Bilinear Graph Networks for Visual Question Answering

Dalu Guo\textsuperscript{\circledast}, Chang Xu\textsuperscript{\circledast}, Member, IEEE, and Dacheng Tao\textsuperscript{\circledast}, Fellow, IEEE

Abstract—This article revisits the bilinear attention networks (BANs) in the visual question answering task from a graph perspective. The classical BANs build a bilinear attention map to extract the joint representation of words in the question and objects in the image but lack fully exploring the relationship between words for complex reasoning. In contrast, we develop bilinear graph networks to model the context of the joint embeddings of words and objects. Two kinds of graphs are investigated, namely, image-graph and question-graph. The image-graph transfers features of the detected objects to their related query words, enabling the output nodes to have both semantic and factual information. The question-graph exchanges information between these output nodes from image-graph to amplify the implicit yet important relationship between objects. These two kinds of graphs cooperate with each other, and thus, our resulting model can build the relationship and dependency between objects, which leads to the realization of multistep reasoning. Experimental results on the VQA v2.0 validation dataset demonstrate the ability of our method to handle complex questions. On the test-set, our best single model achieves state-of-the-art performance, boosting the overall accuracy to 72.56\%, and we are one of the top-two entries in the VQA Challenge 2020.

Index Terms—Bilinear graph, deep learning, graph neural networks (GNNs), visual question answering (VQA).

I. INTRODUCTION

T HE developments in computer vision and natural language processing enable the machine to deal with complicated tasks that require the integration and understanding of vision and language, e.g., image captioning [1], visual grounding [2], [3], visual question answering (VQA) [4]–[6], and visual dialogue [7], [8]. Compared with image captioning that is to simply describe the topic of an image, VQA needs a complex reasoning process to infer the right answers for a variety of questions. Visual grounding aims to locate the related objects in the image, but VQA takes a further step to convert this information into human language. In addition, VQA is the basic and vital component in a visual dialogue. Considering the challenges and significance of VQA, increasing research attention has been attracted to it.

Given an input image and a question, representative VQA models, e.g., stacked attention networks (SANs) [9], multimodal compact bilinear pooling (MCB) [3], and multimodal low-rank bilinear attention networks (MLBs) [10], first generate grid image features by ResNet [11] and represent the question as the last hidden state of long short-term memory (LSTM) [12] and then attend to the image features based on the question vector to ground the target objects; the question vector and the weighted image features are finally projected into a unified embedding for answer prediction. Bilinear attention networks (BANs) [13] consider the interaction between words in the question and objects in the image and propose to build a bilinear coattention map considering each pair of multimodal channels. Moreover, dynamic fusion with intramodality and intermodality (DFAF) [5] and deep modular coattention networks (MCANs) [6] consider intraattention within each modality and interattention across different modalities by the scaled dot-product attention from Transformer [14] and BERT [15]. Furthermore, the BERT-like models, such as Vilbert [16], VL-bert [17], and LXMERT [18], pretrain the BERT model on a large number of out-domain data (image caption and GQA [19]) and then fine-tune it on the VQA task.

However, BAN lacks comprehensive exploitation of the interactions between words in questions for modeling their context. The linear way of using scaled dot-product is mainly to calculate the attention within a single modality (the queries, keys, and values come from the same kind of nodes), such as textual nodes [14], [15] and visual nodes [20], [21], so it is less expressive to fully capture the complex relationship within the multimodal inputs.

In this article, we develop bilinear graph networks (BGNs) for VQA. We first investigate the bilinear attention map between words in the question and objects in the image from a new graph perspective. Then, we highlight the importance of exploiting the intramodality relationship between words in the question and exploring the cross-modality relationship between the question and image for complex reasoning. Two graphs are established to formulate these two kinds of relationships. The image-graph focuses on exploring visual features of the image to their related textual features for the joint embeddings, which links the semantic information of words with the factual information detected from the image. The question-graph exploits information across different joint embeddings in terms of words, which amplifies the implicit
yet important relationships between objects. Given these two graphs complementing each other, the resulting VQA model is able to reason complex and compositional questions.

We conduct experiments on the VQA v2.0 dataset [22]. On the validation dataset, our one-layer graph networks boost the accuracy by 0.62% compared with BAN. Moreover, the graphs of multiple layers show advantages on multistep reasoning for long and complex questions, evidenced by a total 1.4% improvement. With help of the pretrained language model as our question encoder, our graphs gain an extra 1.5% increase. On the test-std dataset, our model achieves state-of-the-art performance without visual-language pretraining, increasing the overall accuracy to 72.56%. By ensembling our graph models with different layers, we got second place in the VQA Challenge 2020.

II. RELATED WORK

In this section, we will first introduce the related research on VQA and then the graph neural networks (GNNs) on both text- and visual-based tasks.

A. Visual Question Answering

VQA is a task to answer the given question based on the input image. The question is usually embedded into a vector with LSTM [12], and the image is represented by the fixed-size grid features extracted from a pretrained model, such as ResNet [11]. Then, both of these features are combined by addition or concatenation [4] before being projected into a unified vector for answering prediction through a multilayer perceptron (MLP). However, not all features of the image are related to the given question, while some of them should be filtered out before generating the unified vector. Therefore, an attention mechanism is introduced to learn the weight of each grid feature. Stack attention networks (SANs) [9] learn visual attention through multisteps, trying to answer the question progressively. Dual attention networks (DANs) [23] learn visual and textual attention, respectively, via the memory vector.

