Research Article

JGRCAN: A Visual Question Answering Co-Attention Network via Joint Grid-Region Features

Jianpeng Liang, Tianjiao Xu, Shihong Chen, and Zhuopan Ao

1School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou 510000, China
2Natural Language Processing & Portuguese-Chinese Machine Translation Laboratory (NLP2CT), Faculty of Science and Technology, University of Macau, Macau 999078, China
3Guangzhou Key Laboratory of Multilingual Intelligent Processing, School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou 510000, China
4School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou 510000, China

Correspondence should be addressed to Shihong Chen; ibm255chen@163.com

Received 24 March 2022; Revised 23 August 2022; Accepted 6 September 2022; Published 15 October 2022

1. Introduction

Visual question answering (VQA) [1] aims to give an image and some corresponding questions and automatically answers the questions related to the image. The task involves two modes of image and text, combining the research of computer vision and natural language processing. How to effectively learn the important features between images and text is a difficult task. At present, significant progress has been made in many image and text multimodal tasks, such as image caption, image-text matching, and visual question answering.

With the popularity of VQA, more and more researchers have proposed related work to solve VQA problems. These research works [2, 3] attempt to understand images and questions in fine-grained scenes. Many existing methods solve the problem by obtaining the information from key image regions related to the question and applying the image and question attention mechanism to solve this kind of problem. Other methods [4, 5] also point out that relevant keywords are of great significance, and better effects can be achieved by using rich visual content and important keywords in the image. However, these co-attention models learn rough interactions between multimodal instances, and the learned co-attention cannot infer the correlation between each image region and each question word. This leads to significant limitations in these co-attention models.
In addition, many researchers focused on the visual feature extractor, and they found that a good feature extractor has a great impact on the performance of the VQA-V2 task. Peter et al. [3] proposed a bottom-up attention mechanism and used faster-RCNN [6] to extract region features of targets in images. The results showed that effective feature extraction greatly improved the performance of the VQA task, so many researchers used faster-RCNN as the visual feature extraction model. However, region features also have some disadvantages, which will lead to the loss of some important visual information, such as the shape, spatial relationship, and overlap of objects in the image. Therefore, only detecting targets cannot well represent the features we need. The grid features proposed by Jiang et al. [7] in 2020 also have good performance, and the inference speed is twice as fast as the regional features. The grid features extracted from the pretraining model at the same layer can be similar to the region features, and even better results can be achieved if the training parameters are adjusted. However, the grid features do not perform well in some samples, with strong dependence on the target region. Lu et al. [8] proposed the idea of fusing grid features and regional features to improve VQA performance. Compared with the baseline model [2], the experiment has improved. However, the ordinary attention mechanism of this method is not very effective, and it cannot integrate the features of natural language and images well. To solve the above problems, we propose a visual question-answering co-attention network via joint grid-region features (JGRCAN) to realize the complementary advantages of region features and grid features. The main contributions of this paper are as follows:

1. We solve VQA problems by combining region features with grid features, rather than traditional single visual features. The region feature extracts the bounding boxes related to the target from an image, and the grid feature evenly pools an image into multiple grids of the same size. The combination of the two features can effectively improve the overall performance of the VQA task.

2. We also propose a co-attention layer based on a multi-head attention mechanism, which integrates question, grid, and region features, dynamically determines the attention weight distribution of each feature, and generates the final joint feature representation to predict the correct answer. The experimental results of the JGRCAN model based on the above improvements on the VQA-V2 dataset are better than many baseline models.

2. Related Works

In this part, we briefly review the previous research on visual question answering, including feature extraction, feature fusion, and attention algorithms. Among them, feature extraction is to extract their feature representation from a given question and image. A good feature extractor also plays a great role in the visual question answering tasks, especially the selection of image features. Multi-modal feature fusion [9, 10] refers to the input of multiple feature vectors of different modes and the output of the fused vector representation. The modes can be text, image, voice, and video. The function of the attention algorithm is to select the important areas related to the question in the image according to the given question or to pay attention to the keywords in the question according to the image, and discard those irrelevant content or noises. The VQA task involves two different modes and requires the fusion of the visual features extracted from the image and the text features extracted from the question.

