Positional Attention Guided Transformer-like Architecture for Visual Question Answering

Aihua Mao, Zhi Yang, Ken Lin, Jun Xuan, Yong-Jin Liu, *Senior Member, IEEE*

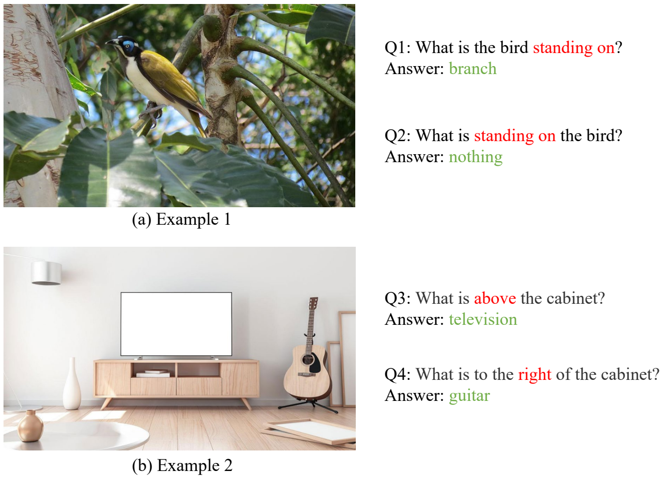
***Abstract*—Transformer architectures have recently been introduced into the field of visual question answering (VQA), due to their powerful capabilities of information extraction and fusion. However, existing Transformer-like models, including models using a single Transformer structure and large-scale pre-training generic visual-linguistic models, do not fully utilize both positional information of words in questions and positional information of objects in images, which are shown in this paper to be crucial in VQA tasks. To address this challenge, we propose a novel positional attention guided Transformer-like architecture, which can adaptively extracts positional information within and across the visual and language modalities, and use this information to guide high-level interactions in inter- and intra-modality information flows. In particular, we design and assemble three positional attention modules into a single Transformer-like model MCAN. We show that the positional information introduced in intra-modality interaction can adaptively modulate inter-modality interaction according to different inputs, which plays an important role for visual reasoning. Experimental results demonstrate that our model outperforms the state-of-the-art models and is particularly good at handling object counting questions. Overall, our model achieves the accuracy of 70.10%, 71.27%, 71.52% on the datasets of COCO-QA, VQA v1.0 test-std and VQA v2.0 test-std, respectively. The source code will be publicly available at https://github.com/waizei/PositionalMCAN.**

***Index Terms*—Visual question answering, Positional attention, Transformer-like models.**

I. Introduction

VISUAL question answering (VQA) [1] is a task in which a machine automatically provides an appropriate answer to a natural language question about a given image. Vision and language are two major modalities in human communication, and many representative works have been proposed for the information fusion and processing of these two madalities, e.g., image caption [2], image-text matching [3], and multimodal compatibility modeling [4], etc. Compared to the tasks in these representative works, VQA is a much difficult task that needs a deep understanding of image semantic scenes and fine-grained multimodal reasoning [5] to get accurate answers.

So far, most existing researches on VQA are mainly based on the attention mechanism. Many of them adopt shallow co-attention layers to relate questions to key objects in images (e.g., [2], [6]). State-of-the-art VQA models, such as MCAN [7], DFAF [8] and large-scale pre-training models [9], [10], [11], further adopt Transformer-like architectures which greatly improve the results on VQA tasks. However, these Transformer-like models either do not consider positional information, or simply use partial positional information in a straightforward way; e.g., MCAN possibly encodes some positional information by convolutional neural networks in the image pre-processing stage [12] and some BERT[13]-model-inspired large-scale pre-training models directly concatenate positional features to semantic features before the attention operations, which usually generate potential noises [14].


Q1: What is the bird standing on?
Answer: branch
Q2: What is standing on the bird?
Answer: nothing
Q3: What is above the cabinet?
Answer: television
Q4: What is to the right of the cabinet?
Answer: guitar

(a) Example 1

(b) Example 2

Fig. 1. Examples of the importance of positional information in VQA tasks. In (a), due to the semantic difference between Q1 and Q2, the positional information of the words "standing on"is crucial to correctly answer both questions. In (b), to answer Q3 and Q4 correctly, we must determine the positional relationship between the cabinet, television and guitar to check whether the conditions "above" and "right" are met.

All above VQA works do not make full use of the positional information in both natural language questions and images. In principle, positional information are very useful for visual reasoning [15]. As illustrated in Figure 1, our study presented in this paper is based on two key observations: (1) the position of words in a sentence is crucial to understanding natural language questions; for instance, although Q1 and Q2 use the same words, their semantics are completely different due to the different locations of the phrase "standing on", and (2) the position of objects in an image is indispensable for visual reasoning; for instance, to answer Q3 and Q4, we must find the cabinet first and then determine what is on top and what is on the right side, that is, the positional relationship between the cabinet, TV and guitar is the basis for reasoning. The

A.H. Mao, Z. Yang, K. Lin and J. Xuan are with School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. E-mails: ahmao@scut.edu.cn, 202021044575@mail.scut.edu.cn, csklin@mail.scut.edu.cn, 202020143921@mail.scut.edu.cn,

Y.J. Liu is with BNRist, the Department of Computer Science and Technology, Tsinghua University, and MOE-Key Laboratory of Pervasive Computing, Beijing 100084, China. Corresponding author, E-mail: liuyongjin@tsinghua.edu.cn.

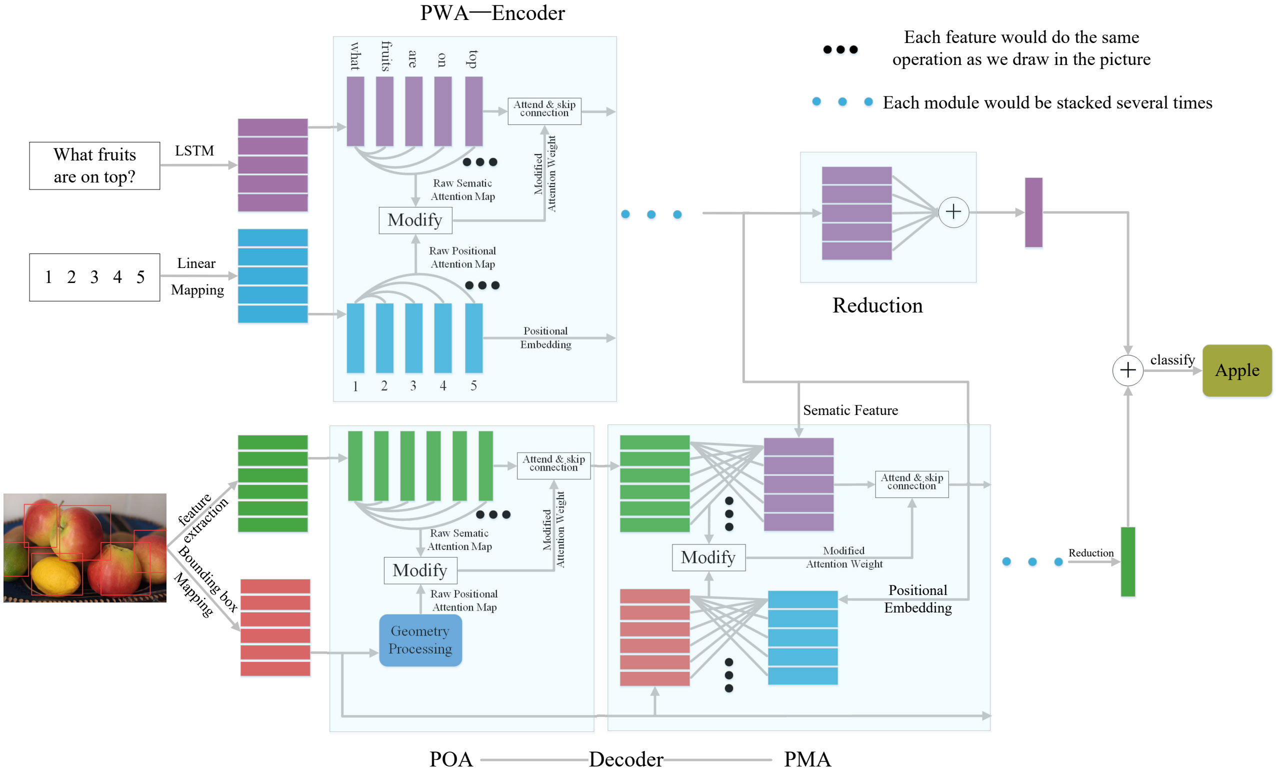


Fig. 2. The framework of our proposed positional attention guided Transformer-like architecture. We adopt an encoder-decoder Transformer-like structure and replace the core attention mechanisms with our proposed PWA, POA and PMA. Specifically, we stack PWA six times as the encoder, and stack six layers of POA followed by PMA as the decoder.

key challenge in our study is that when using co-attention to fuse the information in vision and language modalities, how to extract and maintain correct positional information for fusion and reasoning in a novel structure compatibly with a Transformer-like model.