Due to the different distributions of question and image features, the outer product of both features has a better explanation and performance compared with the linear combination. However, because of its high-dimensional output, it is hard to be optimized. MCB [3] is approaching this process by calculating the count sketch of two features and convolving them in faster Fourier transform (FFT) space. Nevertheless, MCB uses sampling features instead of the original ones, which leads to bias and needs a large projected dimension to reduce it. The Hadamard product for low-rank bilinear pooling (MLB) [10] models the common vector with a low-rank matrix by an elementwise multiplication, and multimodal factorized bilinear pooling (MFB) [24] increases the rank from 1 to \(k\) to accelerate the convergence rate and improve the model’s robustness. Furthermore, BANs [13] learn the textual and visual attention simultaneously, which builds a mapping from the detected objects of the image to the words of the question.

Bottom-up and top-down (BUTD) [1] focus on bottom-up attention of image features and propose a set of salient image regions with natural expression detected by Faster-RCNN [25]. Furthermore, its training set contains 1600 object classes from visual genome [26], larger than the original 80 object classes from COCO [27], and it also needs to predict the attributes for the detected objects, such as their color and shape. In Defense of Grid Features (GridFeat), Jiang et al. [28] advocate the \(1 \times 1\) ROI pooling to speed up extracting the image features while keeping the accuracy.

B. Graph Neural Network

GNN [29] is used to build the relationship between nodes, such as citation link [30], knowledge graph [31], action recognition [32], and protein–protein interaction [33]. It overcomes the limitation of Euclidean distance between each node in the inputs and involves more context information from neighbors. In text-based tasks, such as machine translation and sequence tagging, GNN breaks the sequence restriction between each word and learns the graph weight by attention mechanism, such as Transformer [14]. Since each node is directly linked with others via learned weights instead of through hidden states and gates, it is easier to model longer sequences than LSTM and gated recurrent units (GRUs) [34]. Pretraining of Deep bidirectional transformers (BERTs) [15], which is trained on a large corpus with unsupervised learning approaches, can be easily explained and transferred to other tasks, such as question answering [35] and commonsense inference [36]. In image-based tasks, GNN gathers information from all the grids [20], [37] or proposals [38] other than surroundings whose size is limited by the receptive fields of convolution neural networks (CNNs), and it aggregates features over coordinate space to improve the performance of object detection and scene generation [21].

In VQA tasks, DFAF [5] and MCAN [6] consider all the relationships between inputs by calculating the attention weight with scaled dot-product, including word and word, object and object, and object and word. Relation-aware Graph Attention Network (ReGAT) [39] involves the explicit relationship between objects, including spatial and semantic graphs. Out of the Box [40] builds a relation graph based on the retrieved triplet (visual concept, relation, and attribute) from FVQA [41], and it regards the visual concept and attributes as nodes and relationships as links to exchange information. Given a set of atomic operations, MMnasNets [42] identifies the optimal architectures by searching for the composition of these operations. Moreover, these BERT-like models, such as Vilbert [16], VL-bert [17], and LXMERT [18], even fine-tune the BERT model by reconstructing the masked image region categories and words and predicting the alignment of the image and its caption. Similar to other BERT-like models, UNITER [43] requires much out-of-domain data for the pretraining, but it involves the conditional masking and a novel word-region alignment on the pretraining tasks; 12-in-1 [44] pretrains a single model on 12 datasets, which significantly reduces the parameters to be trained while improving the performance on all tasks at the same time. DeVLBert [45] targets
at the out-of-domain visual-linguistic pretraining. It introduces the idea of backdoor adjustment from the causality field into the Bert-style out-of-domain pretraining. However, all of them rely on a large number of image caption data, such as COCO [27], GQA [19], and conceptual captions [46].

III. PRELIMINARIES

The goal of VQA task is to answer the given question \( T \) based on the input image \( I \). With the object-detector Faster-RCNN [1], [25], we convert the input image \( I \) into object features \( V = (v_1, \ldots, v_n) \) with \( v_i \in \mathbb{R}^d \), where \( n \) is the number of detected objects and \( D \) is the feature dimension. The question \( (q_1, \ldots, q_m) \) is a sequence of \( m \) words. It can be encoded using either LSTM [12] or Transformer [14], [15] to \( Q = \text{Transformer}(T) \) or \( Q = \text{LSTM}(T) \), \( Q \in \mathbb{R}^{C \times m} \), and \( C \) is the dimension of output features.

In order to represent the common vector of \( v \in V \) and \( q \in Q \), a weight matrix \( W_i \) is introduced to calculate the scalar output \( f_i \) and can be approximated with multiplication of two submatrices \( U_iV_i^\top \) following MLB [10] (bias terms are omitted without loss of generality)

\[
f_i = q^\top W_iq \approx q^\top U_i V_i^\top q = \mathbb{I}^\top (U_i q \circ V_i^\top v) \tag{1}
\]

where \( W_i \in \mathbb{R}^{C \times D} \), \( U_i \in \mathbb{R}^{C \times d} \), \( V_i \in \mathbb{R}^{D \times d} \), \( \mathbb{I} \in \mathbb{R}^d \) is a vector with all elements equal to 1, and \( \circ \) is the Hadamard product (elementwise multiplication). This decomposition makes the rank of matrix \( W_i \) to be at most \( d \leq \min(C, D) \). To obtain the output feature \( f \in \mathbb{R}^K \), two \( D \)-Tensors, \( U \in \mathbb{R}^{C \times d \times K} \) and \( V \in \mathbb{R}^{D \times d \times K} \), are learned, and empirically, \( d \) is set to 1 [10], [13], resulting in \( U \in \mathbb{R}^{C \times K} \) and \( V \in \mathbb{R}^{D \times K} \) for simplicity.