2.1. Feature Extraction. For text feature extraction, the vast majority of VQA methods use LSTM [11] or GRU and RNN [12] as the text feature extractor to encode the question words and obtain the text feature embeddings of the question. In the early VQA methods, the VGG [13] network is usually used to extract visual features. With the ResNet proposed by Kaiming et al. [14], researchers gradually turn to the ResNet, whose visual feature extraction performance is superior to VGG. At present, most existing VQA methods adopt the feature extraction method combining bottom-up and top-down attention proposed by Anderson et al. [3] and use faster-RCNN [6] to extract region features from images. However, the disadvantage of this method is that each region is represented by a single feature vector. This inevitably loses a lot of object details, such as the color of the sky. Jiang et al. [7] reviewed the grid features in VQA and found that compared with region features, grid VQA features covered all the content of a given image in a more fragmentary form, and the inference speed was twice as that of region features. However, the accuracy is not high for the question of the number type, especially in the case of many small objects in the image. To sum up, grid features and region features have their own advantages and disadvantages. Based on the above situation, Lu et al. proposed that [8] to improve the advantages and disadvantages mentioned above, they adopted the common attention mechanism and simultaneously learned the grid-based image region and detection-based image region related to input questions in the image.

2.2. Attention Mechanism. Many researchers have introduced the attention mechanism to focus on the most relevant image regions and question words and learn the features of the images in the key regions adaptively under the guidance of a given question to obtain better visual and text performance. Yang et al. [15] proposed a stacked attention network, which can learn image region attention through multiple iterations. Lu et al. [4] proposed a hierarchical co-attention approach in which the network uses the co-attention layer to learn key areas in the image and the keywords in question. To overcome the deficiency of multimodal interaction based on co-attention, the bilinear attention network (BAN) [16] and dense symmetrical co-attention network (DCN) [17] proposed a dense interaction model between arbitrary image regions and arbitrary question words. The dense common attention mechanism
helps to understand the image-question relationship and thus answer questions correctly. Yang et al. [18] proposed using the types of problems to classify them and carry out a common attention mechanism. In 2019, Peng et al. [19] used two general attention units, SA (self-attention) and GA (guide-attention) to form a modular common attention structure through the combination of SA and GA. In 2020, Guo et al. [20] proposed a visual question answering method based on fusion grid and region features, named joint grid-region features co-attention network (JGRCAN). In 2020, through the module of ResNet and AvgPool, each of the n average segmented grid boxes of equal size in the image is obtained, namely grid feature Pg. Finally, the bounding boxes related to the target are extracted from the given image through faster-RCNN, that is, the region feature Pr. The three extracted features are then presented to the co-attention layer.

### 2.3. Feature Fusion

At present, feature fusion methods include the linear fusion method and the bilinear pool method. Methods based on linear fusion include linear operations such as addition or multiplication of feature elements and feature splicing. The method based on bilinear pooling is to pool the bilinear fusion features, which are generally expressed as the cross-product of two vectors. However, since the dimension of the feature vector obtained by the common cross product is the square of the original dimension of the feature vector, the subsequent classification model becomes larger and slows down the calculation speed of the model. Therefore, to alleviate the problem of the high dimension of features caused by bilinear pooling, Kim et al. [21] proposed a low-rank approximation algorithm for bilinear pooling, which is easy to operate and very effective. Yu et al. [22] proposed the multimodal factorized bilinear (MFB) pooling with co-attention learning and the multimodal factorized high-order (MFH) pool [23]. Firstly, different modal features are extended to the high-dimensional space and fused with the point product method, and then the pooling layer and normalization layer are entered to squeeze the high-dimensional features into compact output features. Reference [24] extended the self-attentional model of a single mode to a unified attentional model, which can describe complex intra-modal and inter-modal interactions of multimodal data, thus achieving a good effect.

### 3. Approach

In this part, we propose a visual question-answering network based on fusion grid and region features, named joint grid-region features co-attention network (JGRCAN). The structure is shown in Figure 1. JGRCAN consists of a feature extraction layer, a co-attention layer, and a feature fusion layer. The task of the feature extraction layer is to extract the text feature E of the question, the grid feature Pg, and the region feature Pr of the image from the given question and image, respectively. The role of the co-attention layer is to focus on the important features of grid and region features, respectively, through the guidance of the question features after self-attention and remove the interfering information and irrelevant factors. Finally, the feature fusion layer effectively fuses the question features after self-attention, network features after oriented attention, and region features to generate a joint vector representation, which is finally sent to the full connection layer for classification. This method effectively combines two different feature representations of grid features and region features to improve network performance.

