images from MSCOCO (Lin et al., 2014) with the same train/validation/test splits. For each image, an average of 3 questions are generated. These questions are divided into 3 categories: Y/N, Number, and Other. 10 answers are collected for each image-question pair from human annotators. The model is trained on the train set, but when testing on the test set, both train and validation set are used for training, and the max-probable answer is selected as the predicted answer.

VQA-CP v2 (Agrawal et al., 2018) is a derivation of VQA 2.0. In particular, the distribution of answers concerning to question types is designed to be different between train and test splits, which is aimed at overcoming language priors.

Construction of SQS dataset. To provide the corresponding supervised signal for training Questioner, Oracle, and Answerer, we propose a well-designed method, which is chiefly based on linguistic rules and natural language processing technology. **VQA-SQS** and **VQA-CP-SQS** are obtained by applying this method on VQA 2.0 and VQA-CP v2 datasets. The details of the construction process and the specific statistical information of the two datasets can be found in Appendix.

4.2 Implementation Details

Training and inference. During training, Questioner, Oracle, and Answerer are trained independently. For inference, given a question Q and an image I, SQS is firstly generated through the cooperation between Questioner and Oracle, then Q, I and the complete SQS is combined as the input of Answerer, and obtain the final answer.

Parameters. Each question is tokenized and padded with 0 to a maximum length of 14. For Questioner and Oracle, each word is embedded using 300-dimensional word embeddings. The dimension of the hidden layer in GRU is set as 1,024(except for GRU_Q and GRU_s with 1,324).

Our model is implemented based on PyTorch(Paszke et al., 2017). In experiments, we use Adamax optimizer for training, with the mini-batch size as 256. For choice of the learning rate, we employ the warm-up strategy(Goyal et al., 2017a). Specifically, we begin with a learning rate of 5e-4, linearly increasing it at each epoch till it reaches 2e-3 at epoch 4. After 14 epochs, the learning rate is decreased by 0.2 for every 2 epochs up to 18 epochs. We also adopt an early stopping strategy. For the transformer encoder, we fix the learning rate as 5e-5. Every linear mapping is regularized by weight normalization and dropout (p = 0.2 except for the classifier with 0.5).

4.3 Results

To compare with existing VQA methods, we conduct several experiments to evaluate the performance of our Co-VQA framework, further, to verify the generation quality of the SQs and their impact on the performance of the overall model, Questioner and Oracle are tested additionally.

In Table 1, we compare our method with previous work on VQA 2.0 validation and test-standard split. From Table 1, it can be seen that on validation split, Co-VQA achieves the top-tier performance, an accuracy of 67.26, which surpasses that of MCAN(Yu et al., 2019) by 0.06. Although the absolute improvement is slight, we report the standard deviation in Table 1, compared with MCAN,

Model	All	Y/N	Num	Other
MuREL	39.54	42.85	13.17	45.04
ReGAT∗	40.42	-	-	-
MCAN‡	42.35	42.29	14.51	**50.02**
Ours	**42.52**	**44.42**	**14.68**	49.17

Table 2: State-of-the-art comparison on the VQA-CP v2 dataset. "∗" means ensembling result. "‡" Results based on our re-implementations.

BLEU-1	**BLEU-2**	**BLEU-3**
67.8	38.4	21.2

Table 3: BLEU evaluation scores of Questioner. We don't report BLEU-4 score because the length of some sub questions is shorter than 4.

the p-value is 0.006, so the improvements are statistically significant with $p < 0.05$. Moreover, we achieve an obvious performance improvement on the number questions. On VQA 2.0 test-standard split, without additional augmented samples from Visual Genome (Krishna et al., 2017), our performance is still the third place. We assume the gap between the two splits is mainly due to the difference in SQS generation quality.

To demonstrate the generalizability of Co-VQA, we also conduct experiments on the VQA-CP v2 dataset, where the distributions of the train and test splits are quite different. Table 2 illustrates the overall performance, and our model gains a significant advantage (+2.1) over ReGAT. Compared with MCAN, our model also improved by 0.16.

For Questioner and Oracle, we train and evaluate the train/validation split of the VQA-SQS dataset.

Oracle. The accuracy of Oracle is 93.73 and the average F-value is 90.13. On the one hand, the high accuracy is due to SQ itself being simple; On the other hand, decomposition of question leads to many same SQs, strengthening image-language correlation ability at SQ level.