However, the question features \( Q \in \mathbb{R}^{C \times m} \) and image features \( V \in \mathbb{R}^{D \times n} \) are multiple channels. BAN [13] reduces both input channels simultaneously and obtains a unified representation of them. It first calculates a bilinear attention map \( G \in \mathbb{R}^{m \times n} \) between \( Q \) and \( V \), conditioned on which it then generates the joint embedding \( z \) as follows:

\[
z = \text{BAN}(Q, V; G) \tag{2}
\]

The attention map \( G \) is defined as

\[
G = \text{softmax}((\mathbb{I} \cdot p^\top) \circ \sigma(U^\top q)) \sigma(V^\top v) \tag{3}
\]

where \( U \in \mathbb{R}^{C \times K} \), \( V \in \mathbb{R}^{D \times K} \), and \( p \in \mathbb{R}^K \) are variables to be learned, \( K \) denotes the shared embedding size, and \( \sigma \) is the ReLU activation function denoted as \( \sigma(x) = \max(x, 0) \). Notice that the softmax function works on the rows and columns, i.e., \( \sum_{i=1}^{m} \sum_{j=1}^{n} G_{i,j} = 1 \). The logit \( G_{i,j} \), element of \( G \) before softmax, is the output of low-rank bilinear pooling as

\[
G_{i,j}^* = p^\top (\sigma(U^\top q_i) \circ \sigma(V^\top v_j)) \tag{4}
\]

The matrix \( p^\top \) projects the unified vector of \( q_i \) and \( v_j \) into a scalar to represent the relationship between them.

Then, the \( k \)-th element value of joint embedding \( z \in \mathbb{R}^K \) is given by

\[
z_k = \sum_{i=1}^{m} \sum_{j=1}^{n} G_{i,j} \sigma(q_i^\top U_k) \sigma(V_k^\top v_j) \tag{5}
\]

where \( U \in \mathbb{R}^{C \times K} \) and \( V \in \mathbb{R}^{D \times K} \) are the parameters to be optimized. It can also be rewritten as

\[
z_k = \sigma(Q^\top U_k)^\top G \sigma(V^\top V_k) \tag{6}
\]

where \( (Q^\top U_k) \in \mathbb{R}^m \) is the \( k \)-th column of \( Q^\top U \), and \( (V^\top V_k) \in \mathbb{R}^n \) is the \( k \)-th column of \( V^\top V \).

After that, the joint embedding \( z \) is passed to a classifier, such as MLP, to calculate the score \( p_i \) for answer \( a_i \in A \) and choose the highest one as the predicted answer, where \( A \) is the candidate answers.

IV. BILINEAR GRAPH NETWORKS

In this section, we start with interpreting BAN from a graph view and detail the differences between our graphs and other similar methods; then, we introduce the proposed image-graph and question-graph.

The graph attention network and its variant, Transformer, are efficient in modeling the relationship within single modality, such as textual nodes [14], [15], visual nodes [20], [21], and citation nodes [33], whose outputs can be calculated as

\[
\text{Tr}(Q, K, V) = \text{softmax}(Q^\top K^\top V) \tag{7}
\]

where \( Q, K, \) and \( V \) denote the queries, keys, and values, respectively, and the softmax function only works on the rows. Motivated by (7), we can easily illustrate BAN from the perspective of graph.

Given the calculation of \( z_k \) in (6), (2) can be reformulated as

\[
Z^\top = \text{BGN}(Q, V; G) = \sigma(Q^\top U) \circ \sigma(V^\top V) \tag{8}
\]

where \( Z^\top = (z_1^\top, \ldots, z_m^\top) \) are calculated based on the input nodes \( Q \cup V \) and their attention weight \( G^a \). The attention map \( G^a \) in (8) is equivalent to the graph weight softmax \( \text{softmax}(Q^\top K^\top V) \) in (7), and \( \sigma(V^\top V) \) is the graph value \( V \). Looking into the definition of attention map in (3), the map \( G^a \) implies how much information should flow from the nodes \( V \) to the nodes \( Q \). \( (\mathbb{I} \cdot p^\top) \circ \sigma(U^\top q_i) \) and \( \sigma(V^\top v_j) \) correspond to query \( Q \) and key \( K \) in (7), respectively. Instead of simply using scaled dot-product, low-rank bilinear pooling is utilized to minimize the difference between distributions of \( Q \) and \( V \) based on (4). Moreover, (7) only considers single modality of inputs, while VQA models need to consider the multimodal inputs (i.e., image and question). An additional Hadamard product of \( \sigma(Q^\top U) \) and \( G^a \sigma(V^\top V) \) is, thus, included in (8) to generate the output nodes \( z_1^\top, \ldots, z_m^\top \), where \( z_i^\top \in \mathbb{R}^K \). Finally, the joint embedding \( z \) represents the whole graph by summarization of all nodes in \( Z^\top \) based on their weight \( G^b \) in (9).