### 3.1. Feature Extraction Layer

Given an image and a question, we first use Bi-LSTM to encode the question text and get the text feature $E$ after the encoding. Then, from the given image, through the module of ResNet and AvgPool, each of the n average segmented grid boxes of equal size in the image is obtained, namely grid feature $Pg$. Finally, the bounding boxes related to the target are extracted from the given image through faster-RCNN, that is, the region feature $Pr$. The three extracted features are then presented to the co-attention layer.

#### 3.1.1. Question Features

The long and short-term memory unit (LSTM) can capture long-distance dependencies in sentences, and the special structure of memory gates and forgetting gates can make the model remember important information and forget unimportant information when extracting features. Bi-LSTM can capture the bidirectional semantic dependence of sentences, achieve more fine-grained feature extraction, and effectively solve problems such as gradient disappearance or explosion in neural networks. Here, we mainly use the Bi-LSTM model to extract question features. After inputting a question, the sentence is segmented and trimmed to a maximum of 14 words, which is converted into a vector representation by word embedding to obtain the question representation $W_t = \{w_1, w_2, \ldots, w_{14}\}$, and then the initial feature vector $W$ is input into Bi-LSTM for further feature extraction. The following takes the calculation of the $t$th word vector $W_t$ as an example. The calculation methods of forgotten information $F_t$, memory information $M_t$, and current word state information $C_t$ are shown in the following:

\[
F_t = \sigma(W_f \cdot [h_{t-1}, w_t] + b_f), \tag{1}
\]

\[
M_t = \sigma(W_t \cdot [h_{t-1}, w_t] + b_t), \tag{2}
\]

\[
C_t = F_n \cdot C_{t-1} + M_t \cdot \tanh(W_c \cdot [h_{t-1}, w_t] + b_c). \tag{3}
\]

In formulas (1)–(3), $h_{t-1}$ represents the hidden layer information transmitted by the previous word; $C_t−1$ is the state information of the previous word; and $\sigma$ is the activation function. Finally, the feature vector $o_t$ of the $t$th word and the hidden layer information $h_t$ transmitted to the next word are calculated in the following way:

\[
o_t = \sigma(W_e \cdot [h_{t-1}, w_t] + b_e), \tag{4}
\]

\[
h_t = o_t \cdot \tanh(C_t). \tag{5}
\]

Single layer LSTM output characteristics $O = \{o_1, o_2, \ldots, o_{14}\}$, while inverse LSTM output
After the two are spliced together, the output final question feature is \( E = \{ e_1, \ldots, e_t, \ldots, e_{14} \} \in (14, d) \), where \( D \) is the hidden layer size of LSTM. In all of these formulas; \( W^* \) represents the weight matrix; \( B^* \) represents the bias matrix; and \( \sigma \) and tanh are the activation functions.

3.1.2. **Grid Features.** We extracted grid features based on the residual block idea of residual network (ResNet), which can well solve the problem of gradient explosion disappearance caused by deep convolutional network [14]. ResNet has different network layers, the more commonly used are ResNet50, ResNet101, and ResNet152. We use ResNet101 to extract features from the image. AvgPool and Reshape layers are added to ResNet101 to make ResNet’s originally generated feature map capable of being converted to grid features as one of the graphical features for the VQA-V2 task. The structure of the grid feature extraction method is shown in Figure 2.

With the traditional feature extraction based on convolution neural networks (CNN), with deeper layers of the network, a series of problems arise, such as typical information loss, gradient explosion, and gradient disappear. This kind of problem has been mitigated to a large extent through methods such as standard initialization and intermediate standardization layers. However, another problem still exists. When the accuracy of the deep network tends to saturation, serious degradation occurs. The residual element added by ResNet through a short-circuit mechanism can effectively solve the problems of gradient explosion or disappearance and network degradation. The schematic diagram of the residual unit structure in ResNet is shown in Figure 3.