In this paper, we propose a novel positional attention guided Transformer-like model to improve the visual reasoning in VQA. To embed the positional information in a Transformer-like model, we use the attention weights obtained by positional features to modify the attention weights of semantic features, inspired by Relation Networks in object detection [16]. Specifically, as shown in Figure 2, we design three positional attention modules: (1) a positional word-to-word attention module (PWA), which focuses on improving the understanding of natural language questions, (2) a positional object-to-object attention module (POA), which can construct accurate positional relationship between objects; POA is also able to handle the questions related to object-counting type well, which is one of major challenges in current VQA research, and (3) a positional multi-modality fusion attention module (PMA), which guides our VQA model to fuse vision and language modalities appropriately. The ablation study in Section IV-D shows that each of these three modules can boost performance well.

We make the following contributions in this paper:

- We propose a novel positional attention structure which consists of three modules of PWA, POA and PMA, and is compatible with the Transformer-like architecture to fully take account of the positional information.
- Our proposed positional attention guided Transformer-like architecture, to the best of our knowledge, is the first to use the attention map obtained from positional information to modify the attention map of semantic features for tackling VQA tasks. Such operation enables the positional information to guide the interactions in intra- and inter-modality information flows and thus achieve performance improvement.
- Extensive evaluations on three benchmark datasets including COCO-QA, VQA v1.0 and v2.0, demonstrate that our proposed modules can effectively outperform state-of-the-art models, especially in handling the questions related to object counting.

II. Related Work

A. Early VQA Work

VQA has attracted increasing attention in the past few years. In the early studies of VQA, the most straightforward method is to directly combine image features and question features for classification. Ren *et al.* [17] adopted LSTM to predict the answer, and used the image features extracted by CNN as its initialization parameters. DppNet [18] replaces the last layer of VGG [19] by a fully connected layer with dynamic parameters, allowing it to change according to different input questions. However, these methods treat all objects in an image equally, and possibly lose focus on the key objects. Module networks [20] [21] were also proposed to imitate the process of human reasoning in VQA tasks, by introducing several jointly-trained neural modules with different functions (e.g., attention, measurement) and then parsing the question into some parts which can be processed by these modules. However, these models only work well in the datasets containing artificial images, since the module networks cannot accomplish the analysis for the questions about complicated situations in the real images.

B. Attention Mechanism for VQA

Recently attention-based approaches have become popular because they allow the model to focus on critical components or parts. SAN [22] used a stacked attention module for multi-step reasoning. HieCoAtt [23] analyzed the language modality using three dimensions: word, phrase and question, and proposed a novel co-attention module for multimodal reasoning.

Different from the above CNN-based attention networks, detection-based VQA networks made use of refined features obtained by attention maps, which were generated through comparison between questions and objects in an image. BUTD [2] introduced a pre-processing strategy to extract object features from images with Faster RCNN [24], so that the model can focus on the key objects by assigning large attention weights. Inspired by the success of BUTD [2], most subsequent models attempted to optimize the attention mechanism applied in VQA tasks. CVA [25] selected crucial features in both channel and spatial dimensions. ODA [26] calculated the probability of attention between objects. DenIII [27] modeled dense inter-, and intra-modality interactions. BAN [28] adopted the low-rank bilinear pooling technique to model full interaction between vision and language modalities, and achieves fast inference speed. Dual-MFA [29] leveraged both grid-level and object-level features for reasoning. ALSL [30] used the annotations of VQA-HAT [31] to supervise the obtained attention map, and applied adversarial learning between free-form-based and detection-based attention modules. RE-ATT [32] made use of attention twice, and can pay attention to high-order features. AttReg [33] applies regularization approach for better visual grounding in VQA. However, these methods still ignored the dense relations between each word in questions and each object in images. As a comparison, our method is based on the dense co-attention structure (i.e., a Transformer-like model) and can more accurately account for attention features.

C. Transformer-like Architecture in VQA

Since the pioneering Bert model [13] was proposed, Transformer structures have been introduced into many important research areas. Recently, several Transformer-like models have been proposed for VQA tasks, which achieved remarkable progresses. Representative works that used single Transformer-like structure include MCAN [7] and DFAF [8], both of which use self-attention in vision and language modalities and use co-attention for modality fusion. MCAN [7] adopted an encoder–decoder structure, in which self-attention for the language modality was used for the encoder, and self-attention for the vision modality and co-attention were used for the decoder. DFAF [8] stacked several layers and executed co-attention before self-attention in vision and language modalities in each layer. MCAoA [34] introduced a secondary attention and a new fusion module based on MCAN.

Another Transformer-like solution for VQA tasks is to use pre-training generic visual–linguistic representations. After accomplishing some large-scale pre-training tasks, such as masked language modeling and masked visual-feature classification, these representations can be easily generalized for various downstream tasks, such as VQA, visual commonsense reasoning, and grounding referring expressions. Some recent works such as VL-BERT [9], ViLBERT [10] and LXMERT [11] are all based on the backbone of Transformer but with different structures. VL-BERT [9] is a single cross-modal Transformer, and ViLBERT [10] adopts one single-modality Transformer (i.e., language a modality) and one cross-modal Transformer with restricted attention pattern. LXMERT [11] uses two single-modality Transformers (i.e., vision and language modalities) and one cross-modal Transformer.

Both single and large-scale pre-training Transformer-like models apply self-attention and co-attention units of Transformer [35] as their basic modules but still pay insufficient attention to positional information in natural language questions and images.

D. Processing of Positional Information

In natural language processing, positional features are important for Transformer [35] since it processes input sentences in a parallel manner. The positional information of words must be introduced additionally so that Transformer can understand the sentence appropriately. Several studies have attempted to improve the way of handling word positions in Transformer. Shaw et al. [36] replaced the encoding of absolute positions with efficient representations of relative positions. Raffel et al. [37] simplified the method in [36] by adding relative positional encoding as inductive bias. Wang et al. [38] extended positional encoding to a complex-valued domain so that the representation of words can shift smoothly with updated positions. Ke et al. [14] decoupled the original method that directly adds position and word embeddings to avoid the introduction of noise. The processing of word positions in Transformer can also be extended to the positions of bounding boxes extracted from images. Hu et al. [16] modeled the relation of objects in images with their appearance feature and geometry, and eliminated the parts that require manual intervention in the current object detection pipelines. Swin [39] projected absolute and relative positional information with indexes into attention maps.

However, in the field of VQA, there are only a few researches which specially consider positional information, including both the word positions in natural language questions and the location of objects extracted from images. REGAT [40] used the graph attention network [41] and geometric positions to explore explicit relations between objects. Zhang et al. [6] used the object locations in images to produce counting features by eliminating duplicate bounding boxes in a differentiable manner. Huang et al. [42] built a relational graph through the distance between objects. Although these methods achieved good performance by introducing positional information, these methods are not compatible with state-of-the-art Transformer-like models. Because [40] and [42] are both based on a graph neural network, and [6] takes counting as a single module, they all need to be used independently and thus cannot be connected into multi-head attention of Transformer. As a contrast, our method is the first to make full use of positional information in a Transformer-like model for VQA tasks, which enables the model to capture more effective correlations between natural language questions and images.

III. Our Method

Aiming to introduce positional information appropriately for VQA and avoid noise generation, we propose novel positional attention modules use them to replace the core attention mechanisms in a Transformer-like architecture (Figure 2). Our key idea is to fuse the attention maps of semantic and positional features, and thus both of them contribute to determine which objects in image or words in question should be focused. In particular, our proposed positional attention modules are compatible with Transformer-like models. We first summarize the whole pipeline of our model and then present our proposed positional attention modules in detail.

A. Pipeline of Transformer-like Architecture

Transformer-like models usually adopt bottom-up and top-down features [2] for visual representation. In our method, we use the Faster RCNN model with ResNet-101 as the backbone and conduct the pre-training on the Visual Genome dataset [43]. Then we extract the intermediate features and apply non-maximum suppression. Finally, we obtain the features of objects $I \in \mathbb{R}^{m\times 2048}$ for each image, where m is the number of objects and we set its range[1] to be 100. We pad or truncate the given question to a certain length (i.e., 14 in our implementation) to facilitate subsequent processing. After that, we initialize a recurrent neural network with LSTM by GLoVe word embeddings [44] to encode the question by $Q \in \mathbb{R}^{14\times n}$, where n is a hyperparameter and $1 \times n$ is the dimension of text features. Since the semantic features of both questions and images are available, we are readily to obtain the positional information of words (i.e., their positions in a sentence) and objects (i.e., the locations of their bounding boxes in an image).

We follow MCAN [7] to use an encoder-decoder structure in our pipeline (Figure 2): we use the PWA module as the encoder, and use POA and PMA modules as the decoder. It is worth noting that since our proposed module design takes self-attention and co-attention as basic units, these modules are compatible with all Transformer-like architectures. Specifically, we stack PWA modules six times as the encoder, which corresponds to self-attention for language modality in Transformer. The decoder has six layers and each layer consists of a POA module and a PMA module. Similarly, POA modules correspond to self-attention for vision modality and PMA modules corresponds to co-attention in vision and language modalities. Finally, the encoder outputs question features $E \in \mathbb{R}^{14\times d}$, and the decoder outputs image features $D \in \mathbb{R}^{100\times d}$ (we pad the number of objects in each image to be 100), where d is a hyperparameter representing the dimension size of a single feature in attention, which is determined in the training process.