Even though (8) and (9) provide an elegant approach to investigate the relationship between question features \( Q \) and image features \( V \), a simply summarization over columns of \( Z^\top \) in (9) cannot fully address the connections between the joint embeddings \( z_1^\top, \ldots, z_m^\top ) \) corresponding to words. Given the question and image in Fig. 1, BAN [i.e., (8) and (9)] can locate a variety of fruits in the image according to the word
of nodes and edges, respectively. The image-graph has $V$ as shown in Fig. 2. We also find that the right answer may not be decided at once; instead of an overall representation.

The major target of the image-graph is to locate the objects and words themselves multiple times. Therefore, we stack our graphs to make the words interact with the objects and words themselves multiple times.

“A. Image-Graph

The major target of the image-graph is to locate the objects related to the semantic information of each word in the question. Beginning with (8), we have a multiglimpse extension, as shown in Fig. 2.

Consider the graph $G = \{V, E\}$, where $V$ and $E$ are the set of nodes and edges, respectively. The image-graph has $V = \{Q \cup V\}$ and $E = G^c$, where $Q \in R^{C \times m}$ are textual features of the question, $V \in R^{D \times n}$ are visual features of the detected objects, and $G^c$ are the computed graph edge with weights based on $Q$ and $V$. Multiple-glimpse attention aims to jointly model the graph from different representation subspaces; it can stabilize the training process and improve the performance. Therefore, we extend $G^c$ to multiple glimpses following [13], [14], resulting in $G^c \in R^{m \times n \times g'}$, where $g'$ is the number of glimpse. The $j$th graph attention is computed as

$$G_j^c = \text{softmax}(((\mathbf{1} \cdot \mathbf{p}_j^\top) \circ \sigma(Q^\top U^c))\sigma(V^\top V'^c)^\top) \quad (10)$$

where the parameters $U^c$ and $V'^c$ are shared among glimpses except for $\mathbf{p}_j^c$, which can be seen from the upper part of Fig. 2. After learning the graph attention, we use (8) to generate the joint embeddings as

$$H_j^\top = \text{BGN}_j^c(Q, V; G_j^c) = \sigma(Q^\top U_j^c) \circ G_j^c \sigma(V^\top V'^c) \quad (11)$$

where $H_j^c \in R^{K \times m}$ represents the output of image-graph at glimpse $j$.

Instead of concatenation [14], [15], [33] of joint embeddings from each glimpse, we follow BAN to use the residual form to integrate previous learned joint embeddings, as shown in the lower part of Fig. 2. Then, (11) becomes

$$H_j^c = W_j^c \text{BGN}_j^c(H_{j-1}, V, G_j^c)^\top + H_{j-1}^c \quad (12)$$

where $H_0^c = Q$, and $W_j^c \in R^{C \times K}$ projects the joint embeddings to the same dimension of $Q$. By convention, we use the output of the last glimpse to represent the whole image-graph, denoted as $H = H_g^c$.

B. Question-Graph

For the question-graph, similarly, we have the graph nodes $V = H$ and graph weight $E = G^q$, where $H \in R^{C \times m}$ are the output nodes of the image-graph, and $G^q \in R^{m \times n \times g'}$ are the self-attention graph weights of multiple glimpses based on $H$ denoted as

$$G_j^q = \text{softmax}(((\mathbf{1} \cdot \mathbf{p}_j^q) \circ \sigma(H^\top U^q))\sigma(H^\top V'^q)^\top). \quad (13)$$
The structure of our question-graph is similar to the image-graph in Fig. 2, except that both inputs are $H$. Different from (9), which summarizes the outputs from the image-graph based on $G^r$ to represent the whole graph, $G^r$ in (13) learns the context of each node by exchanging their information. Based on the graph weight $G^r$, nodes of the question-graph at glimpse $j$ gather information from others and are represented as (12)

$$ O_j = W_j^r \text{BGN}_j^r (O_{j-1}^r, H; G^r_j)^T + O_{j-1}^r $$

where $W_j^r \in \mathbb{R}^{C \times K}$ and $O_0^r = H$. The outputs of question-graph $O$, abbreviated version of $O_j^r$, can be utilized to answer the question by summarizing all the nodes to represent the whole graph.

As we mentioned above, the question may be compositional and complex, which means that they need multistep reasoning. Thus, we form the basic module of our BGNs with one image-graph following by one question-graph, and we stack the module for multiple layers to compose our framework shown in Fig. 1. The first layer of the image-graph takes textual nodes $Q$ as a query to locate the related visual information in $V$ and outputs their joint nodes $H_1$, and the higher layer of it takes the outputs of $i−1$ layer of the question-graph, $O_{i−1}$, as a query to involve more visual information related to the prior knowledge learned from the lower layers. The layer $i$ of question-graph aims at exchanging the information between nodes of $H_i$ to model their context and generates $O_i$ either for the prior knowledge of the next layer or for the answer prediction.