Assume that the initial input value of the neural network is \( m \), and \( H (m) \) is the output value of the network stack layer. For the ordinary linear network structure, the function \( F (m) \) set by the learning objective is the same as the function \( H (m) \) of the above low-level mapping. However, the residual network structure is different. To prevent the deep network from degradation and at least retain the original information, \( H (m) \) not only contains the mapping \( F (m) \) output through network learning but also directly adds the input features of identity mapping to retain the original information while learning the features. The relation between \( F (m) \) and \( H (m) \) can be represented by equation \( F (m) = H (m) - m \). In this way, the situation of the latter layer in the general neural network model can only be learned by the previous layer, and the identity mapping can be carried out nonindirectly to avoid the degradation problem in the deep network. The residual element is defined as the following formula:

\[
m_{i+1} = f(m_i + F(m_i, W_i)).
\]

In formula (6), \( m_i \) and \( m_{i+1} \) are, respectively, the input and output of the \( i \)th residual unit; \( F (m_i, W_i) \) represents the learned residual mapping; \( W_i \) is the weight matrix of the \( i \)th residual unit; and \( f \) is the relu activation function, adopting rectified linear units. The activation function \( f \) is shown as follows:

![Grid feature extraction method.](image)

**Figure 1:** The overall architecture of JGRCAN.

![Schematic diagram of the residual element structure.](image)

**Figure 2:** Grid feature extraction method.

![Schematic diagram of the residual element structure.](image)

**Figure 3:** Schematic diagram of the residual element structure.
\[ f(m) = \max(0, m). \] (7)

Given a preprocessed image, feature images of 64 \( \times \) 224 \( \times \) 224 are generated by the first convolution layer. After dimensionality reduction of the maximum pooling layer, the output size is 64 \( \times \) 112 \( \times \) 112, and then the output feature images are input to the average pooling layer by several residual blocks in each module. A matrix of 7 \( \times \) 7 \( \times \) 2048 is obtained, and the dimension transformation in the convolution process is shown in Table 1.

After residual cell calculation and average pooling, we finally sent the feature map to the reshape layer. The function of reshape layer is to convert it into grid feature \( P_g \), and its transformation formula is shown as follows:

\[ P_g = \left[ \text{merge}_{2,3}(I) \right]^T \epsilon(49,2048). \] (8)

In formula (8), \( \text{merge} \) \( 2, 3 \) is a dimension combination function. For example, the shape of the input feature I obtained from the above table is 2048, 7, and 7. After dimension, combination, and transpose, the shape will change to 49 and 2048.

### 3.1.3. Region Features

The idea of faster-RCNN is mainly adopted for region feature extraction, and the main process includes three parts: the feature extraction module, the proposal region extraction module, and the target detection module. The specific extraction process of region features is shown in Figure 4.

In the feature extraction module, the convolutional layer of VGG-16 is mainly used to extract image region features. The full name of VGG-16 is GG-Verydeep-16 CNN, which contains 13 convolutional layers and 3 full connection layers. It deepens the number of network layers while adopting a 3 \( \times \) 3 convolution kernel and sets the step size of the convolution layer as 1, which is faster than other models in convergence. The features in the input image are extracted through the 3 \( \times \) 3 convolution kernel to obtain the feature map of the image, which can be shared by the subsequent region proposal network (RPN) layer and region of interest (ROI) pooling layer.

In the proposal region extraction module, the main task is to output the set of bounding boxes from the feature map obtained from the feature extraction module, of which the core part is the region proposal network (RPN), as shown in Figure 5. RPN is a full convolutional network, which reduces the number of candidate bounding boxes in the detection process and improves the detection efficiency of faster-RCNN. The RPN network mainly has two branches: one is the classification layer, which is used to judge positive or negative samples and binary classification; the other is the regression layer, which is used to predict the position of the positive sample candidate box. RPN implementation process: A 3 \( \times \) 3 sliding window is used to slide on the feature map to traverse each pixel of the feature map to generate low-dimensional feature maps, and K predefined bounding boxes are generated for each pixel position on the feature map. Then, two 1 \( \times \) 1 convolution operations are performed on the low-dimensional feature images, respectively, and 2k probability values and 4k bounding boxes offset values are obtained at each pixel. Finally, postprocessing operations such as boundary-cutting, small bounding boxes removal, and nonmaximum suppression (NMS) are carried out in combination with the predefined bounding boxes to obtain the candidate bounding boxes. The multitask loss function is used in the training of the RPN network, and the calculation method is shown in the following equation:

\[ L(P_i, t_i) = \frac{1}{N_{cls}} \sum \lambda_{cls} (p_i, p_i^*) + \frac{1}{N_{reg}} \sum \lambda_{reg} (t_i, t_i^*). \] (9)