Then we combine these features as a composite features for prediction. Let $E = [E_1, ..., E_{14}]$, $E_i \in \mathbb{R}^d$, and $D = [D_1, ..., D_{100}]$, $D_i \in \mathbb{R}^d$. We compute the middle features $E^{'}$ and $,D^{'}$ as follows:

$$\begin{aligned} E^{'} &= \sum_{i=1}^{14} \text{softmax}(MLP(E_i))E_i \\ D^{'} &= \sum_{i=1}^{100} \text{softmax}(MLP(D_i))D_i \end{aligned} \tag{1}$$

The final composite feature F used for prediction is defined as:

$$F = \text{LayerNorm}(E^{'}W_E^T + D^{'}W_D^T) \tag{2}$$

where $W_E^T, W_D^T \in \mathbb{R}^{d\times d_F}$ denote projection matrices, and layer normalization [45] is used for numerical stability. With the composite feature F, we use the sigmoid activation function to predict a score $\tilde{s}$ for each candidate answer:

$$\tilde{s} = \sigma\big(\omega_o F\big) \tag{3}$$

where σ is the sigmod function and ω_o denotes a linear mapping. Finally, following the strategy in [46], we use the binary cross-entropy (BCE) loss function to train the model on a N-way classifier:

$$L = -\sum_{i}^{M}\sum_{j}^{N} s_{ij}log(\tilde{s}_{ij}) - (1 - s_{ij})log(1 - \tilde{s}_{ij}) \tag{4}$$

where N is the number of candidate answers (see Section IV-B), M is the batch size, $\tilde{s}$ is the soft accuracies (see Section IV-A) of the ground-truth answers and s is the accuracy of predicted results.

B. General Structure of Positional Attention Modules

Our pipeline makes use of three types of positional attention modules, namely, PWA, POA and PMA. To be compatible with Transformer-like architectures, we design these modules based on a general structure. Before presenting the design details of the three positional attention modules in Sections III-C-III-E, we first present the general structure adopted by them.

Refer to Figure 3. X_{sem} and Y_{sem} represent the semantic features of X and Y modalities, and X_{pos} and Y_{pos} represent their positional features. Since it is a general structure, X and Y can be similar or different, corresponding to self-attention and co-attention respectively. The general structure uses $X_{sem}, Y_{sem}, X_{pos}, Y_{pos}$ as input, and then Y modality

[1] For the number of objects less or larger than 100, the feature representation is padded or truncated to $m = 100$.

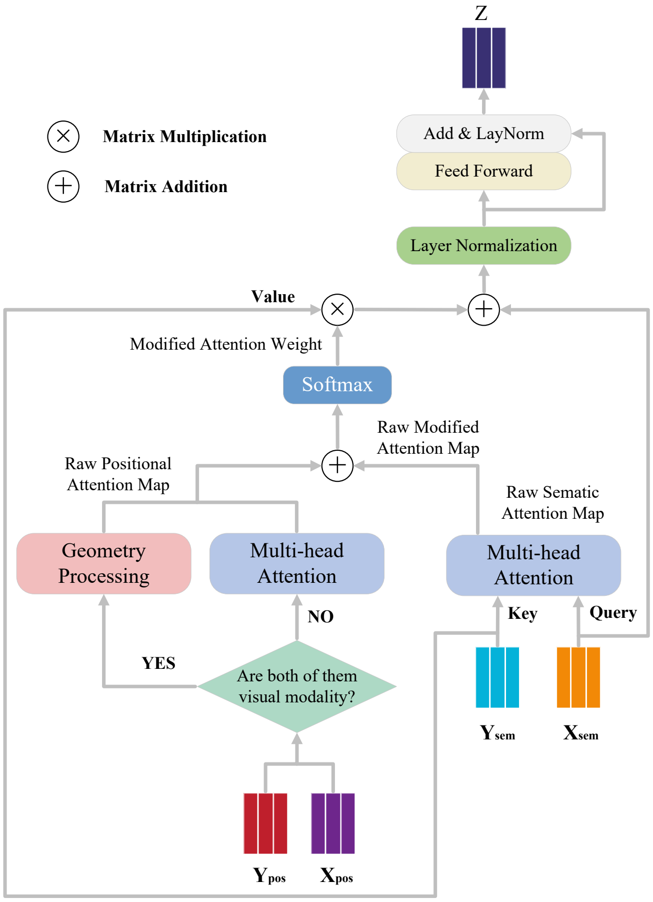


Fig. 3. The general structure of proposed positional attention modules. This structure takes the semantic features (X_{sem}, Y_{sem}) and the positional features (X_{pos}, Y_{pos}) of X and Y modalities as input and outputs the composite features Z for X guided by Y. X and Y can be same or different modalities, corresponding to self-attention and co-attention. The input 'key' and 'value' needed for PMA are obtained from the output of the encoder (see Figure 2).

guides the X modality to perform several attention operations and output the composite feature Z.

Multi-head attention [35] is a kind of scaled dot-product attention with several parallel *heads*, which aims at enhancing the representation capability. It is the core mechanism of Transformer and is adopted in all our proposed modules. The input of scaled dot-product attention consists of queries $Q \in \mathbb{R}^{m \times d_{query}}$, keys $K \in \mathbb{R}^{n \times d_{key}}$ and values $V \in \mathbb{R}^{n \times d_{value}}$. For simplicity, we set $d_{query} = d_{key} = d_{value} = d$. Here, we explain how the i-th object of X obtains the composite feature Z_i under the guidance of Y, and Z can be easily obtained through tensor operations in Pytorch. For a given query $q \in \mathbb{R}^{1 \times d}$ and K, the raw attention map is obtained by measuring the similarity between the query and keys:

$$f(q, K) = \frac{qK^T}{\sqrt{d}} \tag{5}$$

To strengthen the representation ability of the modules, the multi-head operation, which has h heads corresponding to independent attention maps of the query, is performed as:

$$a_i = \text{Concat}[head_{i1}, ..., head_{ih}], \text{ for } i \in [1, m] \tag{6}$$

$$head_{ij} = f(Q_i W_j^Q, K W_j^K) \tag{7}$$

where $W_j^Q, W_j^K \in \mathbb{R}^{d \times d_h}$ are the projection matrices for the j-th head, d_h denotes the dimension of output features for each head, which is set to be $d_h = d/h$, and $a_i \in \mathbb{R}^{h \times n}$ is the attention map of Q_i to K.

As illustrated in Fig. 3, when receiving X_{sem} and Y_{sem}, the structure performs the computation in Eq. (6) to obtain the raw semantic attention map $A_i^{sem} \in \mathbb{R}^{h \times n}$ of X_i to Y. For the input of X_{pos} and Y_{pos}, if both X and Y are the vision modality, they are further sent to geometry processing (Section III-D); otherwise, they are undertaken the same method as X_{sem} and Y_{sem} (see details in Sections III-C and III-E). After that, we obtain the raw positional attention map $A_i^{pos} \in \mathbb{R}^{h \times n}$ of X_i to Y. In literature, to merge the raw positional attention map and semantic attention map, Relation Networks [16] adopts $A_i = \text{softmax}(A_i^{pos} \cdot \exp(A_i^{sem}))$, indicating that the positional relationship determines whether to use semantic features. As a contrast, we use the scaled addition of A_i^{pos} and A_i^{sem}, so that they both contribute to the obtained attention map A_i:

$$A_i = \text{softmax}(\frac{A_i^{pos} + A_i^{sem}}{\sqrt{2}}) \tag{8}$$

The ablation study in Section IV-D demonstrates the effectiveness of using Eq. (8). Then the output of the whole attention operation $O_i \in \mathbb{R}^{1 \times h*d_h}$ is a weighted average of the value $V \in \mathbb{R}^{n \times d}$ based on A_i. Based on the mechanism of the multi-head attention, O_i is concatenated by all the parallel heads $O_{ij}, j \in [1, h]$:

$$O_{ij} = A_{ij}(V W_j^V) \tag{9}$$

$$O_i = \text{Concate}[O_{i1}, ..., O_{ih}] \tag{10}$$

where $A_{ij} \in \mathbb{R}^{1 \times n}$ denotes the attention weight of the j-th head in A_i, and $W_j^V \in \mathbb{R}^{d \times d_h}$ is the projection matrix for V in the j-th head. Then, we project the obtained O_i to $\mathbb{R}^{1 \times d}$ using $W^O \in \mathbb{R}^{h*d_h \times d}$. In order to make the model stable, we add a residual link and perform the layer normalization [45]:

$$Z_i^{'} = \text{LayerNorm}(Q_i + O_i W^O) \tag{11}$$

Following the Transformer structure, we perform feed forward and a residual link after layer normalization to get final output of the module. Feed Forward Net (FFN) is constructed by two fully connection layers with RELU activation:

$$Z = \text{LayerNorm}(Z^{'} + \text{FFN}(Z^{'})) \tag{12}$$

Based on the above general structure, the three positional attention modules, namely, PWA for improving question understanding, POA for capturing dense interaction between objects in an image, and PMA for fusing vision and language modalities, are built up. Next, we introduce their instantiations one by one.