Questioner. For Questioner, the BLEU score is adopted to measure the quality of the generated SQs. As is shown in Table 3, we attribute the low BLEU scores to the diversity of syntax details.

4.4 Ablation Study

We conduct several ablation studies to explore critical factors affecting the performance of Co-VQA.

Model	All	Y/N	Num	Other
Full	**67.26**	**84.71**	**50.38**	**58.44**
wo-sub-loss	66.94	84.58	48.95	58.29
wo-SQS	66.55	84.43	46.78	58.18

Table 4: Ablation studies on impact of SQS on VQA 2.0 validation set.

Model	SQS-0 (57,411)	SQS-1 (119,285)	SQS-2 (34,226)	SQS-3&4 (3,432)	All (214,254)
Full	**69.51**	**66.70**	**65.78**	63.62	**67.26**
wo-SQS	69.48	65.68	64.85	**64.35**	66.55

Table 5: Ablation studies of SQS in detail on VQA 2.0 validation set. SQS-n represents the subset of samples with n SQs in VQA-SQS validation set. We report the average accuracy on each subset.

The impact of SQS. In general, as we can observe from Table 4, though there are noises in the answers for SQs, the weak supervision signal provided by them shows a gain of +0.32. Furthermore, the decrease is obvious(-0.71) when we remove total SQS from the model, indicating that though the SQS generated from Questioner is not good enough, it still plays an important role in improving the performance of the model.

Detail Analysis of SQS. To analyze the impact of SQS in detail, we divide the validation split of VQA-SQS into SQS-0 / SQS-1 / SQS-2 / SQS-3&4 subsets, where SQS-n represents samples with n SQs. Then, the average accuracy of different models on each subset is reported in Table 5. For SQS-1 and SQS-2, the additional reasoning brought by SQS achieves an improvement of 1.02 and 0.93 respectively. However, for SQS-3&4, the performance decreases compared with wo-SQS.

We perform statistics in two aspects to comprehensively explore the causes of this phenomenon. As shown in Table 6, compared with other subsets, SQS-3&4 has fewer samples, causing insufficient learning for these samples of a long sequence. Moreover, SQs in SQS-3&4 occur less frequently,

Subset	SQS-0	SQS-1	SQS-2	SQS-3&4	All
Samples-Num	57,411	119,285	34,226	3,432	214,254
Avg(Freq of SQ)	-	870	851	693	854

Table 6: Data statistics of SQS in detail on VQA 2.0 validation set. The first row shows the number of original questions contained in different SQS sets, and the second row counts the average number of occurrences of the sub questions contained in each subset in the VQA-SQS train split.

Figure 6: Visualization of attention maps learned by complete Co-VQA with those learned by wo-SQS. The second and last column corresponds to the prediction of wo-SQS and complete Co-VQA respectively. Red and blue bounding boxes shown in each image are the top-2 attended regions.

Model	All	Y/N	Num	Other
Full	**67.26**	**84.71**	**50.38**	**58.44**
shuffle	67.15	84.68	49.78	58.42
random	67.08	84.68	49.75	58.28

Table 7: Ablation studies of coherence of SQS on VQA 2.0 validation set.

thus it is inadequate for the model to establish accurate semantic connections between these images and questions.

Coherence of SQS. We also study the impact of the coherence of SQS on performance. We run two different cases: 1) randomly shuffle the SQs in a sequence; 2) remove some SQs in a sequence with 50% probability. As we can observe from Table 7, the declines from the original one are not significant, partly due to the fact that the coherence of SQS in the current dataset VQA-SQS is not good enough.

4.5 Visualization

To better illustrate the effectiveness, explicit interpretability, and traceability of errors of Co-VQA, we visualize and compare the attention maps learned by complete Co-VQA with those learned by model wo-SQS. As shown in Figure 6. Column 1 is the original question and ground truth, while Column 2 corresponds to the prediction of model wo-SQS. The middle columns and last column correspond to the generated sub q&a, and the prediction of Co-VQA, respectively. To visualize the attention maps, we use the in-degree of each node as the attention value and circle the top-2 attended regions with red and blue boxes.