In formula (9), \( i \) represents the ith anchor; \( p_i \) is the prediction probability of the target as the ith anchor. When the ith anchor is the target, \( p_i \) is 1; otherwise, \( p_i \) is 0. \( t_i \) is a four-dimensional vector which represents the coordinates and size of the predicted bounding box. \( N_{cls} \), \( N_{reg} \), and \( \lambda \) are coefficients; \( p_i^* \) is the marked label of whether ith anchor is the target; \( t_i^* \) represents the coordinates and size of the marked bounding box of the ith anchor; and \( L_{cls} \) and \( L_{reg} \) are loss functions of classification and regression, respectively.

In the target detection module, the candidate bounding boxes obtained are input into the ROI pooling layer and fixed into a uniform size. Then, the output features from ROI are input into the full connection layer combined with the feature map extracted by the VGG-16 convolution neural network. Finally, the probability of a target and a nontarget is output through the full connection layer. When the probability of a target exceeds the threshold value of 0.7, it is considered that there are targets in the bounding box. Then, the target bounding box is added inside the region features and the maximum number is limited to 50. Therefore, the region features can be expressed as \( \text{Pr} \in (50, 2048) \).

### 3.2. Co-Attention Layer

The co-attention layer uses an encoder-decoder structure, consisting of one encoder and two decoders. The encoder layer contains some parallel question self-attention units, and the decoder contains image self-attention units and guided-attention units, as shown in Figure 6.

#### 3.2.1. Encoder

The original question features \( E \) taken as input \( Q(0) \) and output \( Q(1) \) through the self-attention unit. The input of each layer is the output of the previous layer, and the calculation formula for this process is shown in the following formula:

\[ Q(t) = \text{SA}(Q(t-1)). \] (10)

In formula (10), \( t \) is the layer number of encoder and decoder, and SA is similar to the self-attention method and the scaled dot-product attention and multiattention mechanism based on [25]. After encoding, the attention vector representation of the question is obtained, and the attention vector representation of the question is sent to the
decoder to focus on the important areas of the image by guided-attention.

3.2.2. Decoder. We have two decoders for grid features and region features. First, the original visual features are input into the self-attention unit to obtain the visual features of self-attention. Secondly, after self-attention, the visual features and question features are sent to the guided-attention unit. Finally, the output features of each decoder layer, including grid features and region features, are defined as the following formula:

\[
P^g_g(i) = GA(SA(P^g_g(i-1))),
\]

\[
P^r_r(i) = GA(SA(P^r_r(i-1))).
\]

GA is similar to the guided-attention mechanism in [2], both of which are based on the scaled dot-product attention and multihead attention mechanism in [25].

3.3. Fusion Layer. After the co-attention layer learning, grid features, region features, and question features have learned their own attention weight distributions.
First of all, in order to avoid excessive computation, we adopted two-layer MLP to reduce computation. The attention features of the grid, region, and question can be obtained by calculating the following formulas:

$$P_g = \sum_{i=1}^{m} a_g P_g^i, \quad (13)$$

$$P_r = \sum_{i=1}^{m} a_r P_r^i, \quad (14)$$

$$Q = \sum_{i=1}^{l} a_q Q_i. \quad (15)$$

The $$a_g = \{a_1, a_2, ..., a_m\}$$, $$a_r = \{a_1, a_2, ..., a_n\}$$, and $$a_q = \{a_1, a_2, ..., a_l\}$$ is the distribution of attention weight learned from the features of grid, region, and question. Next, the joint representation $$v$$ after the fusion of these three different features is shown as follows:

$$v = [P_g + Q, P_r + Q], \quad (16)$$

where $$V \in R^{2xd}$$ is the feature representation after the fusion of grid, region, and question features, and it is also the feature representation of the answer. Finally, the fused features are sent into the fully connected layer, and the prediction answer $$c$$ is generated after the sigmoid activation function. The calculation formula is shown in the following formula:

$$C = \sigma (f_c (v)). \quad (17)$$

In order to train JGRCAN, we adopted binary cross entropy (BCE) as the loss function. The binary cross entropy loss function is defined as shown in the following formula:

$$\text{BCELoss} (o, t) = - \frac{1}{n} \sum_{i=1}^{n} [t_i \log c_i + (1 - t_i) \log (1 - c_i)]. \quad (18)$$

Among the formulas, $$o$$ is a candidate answer; $$t$$ is the real answer; and $$n$$ is the number of all questions.