C. Positional Word-to-Word Attention (PWA) Module

The PWA module corresponds to self-attention of language modality in Transformer. In this module, the input modalities X (X_{sem}, X_{pos}) and Y (Y_{sem}, Y_{pos}) in the general structure (Figure 2) are both the language modality. Specifically, referring to the positional features which are described by the position of words in a sentence, X_{pos} and Y_{pos} are represented by a learnable embedding with dimension $\mathbb{R}^{14\times d}$. For the semantic features X_{sem} and Y_{sem}, we use learnable word features to represent them in the first layer, while in the next layers, they are the output of previous PWA because PWA is stacked six times in the encoder. With the input of positional and semantic features, the raw positional attention map and raw semantic attention map are obtained respectively using multi-head attention (refering to Eq. (6)). Finally, we adopt Eq. (8) to fuse the two raw attention maps, so that the positional information can guide the semantic interaction between words in the information flow of language modality.

D. Positional Object-to-Object Attention (POA) Module

The POA module corresponds to self-attention of vision modality in Transformer. In this module, both X and Y are the vision modality. We set the features extracted by Faster RCNN as the input X_{sem} and Y_{sem} in the first layer of decoder. For other layers, X_{sem} and Y_{sem} are the output of previous layer (Figure 2). Similar to PWA, we take X_{sem} and Y_{sem} as input and use multi-head attention (Eq. (6)) to obtain the raw semantic attention map. Since bounding boxes are widely used as positional features by most state-of-the-art works in multimodal learning (e.g., VL-BERT [47], M4C [48] and Transrefer3d [49], which can achieve good results in multimodal learning, we use them to represent the positional features of images in our method.

According to the general structure, the raw positional attention map in POA is generated by going through the branch of geometry processing. The raw positional attention weight between two objects a_{ij}^G is computed as:

$$a_{ij}^G = E_G(b_i, b_j)W^G \tag{13}$$

where b_i, b_j denote the bounding box of objects i and j respectively, and E_G is a mapping function in which we describe the relative geometry features between two objects as:

$$R_{ij} = \Big(\log(\frac{|x_i - x_j|}{w_i}), \log(\frac{|y_i - y_j|}{h_i}), \log(\frac{w_j}{w_i}), \log(\frac{h_j}{h_i})\Big) \tag{14}$$

where x, y are the coordinates of the center of the bounding box, and w, h are the width and height of the bounding box. Note that R_{ij} in Eq. (14) makes the feature invariant to translation and scaling. To compute E_G, R_{ij} is embedded into a high-dimensional representation with the dimension $\mathbb{R}^{1\times d}$ using the method in [35], which calculates the sine and cosine functions of different wavelengths for absolute positional representation. Finally, we employ $W^G \in \mathbb{R}^{d\times 1}$ to map $E_G(b_i, b_j)$ to the raw positional attention map. Based on the mapping method of a *single head* between objects i and j as described in Eq. (13), we extend it to the *multi-head* version by applying Eq. (6) so that the mapping results match the raw semantic attention map with respect to the dimensions. Finally, they are fused through Eq. (8) to construct the full interaction between objects in the information flow of vision modality.

E. Positional Multi-Modality Fusion Attention (PMA) Module

The PMA module corresponds to co-attention in Transformer. This module serves to fuse the vision and language modalities, and the input modalities X, Y refer to these two different modalities respectively. Y_{sem} is the output of encoder (i.e., E in Eq. (1)) and Y_{pos} is the same as in PWA. As shown in Figure 2, PMA always follows POA, so X_{sem} is the output of the POA in the same layer. We use a different method to determine X_{pos}: given an image consisting of a set of objects, the bounding box of each object is defined as $(x_{min}, y_{min}, x_{max}, y_{max})$ which are described using pixel values. We normalize these values by the height and width of an image and then add the area of a bounding box as a new feature; i.e., the obtained feature is $(\frac{x_{min}}{W}, \frac{y_{min}}{H}, \frac{x_{max}}{W}, \frac{y_{max}}{H}, Area)$. We linearly transform this 5D feature to a high-dimensional representation, so that X_{pos} has the dimension of $\mathbb{R}^{100\times d}$.

By now, we have specified $X_{sem}, X_{pos}, Y_{sem}, Y_{pos}$. Then we use multi-head attention (referring to Eq. (6)) to get the two raw attention maps which are modulated through the intra-modality information flow. We further fuse the two raw attention maps to capture the key word-object pairs by Eq. (8) in this inter-modality information flow.

IV. Experiments

A. Datasets and Evaluation Metrics

We evaluate our transformer-like struture with proposed modules on three benchmark datasets: COCO-QA, VQA v1.0 and VQA v2.0.

COCO-QA [17] contains 123,287 images collected from the MS-COCO dataset, 78,736 training questions and 38,948 test questions. The questions are divided into four types: ”Object”, ”Number”, ”Color”, and ”Location”. All the answers are one word, and the number of total answers is 435.

VQA v1.0 [1] is built on the MS-COCO dataset. There are 3 types of questions in VQA v1.0, which are “Y/N”, “Num”, and “Other”. It has 248,349 training questions (train split), 121,512 validation questions (validation split) and 244,302 testing questions (test-standard split). Moreover, 25% of the test-standard set is collected into a test-dev split. Since the annotations of test-standard split in VQA v1.0 are not given, we have to evaluate our model by submitting the predicted answers to the official sever. Because the test-standard set has a limit on the number of submissions, we perform our ablation study on the test-dev set and conduct an overall comparison with other state-of-the-art models on the test-standard set.

VQA v2.0 [50], which is an advanced version of VQA v1.0, is the largest and most widely used dataset in the VQA community. VQA v2.0 adds an image with different answers to each question, so that the model cannot achieve high performance only by learning dataset priors. VQA v2.0 has 443,757 training questions, 214,354 validation questions and 447,793 testing questions. Similar to VQA v1.0, evaluations of

VQA models are submitted online on the developing (test-dev) and standard (test-std) subsets of testing questions.

For VQA v1.0 and VQA v2.0 datasets, we use the following evaluation metric that is robust to inter-personal differences in answer expression. In more details, each question was answered by ten people, and the answer was consider correct as long as more than three people gave the same answer as provided by the model; i.e., the score of each question is calculated by Acc(answer)=min((number of people who gave the same answer as provided by the model)/3, 1). It means that only when the predicted answer is consistent with the answer provided by at least three people, we can get a 100% score for the question. For COCO-QA dataset, we adopt the metric of classification accuracy. Moreover, following [51], we use the Wu-Palmer similarity (WUPS) in terms of WUPS@0.9 and WUPS@0.0 as another metric. This metric is used to evaluate the correlation between the predicted answers and ground-truth answers.

Since not all state-of-the-art works have conducted experiments on these three datasets, we have to choose representative works which reported experimental results on COCO-QA, VQA v1.0 and VQA v2.0 respectively for comparison. In Tables I, II and III, the performance results of existing works are cited from previous papers, and the settings of datasets are the same for all the compared works.

B. Implementation Details

We choose MCAN [7] as the baseline model for comparison, since our model replaces its core attention mechanisms by the newly proposed modules. Accordingly, our experimental settings are consistent with that used in MCAN for a fair comparison. Note that MCAN is only evaluated on VQA v2.0, while our model is evaluated on COCO-QA, VQA v1.0 and VQA 2.0.

First we use the pretrained Faster RCNN model with ResNet-101 as the backbone to perform object feature extraction in images. The dimensions of object features and question word features are 2,048 and 300 respectively. Before input into the encoder-decoder structure, the object features are linearly transformed into the general hidden size ($d = 512$), and the question is encoded as a d-dimensional feature by LSTM, where d is also the dimension for multi-head attention as described in Eq. (5). The number of heads h is 8, so the dimension for each head is $d_h = d/h = 64$. The feed forward layer shown in the Figure 3 is set to be the structure FC (512, 2048)-ReLU-Dropout (0.1)-FC (2048, 512). Both encoder and decoder have six layers.

Then we choose Adam [52] with $\beta_1 = 0.9, \beta_2 = 0.98$ as our optimization algorithm. We train our model for 13 epochs with the batch size 64. In addition, according to [35], we perform the warm up operation at the beginning of the training process. Specifically, the learning rate is set as $2.5e^{-5}, 5e^{-5}, 7.5e^{-5}$ for the first three epochs. After that, the model is trained with the learning rate $1.0e^{-4}$ for seven epochs. Finally, we adopt the learning rate decay operation, i.e., we set the learning rate of the last three epochs as $2e^{-5}, 2e^{-5}, 4e^{-6}$ respectively.

With these settings, we perform open-ended tasks on VQA v1.0 and VQA v2.0 datasets. We generate our answer vocabulary using the strategy in [46], i.e., collecting the answer that appears at least eight times in the training and validation sets. To obtain the answer on test-standard splits, three datasets including the train split, the validation split and some samples of Visual Genome [43] are used for training. We also use the VQA samples from Visual Genome dataset for training, by noting that many state-of-the-art models adopt this dataset as the augmented dataset (e.g., BAN [28] and DFAF [8]). For the COCO-QA dataset, we train our model on the train split, and perform the evaluation on the test split by collecting a total of 435 different one-word answers as the answer vocabulary.