Line 1 shows model wo-SQS only notices one of the dogs and gives a wrong answer "1". However, through SQ "Are there dogs?", Co-VQA focuses on two dogs and gives the correct answer "2". This case demonstrates that asking an existence question firstly is beneficial to number questions. In Line 2, model wo-SQS focuses on unrelated entities. How-

ever, Co-VQA attends to the women and the people wearing short sleeves gradually with SQS, and finally, concentrates on the related woman's shirt. Line 3 shows Co-VQA successively attends to cars, blue cars, and the license plate under the guidance of SQS and gets the correct answer. These examples prove that questions with different complexity will correspond to SQS of variable length, and SQ is indeed related to more accurate image attention. Moreover, generating SQ provides not only the logic of reasoning but also additional language interpretation. Thus, compared with previous works that only explain models by attention maps, Co-VQA has significantly better interpretability.

The last line shows Co-VQA gives a wrong answer after adding SQS. However, we can find some possible causes, such as the wrong answer of Q1, Q2 is not related to the question, and the model doesn't attend to relevant entities in the light of Q1. It shows that Oracle and Questioner may give wrong answers or generate inappropriate questions, as well as Answerer may establish faulty semantic connections between questions and images, which verifies that Co-VQA has sure traceability for errors and provides guidance for future work.

5 Conclusions

We propose a Conversation-based VQA (Co-VQA) framework which consists of Questioner, Oracle, and Answerer. Through internal conversation based on SQS, our model not only has explicit interpretability and traceability of answer errors but also can carry out question-adaptive variable-length reasoning chains. Currently, Questioner is relatively simple, and the quality still has a lot of room to improve. Meanwhile, current SQs are only yes/no questions. For future work, we plan to explore how to more effectively generate more diverse and higher quality SQS, and look forward to better model performance.

Acknowledgement

The work was partially supported by the National Natural Science Foundation of China (NSFC62076032) and the Cooperation Project with Beijing SanKuai Technology Co., Ltd. We would like to thank anonymous reviewers for their suggestions and comments, thank Duo Zhen for his suggestions about several iterations of this paper, thank our colleagues for their contributions in recorrecting the dataset.

References

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4971–4980.

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. 2015. Vqa: Visual question answering. *International Journal of Computer Vision*, 123:4–31.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6077–6086.

Hedi Ben-younes, Rémi Cadène, Matthieu Cord, and Nicolas Thome. 2017. Mutan: Multimodal tucker fusion for visual question answering. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2631–2639.

Rémi Cadène, Hedi Ben-younes, Matthieu Cord, and Nicolas Thome. 2019. Murel: Multimodal relational reasoning for visual question answering. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1989–1998.

Prithvijit Chattopadhyay, Deshraj Yadav, Viraj Prabhu, Arjun Chandrasekaran, Abhishek Das, Stefan Lee, Dhruv Batra, and Devi Parikh. 2017. Evaluating visual conversational agents via cooperative human-ai games. In *HCOMP*.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M. F. Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1080–1089.

Harm de Vries, Florian Strub, A. P. Sarath Chandar, Olivier Pietquin, H. Larochelle, and Aaron C. Courville. 2017. Guesswhat?! visual object discovery through multi-modal dialogue. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4466–4475.

Peng Gao, Hongsheng Li, Haoxuan You, Zhengkai Jiang, Pan Lu, Steven C. H. Hoi, and Xiaogang Wang. 2019. Dynamic fusion with intra- and inter-modality attention flow for visual question answering. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6632–6641.

Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 2017a. Accurate, large minibatch sgd: Training imagenet in 1 hour. *ArXiv*, abs/1706.02677.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017b. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6325–6334.

Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. 2018. Relation networks for object detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3588–3597.

Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. 2018. Pythia v0.1: the winning entry to the vqa challenge 2018. *ArXiv*, abs/1807.09956.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. In *NeurIPS*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.

Alan M. Leslie. 1987. Pretense and representation: The origins of "theory of mind.". *Psychological Review*, 94:412–426.

Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Relation-aware graph attention network for visual question answering. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10312–10321.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *CoRR*, cs.CL/0205028.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *NIPS*.

Mateusz Malinowski, Carl Doersch, Adam Santoro, and Peter W. Battaglia. 2018. Learning visual question answering by bootstrapping hard attention. In *ECCV*.

Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. 2017. Dual attention networks for multimodal reasoning and matching. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2156–2164.

Duy-Kien Nguyen and Takayuki Okatani. 2018. Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6087–6096.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zach DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149.