4. Experiment

4.1. Dataset. In our experiments, we used the most popular and publicly available dataset, VQA-v2 [26, 27], including images from the websites and manually annotated question-answer pairs. For basic information about VQA-V2, it contains 204K images and 1.1M QA pairs. There are three types of questions in the dataset: Y/N, number, and other. Each image contains 3 questions, and each question contains 10 answers. The dataset is divided into a training set, validation set, and test set, with the proportion of 40%, 20%, and 40%, respectively. The test set contains two test subsets: test-dev and test-std. There are some dataset samples listed in Figure 7.

Compared with VQA-v1, VQA-v2 not only minimizes language bias but also adds some images and manually annotated question and answer pairs. VQA-v2 contains three different types of questions: Y/N, number, and other. These are the three different types of questions. Y/N includes yes/no questions and only answers yes or no, such as when a user asks "Is the umbrella upside down?" in pic XXX; number includes counting questions and only answers numbers, such as when a user asks "how many...?"; other includes asking about color and type, such as when a user asks "what color is...", "what food is...."  

4.2. Experimental Details. Since the size of each image is not completely consistent, in order to adapt to the input layer, we performed uniform preprocessing on the images and scaled each image to 448×448. In order to extract visual features, the pretrained faster-RCNN [6] is used to extract region features from images, and the maximum upper limit of region extraction is set to 50. Grid features are extracted by ResNet101 and 1×1 AvgPool [14], and the hidden size of grid and region features is 2048. Glove [28] is used as a word embedding. The dimension of the word vector is set at 300. The embedding vector is input into LSTM, and the size of the LSTM hidden layer is 512.

In the co-attention layer, referring to the parameters in [14], the attention dimension $$D$$ of multiple heads is set as 512; the number of heads $$H$$ is fixed as 8; the hidden dimension of each head is $$dh = D/H = 512/8 = 64$$; the number...
of encoder and decoder layers is 4; and the feature dimension after fusion is 1024. To train the JGRCAN, we used Adam as the optimizer, then set the learning rate to 0.0001 and the batch size to 48 and train the network for 20 epochs.

4.3. Experimental Results. In order to fully verify the effectiveness of the joint grid-region feature fusion network proposed in this paper, we compare three cases: grid feature only, region feature only, and fusion grid feature and region feature. We train and verify the dataset for comparing the performance of these three features, and the experimental results are shown in Table 2.

Experimental results show that the performance of JGRCAN is better than that of only using a grid or region, which proves the effectiveness of JGRCAN.

In addition, we selected several popular methods in recent years to compare the accuracy of answering various questions on test-dev and test-std datasets. The overall experimental results are shown in Table 3. As can be seen from the table, the proposed JGRCAN is 0.24% more accurate than the MCAN of the benchmark network in the test-dev subset and improves the overall accuracy of the test-std test subset, verifying the positive role of the network in answering certain types of questions. It can also be seen that the overall performance of JGRCAN is better than other network models.

The model JGRCAN is not compared with the other models based on transformer [31–35] in visual question answering. Although this type of model is better than JGRCAN in performance, its computation speed is much slower than JGRCAN.

We also visualized some of the results of JGRCAN and gave four examples of the VQA-v2 test set, as shown in Figure 8. Figure 8(a) shows that two different image features of our model can focus on the corresponding correct image region. In Figure 8(b), the region features have a higher attention weight on the seven ships to generate correct answers, while the grid features focus on some of the wrong regions. In Figure 8(c), the grid features are able to notice the lines next to the sidewalk, while the region features fail because there is no region box covering these objects; Figure 8(d) also shows a failure case in which the model fails to generate correct answers because the classification label “no right turn” does not exist in the training set, although the correct image region is involved.