TABLE I
COMPARISON OF OUR METHOD AND STATE-OF-THE-ART METHODS ON THE COCO-QA DATASET. W@0.9 AND W@0.0 STAND FOR WUPS@0.9 AND WUPS@0.0, RESPECTIVELY.

Model	Accuracy	Object	Number	Color	Location	W@0.9	W@0.0
VSE[1]	55.09	58.17	44.79	49.53	44.37	65.34	88.64
DPPnet [18]	61.19	-	-	-	-	70.84	90.61
SAN [22]	61.60	64.50	48.60	57.90	54.00	71.60	90.90
HieCoAtt [23]	65.40	68.00	51.00	62.90	58.80	75.10	92.00
Dual-MFA [29]	66.49	68.86	51.32	65.89	58.92	76.15	92.29
CVA [25]	67.51	69.55	50.76	68.96	59.93	76.70	92.41
MCAN [7]	68.08	69.39	54.19	71.52	60.17	77.34	92.58
ODA [26]	69.33	70.84	54.70	74.17	60.90	78.29	93.02
ALSA [30]	69.97	**71.59**	54.83	72.74	61.78	79.43	**94.15**
MCAN+PA (ours)	**70.10**	71.13	**55.97**	**74.85**	**62.07**	**79.75**	93.94

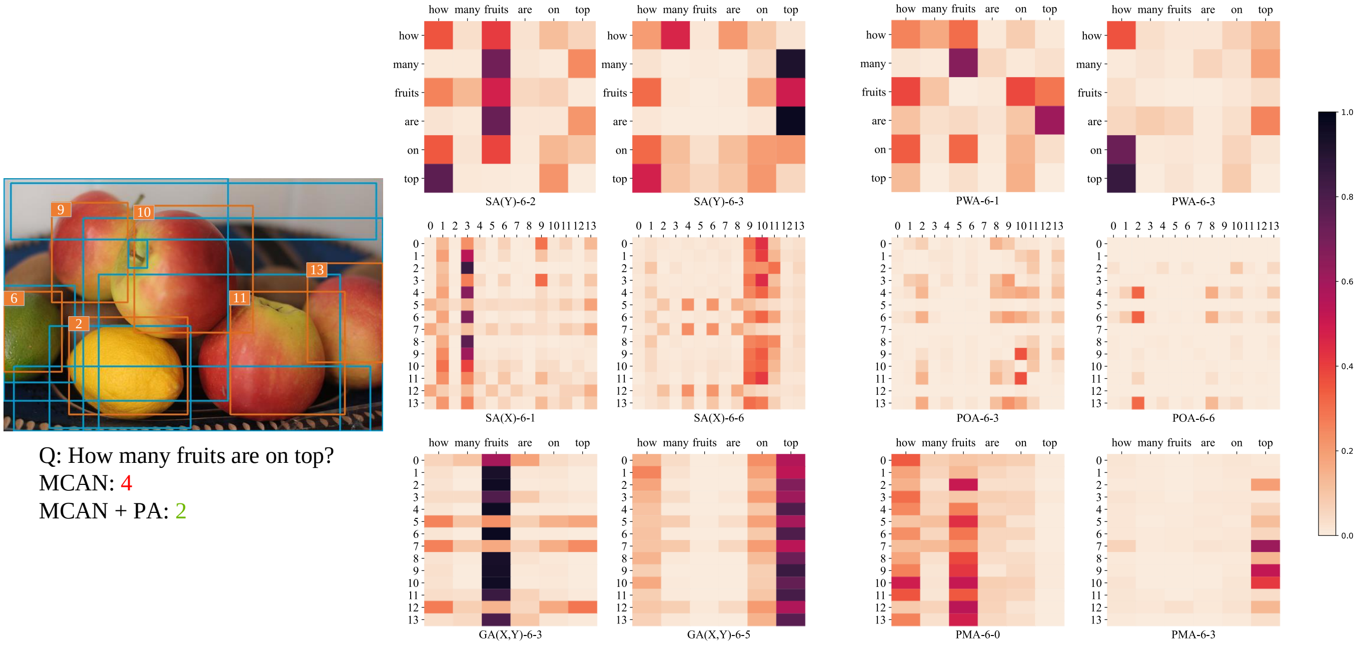
TABLE II
COMPARISON RESULTS OF OUR METHOD AND STATE-OF-THE-ART METHODS ON THE VQA V1.0 DATASET

Model	test-dev				test-std			
	Y/N	Num	Other	All	Y/N	Num	Other	All
VSE [1]	78.94	35.24	36.42	53.74	79.01	35.55	36.80	53.96
DPPnet [18]	80.71	37.24	41.69	57.22	80.28	36.92	42.24	57.36
SAN [22]	79.30	36.60	46.10	58.70	79.11	36.41	46.42	58.90
HieCoAtt [23]	79.70	38.70	51.70	61.80	-	-	-	62.10
Dual-MFA [29]	83.59	40.18	56.34	66.01	83.37	40.39	56.89	66.09
CVA [25]	83.73	40.91	56.36	65.92	83.79	40.41	56.77	66.02
ODA [26]	85.82	43.03	58.07	67.83	85.81	42.51	58.24	67.97
MCAN [7]	86.98	46.25	60.8	69.97	86.97	45.85	60.98	70.12
ALSA [30]	87.12	42.94	59.06	69.52	86.94	43.84	58.21	69.32
DenIII [27]	86.70	44.10	59.70	69.10	86.80	43.30	59.40	69.00
MCAN+PA (ours)	**87.34**	**49.92**	**61.91**	**71.05**	**87.31**	**49.46**	**62.30**	**71.27**

C. Comparison with State-of-the-art Methods

Most state-of-the-art models in VQA, such as ALSA [30], MCAN [7], DenIII [27], ODA [26], RE-ATT [32], ViLBERT [10], and MCAoA [34] are based on the attention mechanism. Here, we compare our method with them on three benchmark datasets, i.e., COCO-QA, VQA v1.0 and VQA v2.0.

Table I reports the comparison results on COCO-QA. It is observed that our method (MCAN+PA) improves the accuracy of VSE, DPPnet, SAN, HieCoAtt, Dual-MFA, MCAN, CVA



(a) MCAN (b) MCAN + PA(ours)

Fig. 4. Visualization of the learned attention map which corresponds to A in Eq. (8) for MCAN and our model (MCAN+PA), respectively. We visualize SA(Y), SA(X) and GA(X,Y) in MCAN, and PWA, POA and PMA in our method. We only visualize the attention map in the 6-th layer since it is learned in the last layer of each module, and represent the final analysis results of the model on the question. There are 14 objects detected by the Faster RCNN, and we highlight the six fruit objects in the image with orange rectangles. Detailed analysis is presented in Section IV-F.

and ODA by 15.01%, 8.91%, 8.50%, 4.70%, 3.61%, 2.02%, 2.59%, 0.77% respectively. Note that in addition to COCO-QA, ALSA also uses the VQA-HAT dataset [31], and the result of our method on COCO-QA is still competitive to its result. That is, our method outperforms ALSA in five out of seven categories. Moreover, our method significantly outperforms ALSA on VQA v1.0 test-std and VQA v2.0 test-dev respectively (see the comparison below).

It is worth noting that our proposed positional attention modules are good at handling object counting questions, as shown in the “Number” column in Table I. Our model outperforms all the previous state-of-the-art models by at least 1.14% on this category. The reason is that in many counting questions, the positional information of objects in the image is needed to get the correct counting result. For example, given the image shown in Figure 4 and the question “how many fruits are on top?”, our method can update the final composite features based on the positional information to include the correct counting information.

Table II shows the comparison results on VQA v1.0. Our method outperforms the state-of-the-art methods including VSE, DPPnet, SAN, HieCoAtt, Dual-MFA, CVA, ODA, MCAN and ALSA by 17.31%, 13.91%, 12.37%, 9.17%, 5.18%, 5.25%, 3.3%, 1.15%, 1.95% on accuracy (test-std), respectively. Although DenIII can achieve dense interactions in inter- and intra-modalities, our method outperforms it by improving 2.27% overall on VQA v1.0 test-std. Again, our method significantly outperforms the state-of-the-art methods on the object counting questions by at least 3.61%.

TABLE III
COMPARISON RESULTS OF OUR METHOD AND STATE-OF-THE-ART METHODS ON VQA V2.0 TEST DATASET.