Note that we can also reformulate (8) as $Z''^T = \sigma (V^T W) \circ G' \sigma (Q^T U)$, where $Z'' \in \mathbb{R}^{K \times n}$ and $G' \in \mathbb{R}^{n \times m}$, based on which we could build the relationship between objects as other similar models [5], [6], [16]–[18] theoretically. However, in practice, this will lead to the memory problem of building the graph among nodes of $Z''$, such as the question-graph. Since the bilinear graph costs more memory than the scaled dot-product, the object-object graph $G \in \mathbb{R}^{n \times n}$ is much larger than the word-word graph $G \in \mathbb{R}^{K \times m}$, where $n = 100$ and $m = 15$. In addition, we model such relationship implicitly by modeling the relationship between joint embeddings $H_i$ and object feature $V$ in the image-graph from the second layer and between the nodes in $H_i$ in the question-graph.

After stacking $L$ layers of BGNs, we summarize all the nodes of $O_L$ to represent the whole graph and pass it to a two-layer MLP to compute the score of answers

$$ p = W a \sigma (W a O_L \cdot 1) $$

where $W a \in \mathbb{R}^{2C \times C}$, $W a' \in \mathbb{R}^{|A| \times 2C}$, and $|A|$ is the size of answer set $A$.

C. Compared With Other Graph-Based Methods

Though we also investigate the VQA problem from a graph view, our model has several differences from existing graph-based methods. Compared with MUREL [47], representing the question as a single vector to fuse with the image features at each step and emphasizing the relationship between objects, our method pays attention to modeling the relationship between words and objects, as well as between words and words. Regarding DAF [5], MCAN [6], Vilbert [16], VL-bert [17], and LXMERT [18], their structures are similar with ours; all of them use the cross- and self-attention to model the relationship between words and objects. However, we implement the attention in a different way. In particular, these related works directly borrow scaled dot-product attention from Transformer [14], which targets modeling the relationship between single-modal inputs (words). Thus, this linear fusion method is less effective in modeling the relationship between the multimodal inputs (words and objects) in the VQA problem. In contrast, we first introduce the BAN, which considers bilinear interactions among two groups with different distributions. By reformulating it from a graph view, we demonstrate its disadvantages and overcome them with our BGN models. Therefore, our model has a better explanation than these linear methods. Also, as shown in Tables I and II, our method achieves a better performance than these models trained only with VQA data and is comparable to these models requiring extra caption data for visual language pretraining, proving the effectiveness of our bilinear graph in modeling the relationship between multimodalities.

V. Experiments

In this section, we evaluate our BGNs on the VQA v2.0 dataset [4], [22]. We first introduce this dataset and then describe our implementation details and results, and, finally, the qualitative analysis.
A. Dataset

VQA v2.0 Dataset: The dataset was built based on the MSCOCO images [27], it contains 1.1M questions asked by humans, and each question is annotated by ten people. Compared with the VQA v1.0 dataset [4], it emphasizes visual understanding by reducing the text bias learned from the questions. The dataset is split into three parts: training, validation, and test provided by the official website,¹ which has 80k images and 444k questions, 40k images and 214k questions, and 80k images and 448k questions, respectively. The answers of the training and validation datasets are published for the training model, while those of the test dataset are unknown and should be predicted by the proposed model before being uploaded to the server for performance evaluation. Based on the answer category, the questions can be classified into three types, i.e., yes/no, number, and others. We train our models with different settings on the training dataset and evaluate their accuracy on the validation dataset by the tools from [4]; then, we pick the settings of the best model to train it on the training and validation datasets with extra data from Visual Genome [26] that has 108k images and 1M questions, reporting the results on the test server. Moreover, the number of questions in the validation set is 214 354, which is large enough to evaluate the model. Thus, following other related works [6], [10], [13], we do not use cross-validation to select the parameters and evaluate trained models.

B. Implementation Details

The number of answers appearing in the dataset is very large. Many of them only have few training examples, which makes it difficult to accurately learn the representation of these answers. Thus, following the processing by related works [4], [6], [13], [16], [42], we construct the answer vocabulary by restricting to the words that appear in the training and validation datasets more than eight times, resulting in |A| = 3129. We then truncate or pad a question’s length \( m \) to 15 words, which covers 99% of questions. The weight of padding tokens in question-graph \( G' \) will be set to \(-\infty\) before softmax to reduce its impact. There are two methods to encode the question: one is LSTM [12], and the other one is BERT [15]. For the former one, we pass the question through a one-layer LSTM, whose input dimension of each word is 600, 300 of which is learned by our model and another 300 from pretrained GloVe vector [48]³ is fixed, and the output dimension \( C \) is 1024. For the latter one, we embed the words with the default encoder of BERT and project the outputs of the last layer into vectors with dimension \( C \). We extract object features from a Faster-RCNN model [1] pretrained on Visual Genome, which has 1600 object classes and 400 attributes. For each image, we obtain top \( n = 100 \) objects with their object features and regions based on their predicted probabilities, and each object feature is presented by mean-pooling of their convolutional features with \( D = 2048 \). The joint embedding sizes \( K \) and \( K' \) are set to 1,024, and the rank \( d \) is set to 3 during computing the graph attention weights in the image-graph and question-graph to increase its capacity. In order to save memory in each layer to make our network go deeper, we reduce the glimpse number from 8 (best performance in BAN) to \( g' = g = 4 \). Weight normalization [49] and dropout [50], [51] with \( p = 0.2 \) are added after each linear mapping to stable the output and prevent overfitting.