Amrita Saha, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. Towards building large scale multimodal domain-aware conversation systems. In *AAAI*.

Ramprasaath R. Selvaraju, Purva Tendulkar, Devi Parikh, Eric Horvitz, Marco Túlio Ribeiro, Besmira Nushi, and Ece Kamar. 2020. Squinting at vqa models: Introspecting vqa models with sub-questions. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10000–10008.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jianyun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio', and Yoshua Bengio. 2018. Graph attention networks. *ArXiv*, abs/1710.10903.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21–29.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6274–6283.

Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. 2017. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1839–1848.

A Appendix

Here we introduce our method for constructing SQS and the statistical information of datasets.

A.1 Data source

We construct our SQS dataset based on VQA 2.0 and VQA-CP v2 datasets.

A.2 Construction principle

To accomplish the process of SQS construction, we first determine the order of questions according to the templates in Table 8. For order-0 and order-1 questions, there is no corresponding SQ, order-2 questions can construct the corresponding order-1 SQs, while the order-3 questions can construct multiple order-1 SQs and order-2 SQs. Then, the principle of dataset construction is: high-order questions can adopt corresponding low-order questions as their sub questions, for each high-order question, these sub questions are arranged according to the order from low to high to form a sub question sequence.

A.3 Construction method

The details of the construction method can be illustrated as following:

1) For each question, we first adopt Spacy[1] and NLTK toolkit (Loper and Bird, 2002) to identify all noun blocks in the question and filter out some noun blocks based on the predefined phrase list. The phrase list mainly includes meaningless quantifiers, pronouns, and abstract nouns, such as lots, someone, something, you, they, it, the day, the picture, a body, emotion, this, type, etc.

2) After finishing the filter process, for questions that still contain noun blocks, according to the dependency relation between the extracted noun blocks, part of these noun blocks may be used as prepositional phrases. For other remaining noun blocks, we use Part-of-Speech Tagging of Spacy to classify them into corresponding nouns, adjectives, quantifiers, and prepositional phrases. For nouns, we save them separately, while for adjectives, quantifiers, and prepositional phrases, we save these modifiers together with the noun blocks in a form of 2-tuple (noun, modifier), such as (flower, red).

3) After step 1, for questions without noun blocks, considering there may be omissions in the process of extraction, we perform pattern matching through Spacy based on the pre-defined matching template to determine the category of these questions. Table 9 illustrates partial matching patterns for different type of questions. Especially, for existence questions, no additional processing is required, while for other types of questions, we save the nouns that are existing in the questions.

4) We further filter the nouns and tuples saved in 2) and 3). This conduction aims to filter out abstract nouns, non-substantial nouns, and 2-tuple corresponding to these nouns. The following are some cases to be filtered:

a) **Abstract Noun**: direction, design, surface, area, emotion, skill etc.

b) **Non Substantive Noun**: mode, base, day, love, name, print, piece etc.

5) For the remaining nouns and their corresponding 2-tuple, we use the pre-defined question template to construct the corresponding sub questions. To facilitate the process of construction, we design all sub questions as yes / no questions. The matching pattern for each type of sub question are revealed in Table 11.

6) The construction process of ground-truth answers for sub questions can be illustrated as follows:

Existence SQ and Attribute SQ we first extract the label and attribute information of the entity by using the detection model and then combine this information to produce the answer.

Prep SQ and Position SQ the location information obtained by the detection model is utilized to judge the relationship of overlapping and orientation between entities, we use the obtained relationship to generate the corresponding answer.

Number SQ we first make a rough quantity estimation based on the image, and then make a manual correction.

6) Considering there may be wrong answers, incoherent sequences, and nonstandard question grammar in the process of automatic construction. So, to increase the diversity of SQs, we invite ten students in our laboratory to further manually recorrect some samples(about 5K samples).

order	question template
0	no entity
1	single entity
2	entity & attribute
3	comparison between different entities

Table 8: Templates for question of different order.

[1] https://spacy.io/

Question Type	Matching Pattern
Existence	(do you see)?[DET | PRON | ADP]* [NOUN | PROPN]* NOUN?
Verb	(do you see)? [DET | PRON | ADP]* [NOUN | PROPN]* NOUN? [VBG | VBN]?
Attribute	BE [DET | PRON | ADP]* [NOUN | PROPN]* NOUN? ADJ?
Num	BE [DET | PRON | ADP]* NUM NOUN NOUN* ?
Prep	BE [DET | PRON | ADP]* [NOUN | PROPN]* NOUN VERB? ADP DET NOUN NOUN* ?