Model	test-dev				test-std
	Y/N	Num	Other	All	All
ALSA [30]	85.73	48.98	59.17	69.21	-
BAN+Counter [28]	85.42	54.04	60.52	70.04	70.35
REGAT [40]	86.08	54.42	60.33	70.27	70.58
RE-ATT [32]	87.00	53.06	60.19	70.43	70.72
DenIII [27]	86.30	50.90	**61.50**	70.50	70.80
DFAF [8]	86.09	53.32	60.49	70.22	70.34
MCAN [7]	86.82	53.26	60.72	70.63	70.90
MCAoA [34]	**87.05**	53.81	60.97	70.90	71.14
MCAN+PA (ours)	86.99	**54.86**	61.09	**71.05**	**71.52**

Tables III and IV show the comparison results on VQA v2.0. Our method uses the same training and testing datasets with the comparison methods BAN+Counter [28], REGAT [40], RE-ATT [32], DenIII [27], DFAF [8], MCAN [7], and MCAoA [34]. As shown in Table III, our method outperforms BAN+Counter, REGAT, RE-ATT, DenIII by 1.17%, 0.94%, 0.80%, 0.72% on the test-std split, respectively. Compared to the models with single Transformer-like structure, our method outperforms MCAN by 0.62% and DFAF by 1.18% on accuracy. MCAoA is another work that improves the attention mechanism in MCAN, and our method outperforms it by 0.38%. We also compare our method with the large-scale pre-

TABLE IV
COMPARISON WITH LARGE-SCALE PRE-TRAINING MODELS ON VQA 2.0 TEST-DEV SPLIT. †MEANS WITHOUT PRE-TRAINING.

Model	test-dev
	All
VisualBERT [53]	70.80
ViLBERT [10]	70.55
VL-BERT$_{\text{BASE}}$ [47]	**71.16**
VL-BERT$_{\text{BASE}}$ [47]†	69.58
VL-BERT$_{\text{BASE}}$ [47]+PA†	69.97
MCAN+PA (ours)†	**71.05**

TABLE V
ABLATION STUDIES OF OUR PROPOSED POSITIONAL ATTENTION MODULES ON VQA V2.0 TEST-DEV SPLIT. PE IS POSITION EMBEDDING USED IN BERT, PA=PWA+PMA+POA, AND $*$ MEANS USING THE FUSION METHOD IN RELATION NETWORK [16].

Model	Y/N	Num	Other	All
MCAN	86.82	53.26	60.72	70.63
MCAN+PE	85.37	52.30	60.05	69.60
MCAN+PWA	**87.06**	53.53	60.83	70.81
MCAN+POA	86.87	54.77	60.88	70.89
MCAN+PMA	86.92	52.86	61.12	70.81
MCAN+PWA+POA	**87.06**	53.56	**61.15**	70.96
MCAN+PWA+PMA	87.01	53.33	61.07	70.87
MCAN+POA+PMA	86.91	54.24	61.03	70.92
MCAN+PA$*$	85.87	52.42	59.94	69.77
MCAN+PA (ours)	86.99	**54.86**	61.09	**71.05**

training Transformer-like models in Table IV and our method is competitive to them. These large-scale pre-training models need to be trained on other datasets in addition to VQA v2.0 and require substantial computational resources for generic visual–linguistic representations, and thus it is actually unfair to directly compare with them. Anyway, as shown in Table IV, our method still outperforms VisualBERT [53] and ViLBERT [10]. Although our model performs slightly worse than VL-BERT$_{\text{BASE}}$ [47] (similar setting as our model) with pre-training, our model still outperforms VL-BERT$_{\text{BASE}}$ without pre-training by 1.47%. Furthermore, our method can also be assembled into the large-scale pre-training Transformer-like models because their basic modules are consistent with MCAN. It can be observed in Table IV that the accuracy is improved from 69.58% to 69.97% after assembling PA (referring to our three PWA, POA, and PMA modules) into VL-BERT$_{\text{BASE}}$ without pre-training. Note that VL-BERT has two versions, i.e., base (VL-BERT$_{\text{BASE}}$) and large (VL-BERT$_{\text{LARGE}}$) versions. Both versions have the same model structure and only differ in quantity of parameters.

In addition, the results in Table III show that our model is consistently good at handling object counting questions on VQA v2.0. Although BAN+Counter [28] introduces a special counting module [6] to improve the object-counting performance, our model achieves an accuracy improvement of 0.82% in the “Num” category. Meanwhile, although REGAT [40] specializes in using geometric relationships to build graphs between objects, our model still achieves an accuracy improvement of 0.44% in the “Num” category.

D. Ablation Studies

In this section, we examine the effectiveness of proposed positional attention modules, i.e., PWA, POA and PMA modules, with respect to the baseline MCAN. We follow [8], [7], [34] to conduct ablation studies on VQA v2.0 test-dev split. The results are summarized in Table V, in which the notations are explained as follows. PE denotes the positional embedding method in BERT [13], which is a typical method for processing positional information in the large-scale pre-training models. PA denotes that the three attention mechanisms of MCAN are replaced with our PWA, POA and PMA modules. MCAN+PA$*$ denotes that using $A_i = \text{softmax}(A_i^{pos} \cdot \exp(A_i^{sem}))$ in [16] instead of Eq. (8) in our method to merge the raw positional attention map and the raw semantic attention map.

The results in Table V are analyzed as follows. First, we show that it is not efficient to directly use PE in MCAN for enhancing positional information processing. That is, we change the input of MCAN from semantic features to the combination of semantic and positional features, and the performance of MCAN+PE decreases compared with MCAN. Similar effects have been reported in previous Transformer-like architectures, such as DFAF [8]. On the contrary, our method (MCAN+PA) with the newly designed positional attention modules can clearly improve the performance of MCAN.

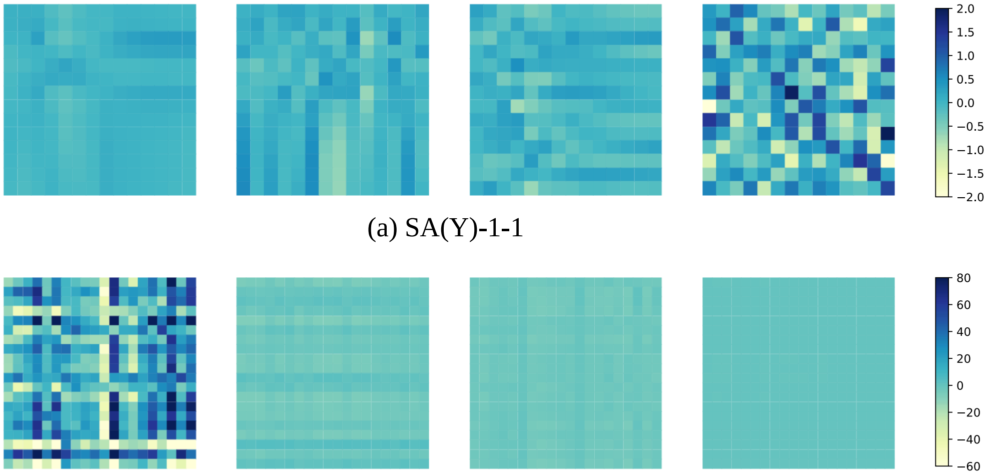
Second, we evaluate the effectiveness of the three positional attention modules. The results in Table V show that they have different effects on different types of questions. For example, POA improves the model performance on the ”Num” category, PWA has the best performance in ”Y/N” questions, and the single PMA module helps to improve the performance in the ”other” category. Meanwhile, we combine any two of the three modules and evaluate their performance, i.e., MCAN+PWA+POA, MCAN+PWA+PMA and MCAN+POA+PMA. The results show that they can improve the performance on some categories such as ”Num”, ”Y/N” and ”Other”.

Finally, we note that when we adopt the fusion method in [16] to merge the positional and semantic attention maps (i.e., MCAN+PA$*$ in Table V), the accuracy of each question category decreases compared with the baseline MCAN. That means it is not effective to directly apply the fusion method in relation network [16] in our model. On the contrast, we design new fusion strategy in our positional attention modules (i.e., MCAN+PA) that effectively improves the performance of the baseline MCAN.

E. Analysis of Time and Space Complexities

In this section, we briefly summarize the number of parameters, floating-point operations per second (FLOPs) and inference time in our model.

Compared to the baseline MCAN, our model mainly introduces new parameters in $W_G^{query} \in \mathbb{R}^{d\times d}$ and $W_G^{key} \in \mathbb{R}^{d\times d}$ which are projection matrices for positional embedding in PWA and PMA modules. There are six layers built up in



(a) SA(Y)-1-1

(b) SA(X)-1-1

Fig. 5. Visualization of the four items (Eq. (15) on a well trained MCAN+PE model for a sampled question-image pair. SA(Y)-1-1 and SA(X)-1-1 denote the first layer, second head in SA(Y) and SA(X) respectively. From left to right, these attention maps are semantic-to-semantic, semantic-to-positional, positional-to-semantic, and positional-to-positional correlation matrices.

TABLE VI
COMPARISON OF TIME AND SPACE COMPLEXITIES.

Model	Time	Params	FLOPs	Inference per image
MCAN (baseline)	4150s	57.8M	5.35G	2.561s
MCAN+PA (ours)	5583s	68.5M	5.85G	2.672s

PWA and PMA modules, and then the number of introduced parameters is 6M ($6 \times 4 \times 512 \times 512$, $d = 512$). In the POA module, we do not employ multi-head attention for positional embedding, and thus the number of introduced parameters can be ignored. As summarized in Table VI, our method introduces 18.5% additional parameters (see 'Params'), but the computation complexity (see 'FLOPs') only increases by 9.3%. In addition, the inference time of our method is similar on each image compared with MCAN.