Due to the fact that there might exist multiple correct answers for a question, we utilize the binary cross-entropy (BCE) loss as loss function, which is calculated as

\[
L = -\sum_{i=1}^{|A|} (y_i \log \phi(p_i) + (1 - y_i) \log(1 - \phi(p_i)))
\]

where \( y_i = \min((\text{number of people that provided answer } a_i / 3), 1) \), and \( \phi(x) \) is the sigmoid function denoted as \( \phi(x) = (1/(1 + e^{-x})) \). Adamax [52], a variant of Adam, is used to optimize our model. The initial learning rate is 0.001 and grows by 0.001 every epoch until reaching 0.004 for warm start, keeps constant until the 11th epoch, and decays by 1/4 every two epochs to 0.00025. The batch size is 128.

We implement our model on Pytorch⁴ v1.2 and train it by using 4 NVIDIA Tesla V100 GPU. It costs 0.4 \( s \) for each step in training the best model (three layers of the bilinear graph) with the LSTM encoder and, thus, takes totally 18.5 \( h \) for 13 epochs. The best model with BERT-large as its question encoder cannot be sent to 4 V100 at once. Thus, we compute the gradients with batch 64 at each step and backpropagate it every two steps. It takes 0.57 \( s \) for every step and totally 52.7 \( h \) for the whole training process.

C. Comparison With State of the Art

In Table I, we evaluate our method on the VQA v2.0 test-dev dataset, which achieves state of the art. The overall accuracy of our BGNs+Glove model is 1.31% higher than it of BAN+Glove, nearly 3.0% on number metric. It can be explained that the counting task is a kind of relationship among objects, which tries to find similar or comparable objects in the latter layers with objects grounded by the previous layers. With the extra Counter module [53] in our BGNs+Glove+Counter model, we make another 3% gain on number metric, which means that our method and Counter solve the number problem from different aspects and can be united. Although we do not model the relationship between objects explicitly as DDAF [5] and MCAN [6], we model it implicitly, as we mentioned in Section IV-B. Moreover, we revisit the BAN from a graph view to propose a variation of scaled dot-product to learn the attention, and two kinds of graphs are generated to exploit the intramodality and cross-modality relationship. Thus, the accuracy of our model with LSTM encoder (70.97% for BGNs+Glove) is 0.75% and 0.34% higher than DDAF and MCAN, respectively, even comparable with these models equipped with stronger question encoder (70.59% for DDAF+BERT and 71.09% for MLIN+BERT).

These BERT-like models (Vilbert [16], LXMERT [18], VL-bert [17], and UNITER [43]) need to pretrained on a

¹https://visualqa.org/index.html
²https://github.com/GT-Vision-Lab/VQA
³https://spacy.io/models/en
⁴https://pytorch.org/
large number of out-domain data, such as image caption (3.3M Conceptual [46], 5.4M VG [26], and 1.0M GQA [19]), which costs much more time (4 V100 and 8.5 days for pretraining in LXMERT). Moreover, they focus on how to reconstruct the words in the captions and objects in the images and the alignment between them, and they just simply regard the words and objects as the nodes of the same type and concatenate them in the input sequence to build the relationship with a universal Transformer. With only in-domain data (0.6M VQA v2.0 and 1.0M VG) without pretraining and BERT structure as our question encoder, our best model (72.56% for BGNs + BERT + Counter) achieves a comparable performance with them on the test-std dataset, proving the effectiveness of our bilinear graph in learning the relationship between objects and words, as well as their joint embeddings.

To improve accuracy, we replace the backbone of Faster-rcnn from ResNeXt-101 to ResNeXt-152 and train it on the VG dataset following BUTD [1]. With this stronger detector, we train our models with and without Counter modules on different layers. Then, we rank the answers by summing the scores of all the models and choose the highest one as the predicted result. Table II reports the accuracy on the leaderboard of the VQA Challenge 2020 5 and MCAN [6] (winner of 2019). Our bilinear graph has secured second place, after GridFeat [28], which replaces the object-based image features with the grid features on MCAN.

After this challenge, we run our best model with the grid feature [28]. Table III reports the performance of our model with the features. The accuracy of our model with grid feature is close to MCAN on the test-dev set, which is used for debugging and validation experiments. 6 However, we perform much better on the test-std set, which has a wider range of questions and is used for reporting results.

D. Ablation Study

We conduct several ablation studies to verify the contribution of each module in our BGNs. The number of questions in the validation set is 214,354, which is large enough to evaluate the model. Thus, following other related works [6], [10], [13], we do not use cross-validation to select the parameters and evaluate the trained models. The first four lines in Table IV show the accuracy of BAN on the VQA v2.0 validation dataset, in which BAN-4 and BAN-8 represent the model with four and eight glimpses, respectively. It can be seen that simply stacking the module of BAN can improve the accuracy to some extent (0.35% and 0.46% in the two- and three-layer models, respectively) compared with the one-layer model. Although the multilayer BAN might gain more visual information related to the global representation in (9) without exchanging context information, it is not clear about the relationship between entities in the input question. In contrast, our one-layer model, BGNs × 1, gains an accuracy 0.62% and 0.43% higher than BAN-4 × 1 and BAN-8 × 1, respectively, proving the effectiveness of our proposed question-graph even with fewer glimpses. However, if we only stack the question-graph for multiple times (V-graph + Q-graph × 2 and V-graph + Q-graph × 3) with only one-layer image-graph, the performance grows slower than that of the BGNs × 2, and this might be caused by that the question-graph can only propagate the information already learned by the image-graph but cannot involve more factual information required in the image to answer the questions.