**Table 2:** Performance comparisons with different feature combinations.

<table>
<thead>
<tr>
<th>Features (%)</th>
<th>All</th>
<th>Y/N</th>
<th>Number</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid</td>
<td>70.56</td>
<td>86.27</td>
<td>52.21</td>
<td>61.85</td>
</tr>
<tr>
<td>Region</td>
<td>70.63</td>
<td>86.82</td>
<td>53.06</td>
<td>60.72</td>
</tr>
<tr>
<td>Grid + region</td>
<td>70.87</td>
<td>86.97</td>
<td>52.88</td>
<td>61.45</td>
</tr>
</tbody>
</table>

Y/N, number, and other are three different types of questions. Y/N includes yes or no questions and only answers to yes or no, for example: “Is it...?”, “Does it...?” Number questions include counting questions, answering only numbers, e.g., “how many...?” Other includes asking about color, type, etc. For example, “what color is...?” and “what food is...?”

**Table 3:** Experiment results compared with other state-of-the-art models on the test-dev and test-std sets.

<table>
<thead>
<tr>
<th>Methods (%)</th>
<th>All</th>
<th>Test-dev</th>
<th>Test-std</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-down</td>
<td>65.32</td>
<td>81.82</td>
<td>44.21</td>
</tr>
<tr>
<td>BAN [16]</td>
<td>69.52</td>
<td>85.31</td>
<td>50.93</td>
</tr>
<tr>
<td>BAN + counter [16]</td>
<td>70.04</td>
<td>85.42</td>
<td>54.04</td>
</tr>
<tr>
<td>MuRel [29]</td>
<td>68.76</td>
<td>84.77</td>
<td>49.84</td>
</tr>
<tr>
<td>DFAF [30]</td>
<td>70.22</td>
<td>86.09</td>
<td>53.32</td>
</tr>
<tr>
<td>MCAN [2]</td>
<td>70.63</td>
<td>86.82</td>
<td>53.26</td>
</tr>
<tr>
<td>JGRCAN</td>
<td>70.87</td>
<td>86.97</td>
<td>53.15</td>
</tr>
</tbody>
</table>

**Figure 7:** Some samples from VQA-v2.
5. Discussion

Bottom-up, MCAN, and other methods use faster-RCNN to extract the regional features of the image, but this method has certain disadvantages. As shown in Figure 8(c), when using the region features to judge the color of non-targets, it cannot pay attention to the important features. It only relies on the targets in the input image to identify the most meaningful areas and neglects the details outside the non-targets. We improved the traditional bottom-up region feature extraction method and proposed a dual feature extraction method of grid feature and regional feature, which realized the complementary advantages of regional feature and grid feature. Using two decoders instead of one in the collaborative attention layer can accurately focus on the area related to the question and answer in the image. Finally, the region features, grid features, and question text features are fused to transfer dynamic information between vision and language modality.

6. Conclusions

In this paper, a visual question answering co-attention network via joint grid-region features (JGRCAN) is proposed to complement grid features and region features. Our model contains co-attention layers composed of encoder-decoder structure. Each layer contains self-attention units and guided-attention units, and the important regions in the grid features and region features are noticed through the question attention features encoded by the encoder. In addition, to fuse the two different features more effectively, a feature fusion layer is designed to reduce the computational burden of the model through a two-layer MLP and avoid semantic noise after direct fusion. Finally, we verify the performance of our model by comparing it with other popular methods on the VQA-v2 dataset. Experimental results show that the fusion of grid features and region features in JGRCAN can significantly improve the performance of the VQA-v2 task.

However, our method also has two disadvantages: first, compared with MCAN [2], the accuracy of the count type is reduced, which may be because the grid feature is not conducive to identifying small targets. In the future, we will consider adding a counting module to improve the accuracy of the count type questions. Second, JGRCAN simply uses the concatenation method to fuse grid features and region features, which may lose some important features in the process of fusion. To solve this problem, we consider improving the fusion algorithm to reduce noise to improve the overall performance in the future.
Data Availability
The data can be downloaded from https://visualqa.org/.

Conflicts of Interest
The authors declare that they have no conflicts of interest.

Authors’ Contributions
Jianpeng Liang and Tianjiao Xu contributed equally to this work.

Acknowledgments
This research was supported in part by the National Natural Science Foundation of China (grant no. 6206053), Ministry of Education of Humanities and Social Science Project (grant no. 20YJA740031), and Science and Technology Program of Guangzhou (grant no. 202102020878).

References