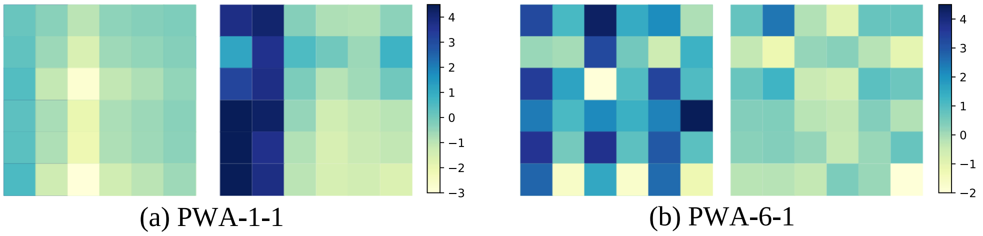
F. Qualitative Analysis

The object counting question is one of the most challenging question types in VQA tasks. We choose such a question for qualitative analysis in this section. Figure 4 visualizes the learned attention maps in both MCAN and our method (MCAN+PA), with the following notations. SA(Y), SA(X) and GA(X,Y) are the modules in MCAN, which corresponds to PWA, POA and PMA in MCAN+PA respectively. SA(Y)-6-2 denotes the attention map of the 2-th head of the 6-th layer in SA(Y) module, and and other similar notations have the same naming rules. Due to space limitation of the figure, we only show two typical heads in each module. We have the following observations:

Comparison between PWA and SA(Y). Both PWA and SA(Y) can catch the key words. For example, in the third row (i.e., the "fruits" row) of PWA-6-1, the columns of "how", "many", "on" and "top" contain the largest values, indicating that the question type is object counting and the model should find the number of fruits on top. Similarly, as shown in the third row of SA(Y)-6-2, the word "fruits" has the largest correlations with "how", "many" and "fruits". SA(Y)-6-3 also shows that "fruits" also has strong correlations with "how", "on" and "top" in the MCAN model.

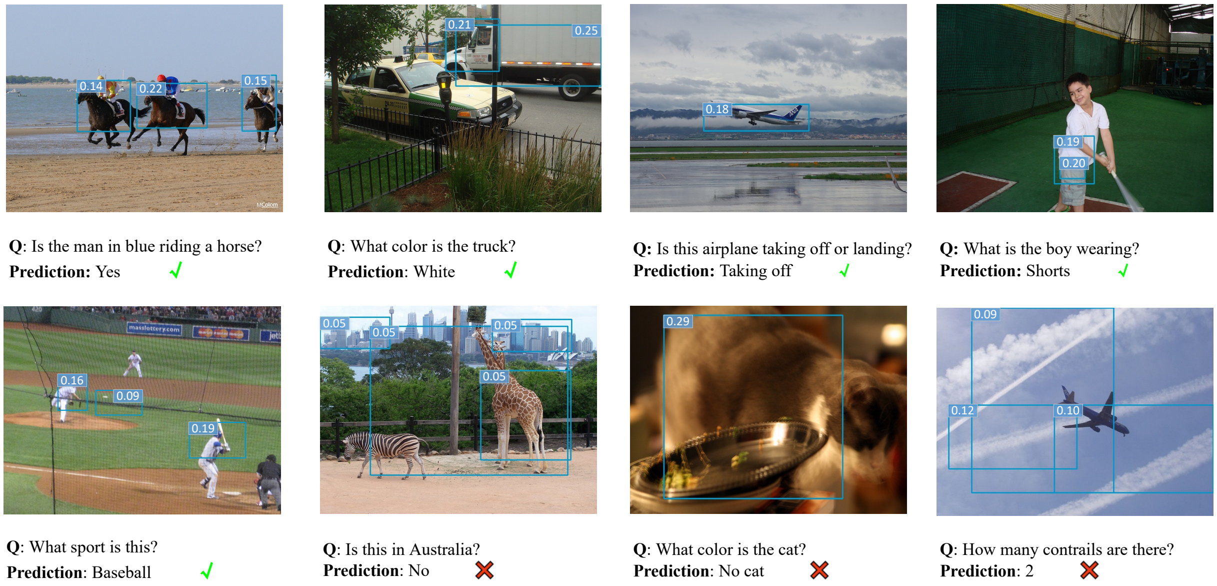
Comparison between POA and SA(X). SA(X) can catch the correlation between objects but possibly introduce some noise. For example, as shown in SA(X)-6-1, many objects have large correlation values with the object 3. However, the bounding box of the object 3 overlaps with the fruit objects 2, 10 and 11, and catching these redundant relations is not helpful to answer the question. As a contrast, in our model MCAN+PA, both POA-6-3 and POA-6-6 show that few objects have relations with the object 3 due to the introduced positional information. Similarly, as shown in the 2-th, 4-th, 6-th, 9-th, 10-th, 11-th, 13-th rows of POA-6-3 and the 2-th, 4-th, 6-th, 13-th rows of POA-6-6, MCAN+PA can catch the key relations very well.

Comparison between PMA and GA(X,Y). The results shown in the "fruits" column of GA(X,Y)-6-3 and in the "top" column of GA(X,Y)-6-5 explain why MCAN provides the wrong answer. Although MCAN can focus on the fruit objects as indicated in the "fruits" column, MCAN fails to identify which bounding box is at the top of the image due to the lack of the positional information. As shown in GA(X,Y)-6-5, many objects have strong correlation with the word "top" but most of them are not actually on the top of the image. As a contrast, in PMA-6-3, only the objects 7, 9 and 10 have strong correlation with the word "top", which is more consistent with the fact. Meanwhile, although the object 7 has strong correlation with the word "top", it does not affect the final result because it does not generate strong correlation with the word "fruits" (as shown in the "fruits" column of PMA-6-0). In summary, with the help of the introduced positional information, MCAN+PA successfully identifies which objects



(a) PWA-1-1 (b) PWA-6-1

Fig. 6. Visualization of the raw attention map which corresponds to A_i^{sem} and A_i^{pos} in Eq. (8) for PWA in our model (MCAN+PA). In both (a) and (b), the left and the right patterns are A_i^{sem} and A_i^{pos}, respectively. We observe that they are in the same level of magnitude so that they would affect each other when determining the attention map corresponding to A_i in Eq. (8).



Q: Is the man in blue riding a horse?
Prediction: Yes ✓

Q: What color is the truck?
Prediction: White ✓

Q: Is this airplane taking off or landing?
Prediction: Taking off ✓

Q: What is the boy wearing?
Prediction: Shorts ✓

Q: What sport is this?
Prediction: Baseball ✓

Q: Is this in Australia?
Prediction: No ✗

Q: What color is the cat?
Prediction: No cat ✗

Q: How many contrails are there?
Prediction: 2 ✗

Fig. 7. Visualization of the learned attention map corresponding to Eq.(1) in some typical examples. For each example, "Q" represents the question, and "prediction" is the answer provided by our model. We use rectangles to specify several objects with large attention weights, and these weights are annotated in the image. In addition, underlined words indicate that they are captured by our model as keywords.

are related to the word "top" (see PMA-6-0), and recognizes the fruit objects in the image through semantic information (see PMA-6-3). Based on these two abilities, MCAN+PA correctly provides the answer "2", which denotes the fruit objects 9 and 10.

Furthermore, to explore why the PE method cannot be directly applied to MCAN (see MCAN+PE in Table V) and explain why our method works well, we visualize the attention maps of Eq. (15) and Eq. (8) in Figures 5 and 6, respectively. In these figures, the darker the color on the attention map, the more it affects the model. From Figures 5 and 6, we can conclude that MCAN+PE does not make good use of semantic information in the language modality and positional information in the vision modality, while our model uses both types of information very well. The detail analysis is as following.

According to [14], the attention map of the first layer in MCAN+PE can be written as:

$$A_{ij} = f(S_iW^Q, S_jW^K) + f(S_iW^Q, P_jW^K) + f(P_iW^Q, S_jW^K) + f(P_iW^Q, P_jW^K) \tag{15}$$

where A_{ij} denotes the obtained attention map of i modality guided by j modality, f is the function described in Eq. (5), S and P stands for the semantic features and positional features, respectively.

In Figure 5, four correlation matrices for language and vision modalities are visualized as heatmaps using the same range of values. We can observe that $f(P_iW^Q, P_jW^K)$ (i.e., the last column of SA(Y)-1-1 in Figure 5) and $f(S_iW^Q, S_jW^K)$ (i.e., the first column of SA(X)-1-1 in Figure 5) dominate the obtained attention maps of SA(Y)-1-1 and SA(X)-1-1 respectively. However, for SA(Y)-1-1, the semantic information of words cannot be ignored. MCAN+PE uses the positional-positional correlation matrix to determine the relations between words. Meanwhile in SA(X)-1-1, MCAN+PE only uses the semantic-semantic correlation matrix to determine the result, and thus the introduction of positional information becomes meaningless. That fact causes the performance of MCAN+PE to degrade.

In contrast, our proposed MCAN+PA handles such problem well. As shown in Figure 6, we visualize two items in PWA-1-

1 and PWA-6-1 that correspond to A_i^{sem} and A_i^{pos} in Eq. (8), respectively. It is observed that both A_i^{sem} and A_i^{pos} contribute to the obtained attention map A_i. We only show the first and last layers of PWA in Figure 6, but in all six layers, both A_i^{sem} and A_i^{pos} contribute to the obtained attention map A_i.