If we replace the proposed BGN in the question-graph with the scaled dot-production (SDP), the accuracy declines (−0.06% and −0.41% than BGNs with the same layer) and grows slightly (0.09%) by stacking this kind of graph. It can be explained that, though the inputs of question-graph H are

---

5https://visualqa.org/roe.html
6https://visualqa.org/challenge.html
the same type of nodes, the nodes themselves are hybrid, and the relationship between them cannot be fully expressed by the linear method. By stacking three layers of the BGNs, our model achieves 67.21% on the overall accuracy, which is chosen as the best model.

In addition, we investigate the absolute increase in the score of our models compared with the single-layer BAN on the questions with varied lengths to show the ability of our model on multitask reasoning in Fig. 3. Our models with different layers outperform the one-layer BAN, especially on these long questions. The one-layer model does not perform and the other three models for long questions due to its shallow graphs. With more layers, our model becomes better at long questions and achieves a 1.7% increase at word number of nine. What interests us is why the performance drops at the four-layer one. Comparing the three-layer model and the four-layer model, the former one works better in short questions (word number < 8), which takes 79% of all questions, while the latter one has a higher score in the long questions, which may explain the cause of performance drop. This phenomenon also inspires us to design a network in the future to classify the questions to fit different layers of graphs.

Compared with the other related deep coattention approaches, such as MCAN (six layers) and LXMERT (14 layers), our BGNs are more shallow. There are two major reasons to explain the preference for the lower depth. First, the bilinear method is more effective in fusing the multimodal inputs than the scaled dot-product, which implies that a few several layers are sufficient for BGN to exchange the information between words and objects. Second, the accuracy
of our four-layer model performs better on longer questions (word number ≥ 8), while the three-layer one is good at handling shorter questions, as shown in Fig. 3. In practice, 79% questions of VQA v2 are short questions; the three-layer model is much preferred to achieve the best overall performance.

Furthermore, we explore the influence of BERT [15] as the question encoder on our method since BERT is trained on large text corpus, which leads it to have better generalization and representation of textual features. Thus, we replace LSTM with BERT when modeling the question and fine-tuning its weight with different strategies as shown in Table V. With the base model of BERT without fine-tuning (BGNs × 1 + Base with \( lr \times 0 \)), the accuracy increases slightly (0.1%), as the domain shifts between the questions generated from the images of COCO [27] and the questions based on Wikipedia from SQuAD [58]. By increasing its learning rate, the performance boosts and achieves the best accuracy at \( lr \times 0.01 \). With this learning rate, we switch to the BERT-large model that is deeper and wider than the base one, and the performance grows and keeps going by stacking our bilinear graph model on it, proving that our model is effective and compatible with BERT.

### E. Experiments on VQA-CP v2

However, the training and test datasets of VQA v2 are highly correlated [59], which means that the model may memorize the bias between question and answer in the training set to demonstrate the performance in the test set. Therefore, we evaluate our model on the VQA-CP v2 dataset [59], which reorganized the VQA v2 dataset to change the distribution of answers for each question type between training and test sets.

We compared our BGN model with the baseline model BAN [13] on its test set. Both models were trained on the VQA-CP v2 training set under the same setting of VQA v2 [4] and then evaluated on the test set of VQA-CP v2. The results were reported in Table VI. Since both BAN and our BGN are designed to answer the questions in VQA v2, which involves the question and answer bias, they do not perform these unbiased models [60]–[62] that have been exclusively designed with modules and objective functions to tackle the problem of VQA-CP. However, the accuracy of our model is still 0.6% higher than that of BAN on the VQA-CP v2 dataset, which demonstrates the advantages of our exploitation of the image-graph and question-graph.

---

**Fig. 5.** Examples illustrating the answers predicted by BAN and our graph models. BAN, L1, L2, and L3 denote the answers predicted by BAN, one-layer, two-layer, and three-layer of our model, respectively.

**TABLE V**

| Model              | \( lr \times 0 \) | VQA Score |
|--------------------|--------------------|-----------|
| BGNs × 1 + LSTM    | 1                  | 66.43     |
| BGNs × 1 + Base    | 0                  | 66.52     |
| BGNs × 1 + Base    | 0.001              | 67.62     |
| BGNs × 0.01        | 0.01               | 68.09     |
| BGNs × 0.1         | 0.1                | 67.84     |
| BGNs × 0.01        | 0.01               | 68.20     |
| BGNs × 0.01        | 0.01               | 68.50     |
| BGNs × 0.01        | 0.01               | 68.71     |
TABLE VI
ACCURACY ON VQA-CP v2. TEST SET

| Model   | Overall | Yes/no | Number | Other |
|---------|---------|--------|--------|-------|
| BAN [13] | 41.62   | 43.33  | 13.35  | 48.49 |
| BGNs (ours) | 42.20   | 43.07  | 14.36  | 49.38 |