Table 9: Matching patterns for different type of questions

Dataset	Split	#Images	#Q&A	#Non-empty SQS	Avg(#SQ)
VQA-SQS	Train	82,783	443,757	328,140	0.94
VQA-SQS	Val	40,504	214,354	156,943	0.925
VQA-CP-SQS	Train	120,932	438,183	322,200	0.93
VQA-CP-SQS	Test	98,226	219,928	162,883	0.946

Table 10: Dataset statistics of VQA-SQS and VQA-CP-SQS.

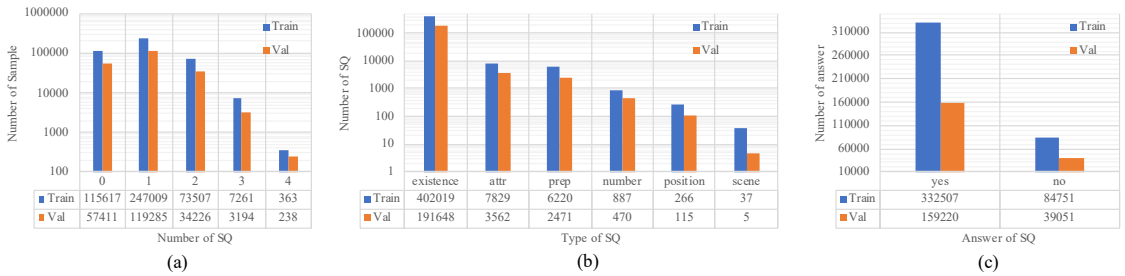

Figure 7: Dataset distribution of VQA-SQS.

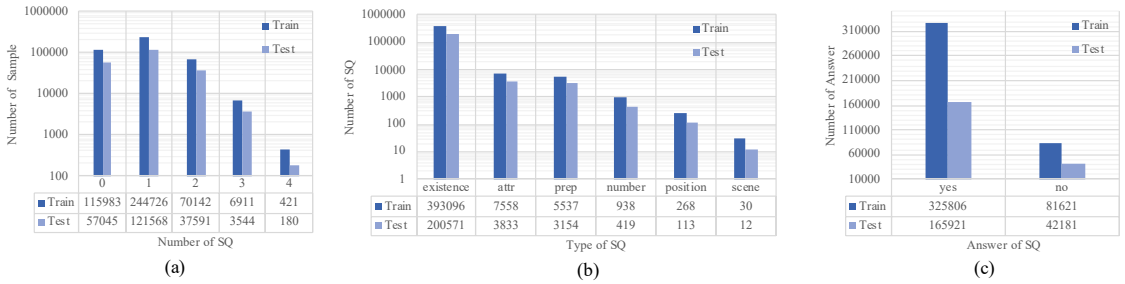

Figure 8: Dataset distribution of VQA-CP-SQS.

Figure 9: Some samples of VQA-SQS, including existence SQ, attribute SQ, prep SQ and number SQ.

SQ Type	Matching Pattern
Existence	Is there any [entity]? Is there any [color] [entity]? Are there [entites]?
Attribute	Is the [entity] [color]? Is any [entity]? Are these [entites] in similar size?
Prep	Is there any [entity] on the [entity2]? Is there any [entity] in the [entity2]?
Number	Are there [number] [entites]? Is there only one [entity]?
Position	Is the [entity] on the left? Is the [entity] on the right? Is the [entity] in the middle?

Table 11: Sub question generation template for different SQ types.

The SQS datasets obtained by performing the above operations on VQA 2.0 and VQA-CP v2 datasets are called VQA-SQS and VQA-CP-SQS respectively.

A.4 Dataset statistics

Table 10 shows general statistical information of the two SQS datasets, then, Figure 7 and Figure 8 respectively reveal three fine-grained distributions of two datasets including number distribution of SQ (7-a / 8-a), type distribution of SQ (7-b / 8-b) and answer distribution of SQ (7-c / 8-c). To display more convenient, in (7-a / 8-a) and (7-b / 8-b), the ordinate axis adopts logarithmic scale.

Figure 9 displays four samples with three sub questions in the VQA-SQS dataset.