In Figure 7, we visualize some of the final obtained attention maps learned by Eq.(1). The results show that our model can capture the correct objects and keywords regardless of the correct prediction samples or the wrong prediction samples. It is also observed that there are some difficult questions that our model cannot handle. For example, in the bottom right image, the cloud and contrail are mixed together, making it difficult to distinguish them. In addition, our model sometimes fails when the answer to the question requires external knowledge (see the second example of the second row) or when the image is blur (see the third example of the second row).

V. Conclusion

In this paper, we propose an efficient positional attention guided Transformer-like architecture for VQA tasks. We allow positional information to be passed between and across language and vision modalities, so that it can guide high-level interaction of semantic attentions in information flow of intra- and inter-modalities. Specifically, we propose three novel positional modules, i.e., PWA, POA and PMA, and they are compatible with Transformer-like models in VQA. Experimental results on COCO-QA, VQA v1.0 and VQA v2.0 datasets show that our model outperforms state-of-the-art methods, epspeically in handling object counting questions.

One limitation of our method is introducing additional computational costs compared to the baseline model MCAN. In the future work, we plan to optimize the model to reduce training time without degrading performance. We will also consider to explore the relative positional representation for possible improvement of model performance.

Acknowledgment

We thank the anonymous reviewers for their constructive comments that help us to improve the paper. This work was supported in part by the NSF of Guangdong Province (2019A1515010833, 2022A1515011573), in part by the NSF of China (61725204) and Tsinghua University Initiative Scientific Research Program (20211080093).

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2425–2433.

[2] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6077–6086.

[3] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked cross attention for image-text matching," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 201–216.

[4] W. Guan, H. Wen, X. Song, C.-H. Yeh, X. Chang, and L. Nie, "Multimodal compatibility modeling via exploring the consistent and complementary correlations," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 2299–2307.

[5] X. Huang, Y. Peng, and Z. Wen, "Visual-textual hybrid sequence matching for joint reasoning," *IEEE Transactions on Cybernetics*, 2020.

[6] Y. Zhang, J. Hare, and A. Prügel-Bennett, "Learning to count objects in natural images for visual question answering," in *International Conference on Learning Representations*, 2018.

[7] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6281–6290.

[8] P. Gao, Z. Jiang, H. You, P. Lu, S. C. Hoi, X. Wang, and H. Li, "Dynamic fusion with intra-and inter-modality attention flow for visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6639–6648.

[9] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "Vl-bert: Pre-training of generic visual-linguistic representations," in *International Conference on Learning Representations*, 2020.

[10] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *Advances in Neural Information Processing Systems*, 2019, pp. 13–23.

[11] H. Tan and M. Bansal, "Lxmert: Learning cross-modality encoder representations from transformers," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 5100–5111.

[12] M. A. Islam, S. Jia, and N. D. Bruce, "How much position information do convolutional neural networks encode?" in *International Conference on Learning Representations*, 2020.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[14] G. Ke, D. He, and T.-Y. Liu, "Rethinking positional encoding in language pre-training," in *International Conference on Learning Representations*, 2020.

[15] X. Liu, Z. Ji, Y. Pang, J. Han, and X. Li, "Dgig-net: Dynamic graph-in-graph networks for few-shot human-object interaction," *IEEE Transactions on Cybernetics*, 2021.

[16] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3588–3597.

[17] M. Ren, R. Kiros, and R. Zemel, "Exploring models and data for image question answering," *Advances in Neural Information Processing Systems*, vol. 28, pp. 2953–2961, 2015.

[18] H. Noh, P. H. Seo, and B. Han, "Image question answering using convolutional neural network with dynamic parameter prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 30–38.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[20] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Deep compositional question answering with neural module networks. corr abs/1511.02799 (2015)," *arXiv preprint arXiv:1511.02799*, 2015.

[21] Z. Huasong, J. Chen, C. Shen, H. Zhang, J. Huang, and X.-S. Hua, "Self-adaptive neural module transformer for visual question answering," *IEEE Transactions on Multimedia*, 2020.

[22] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 21–29.

[23] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 289–297.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing systems*, 2015, pp. 91–99.

[25] J. Song, P. Zeng, L. Gao, and H. T. Shen, "From pixels to objects: Cubic visual attention for visual question answering." in *IJCAI*, 2018, pp. 906–912.

[26] C. Wu, J. Liu, X. Wang, and X. Dong, "Object-difference attention: A simple relational attention for visual question answering," in *Proceedings of the 26th ACM Tnternational Conference on Multimedia*, 2018, pp. 519–527.

[27] F. Liu, J. Liu, Z. Fang, R. Hong, and H. Lu, "Visual question answering with dense inter-and intra-modality interactions," *IEEE Transactions on Multimedia*, 2020.

[28] J.-H. Kim, J. Jun, and B.-T. Zhang, "Bilinear attention networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 1564–1574.

[29] P. Lu, H. Li, W. Zhang, J. Wang, and X. Wang, "Co-attending free-form regions and detections with multi-modal multiplicative feature embedding for visual question answering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[30] Y. Liu, X. Zhang, Z. Zhao, B. Zhang, L. Cheng, and Z. Li, "Alsa: Adversarial learning of supervised attentions for visual question answering," *IEEE Transactions on Cybernetics*, 2020.

[31] A. Das, H. Agrawal, L. Zitnick, D. Parikh, and D. Batra, "Human attention in visual question answering: Do humans and deep networks look at the same regions?" *Computer Vision and Image Understanding*, vol. 163, pp. 90–100, 2017.

[32] W. Guo, Y. Zhang, X. Wu, J. Yang, X. Cai, and X. Yuan, "Re-attention for visual question answering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 91–98.

[33] Y. Liu, Y. Guo, J. Yin, X. Song, W. Liu, L. Nie, and M. Zhang, "Answer questions with right image regions: A visual attention regularization approach," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 18, no. 4, pp. 1–18, 2022.

[34] T. Rahman, S.-H. Chou, L. Sigal, and G. Carenini, "An improved attention for visual question answering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1653–1662.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing systems*, 2017, pp. 5998–6008.

[36] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 464–468.

[37] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[38] B. Wang, D. Zhao, C. Lioma, Q. Li, P. Zhang, and J. G. Simonsen, "Encoding word order in complex embeddings," in *International Conference on Learning Representations*, 2020.

[39] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[40] L. Li, Z. Gan, Y. Cheng, and J. Liu, "Relation-aware graph attention network for visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 10 313–10 322.

[41] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[42] Q. Huang, J. Wei, Y. Cai, C. Zheng, J. Chen, H.-f. Leung, and Q. Li, "Aligned dual channel graph convolutional network for visual question answering," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7166–7176.

[43] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[44] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[45] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[46] D. Teney, P. Anderson, X. He, and A. Van Den Hengel, "Tips and tricks for visual question answering: Learnings from the 2017 challenge," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4223–4232.

[47] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "Vl-bert: Pre-training of generic visual-linguistic representations," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=SygXPaEYvH

[48] R. Hu, A. Singh, T. Darrell, and M. Rohrbach, "Iterative answer prediction with pointer-augmented multimodal transformers for textvqa," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9992–10 002.

[49] D. He, Y. Zhao, J. Luo, T. Hui, S. Huang, A. Zhang, and S. Liu, "Transrefer3d: Entity-and-relation aware transformer for fine-grained 3d visual grounding," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 2344–2352.

[50] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6904–6913.

[51] M. Malinowski and M. Fritz, "A multi-world approach to question answering about real-world scenes based on uncertain input," *arXiv preprint arXiv:1410.0210*, 2014.

[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[53] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "Visualbert: A simple and performant baseline for vision and language," *arXiv preprint arXiv:1908.03557*, 2019.

Aihua Mao is a professor with the School of Computer Science and Engineering, South China University of Technology (SCUT), China, He received the PhD degree from the Hong Kong Polytechnic University in 2009, the M.Sc degree from Sun Yat-Sen University in 2005 and the B.Eng degree from Hunan University in 2002. His research interests include image learning and computer graphics.

Zhi Yang received the B.S. degree in software engineering from Fuzhou University, Fuzhou,China in 2020. He is currently pursuing the M.S. degree in computer science with South China University of Technology, Guangzhou, China. His current research interest is multimodal learning.

Ken Lin received the B.S. degree in mechanical engineering from Beijing Jiaotong University, Beijing, China in 2019. He is currently pursuing the M.S. degree in computer science with South China University of Technology, Guangzhou, China. His current research interest is multimedia learning.

Jun Xuan is a postgraduate with the School of Computer Science and Engineering, South China University of Technology (SCUT), China, He received the B.Eng degree from Guangdong University of Technology in 2020. His research interests include 3D vision and computer graphics.

Yong-Jin Liu is a professor with the Department of Computer Science and Technology, Tsinghua University, China. He received the BEng degree from Tianjin University, China, in 1998, and the PhD degree from the Hong Kong University of Science and Technology, Hong Kong, China, in 2004. His research interests include computer vision, computer graphics, pattern analysis and affective computing.