**F. Qualitative Analysis**

To visualize the effects of each module in our BGNs, we present the learned attention maps of the image-graph and the question-graph in each layer to show how the networks work. In Fig. 4, we visualize the attention maps for our three-layer model. The first image in the first row is the input image with the top-20 detected bounding boxes, and the following three images are the heat maps showing the attention between words and words in the question-graph. The three heat maps in the second row show the attention between words and objects. The multiple glimpses’ attention maps in each graph are summed at each layer to briefly show the attended objects and words. Given the question “What fruit is on the left edge?” in Fig. 4, the image-graph of the first layer attends kinds of objects (object 0 for apple, 3 for orange, 8 for cup, 11 for tomato, and 19 for grape) in the input image, while the question-graph of the first layer only broadcasts the learned fruit information from the word “fruit” to other words and chooses “tomato” as the answer, probably because the amount of “tomato” is the biggest among all detected fruits. The image-graph of the second-layer attends on “orange” (object 3) that is to the left of “tomato” and the question-graph keeps collecting “fruit” and “edge” information. In the third layer, the image-graph locates “apple” (object 0) that is on the left edge, and every word in the question-graph pays its attention to the “edge” information to predict the answer.

In Fig. 5, we show the answers predicted by BAN and our models with one layer, two layers, and three layers, respectively. In the first image of the top row, BAN cannot correctly answer the question because the entities of “young girl” and “bag” learn their positions, respectively, but they do not know each other’s information, while our proposed question-graph exchanges such positional information to make it possible to compare the relative direction of the two entities. A similar question can also be found in the first image of the bottom row, which shows that our model approaches the correct answer step by step as the layer of the graph increases. Moreover, our model can find the implicit relationship between objects even though the sheep are far away from the dog in the second image of the top row, as well as abstract scenes in the second image (five circles representing Olympics) and third image (many trees composing forest) of the bottom row. Furthermore, our model finely discriminates the highly overlapped objects, such as two sheep in the second image and the rope in the fourth image of the top row; it is possibly because the question-graph undertakes some burden from the original graph of BAN, which makes the image-graph spare more effort on learning details in the image.

**VI. CONCLUSION**

In this article, we analyze the disadvantages of graph attention networks and Transformer and interpret bilinear attention works from a new perspective. BGNs composed of layers of image-graph and question-graph have been developed. The image-graph learns the graph between words in the question and objects in the image and generates the joint embeddings of them, while the question-graph models the graph between these joint embeddings in terms of words to exchange context information. Moreover, from the second layer of image-graph, our method models the graph between the joint-embeddings and objects, which implicitly builds the relationship between objects and objects. The ablation studies show that our BGNs significantly outperform the traditional linear way of scaled dot-product on a variety of questions. On the VQA-CP v2 testing dataset, we also achieve a better performance than BAN, demonstrating a stronger reasoning ability of our model. Applying this approach to the VQA 2.0 testing dataset, our method achieves state-of-the-art performance even with fewer layers than DFAF and MCAN, proving the effectiveness of our fusing method. Also, it is comparable to these BERT-like models that required visual language pretraining on a large amount of the out-domain dataset and more GPU resources. By ensembling our models, we rank second in the VQA challenge 2020.
Dalu Guo received the bachelor’s and master’s degrees from the School of Computer Science and Engineering, Beihang University, Beijing, China, in 2010 and 2013, respectively, and the Ph.D. degree from the UBTECH Sydney Artificial Intelligence Centre and the Faculty of Engineering and Information Technologies, School of Computer Science, The University of Sydney, Sydney, NSW, Australia, in 2021.

His papers have been published in IEEE Conference on Computer Vision and Pattern Recognition (CVPR). His research interests mainly include visual question answering, visual dialogue, natural language processing, and so on.

Dr. Guo got the First Place in the Visual Dialog Challenge 2018 (SiVL Workshop of European Conference on Computer Vision (ECCV) 2018) and the Second Place in the VQA Challenge 2020 (Visual Question Answering and Dialog Workshop of CVPR 2020).

Chang Xu (Member, IEEE) received the Ph.D. degree from Peking University, Beijing, China, in 2016.

He is currently a Senior Lecturer and an ARC DECRA Fellow with the School of Computer Science, The University of Sydney, Sydney, NSW, Australia. He has published over 100 articles in prestigious journals and top-tier conferences, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE T-NNLS, IEEE TRANSACTIONS ON IMAGE PROCESSING, the International Conference on Machine Learning (ICML), Neural Information Processing Systems (NIPS), the International Joint Conference on Artificial Intelligence (IJCAI), and the Association for the Advancement of Artificial Intelligence (AAAI). His research interests lie in machine learning algorithms and related applications in computer vision.

Dr. Xu received several paper awards, including the Distinguished Paper Award in IJCAI 2018. He regularly served as the (Senior) Program Community Member for many conferences, e.g., NIPS, the International Conference on Learning Representations (ICLR), ICML, Conference on Computer Vision and Pattern Recognition (CVPR), the International Conference on Computer Vision (ICCV), IJCAI, and AAAI. He has been recognized as a Top Ten Distinguished Senior PC in IJCAI 2017.

Dacheng Tao (Fellow, IEEE) is currently the President of the JD Explore Academy, Beijing, China, and the Senior Vice President of JD.com, Beijing. He is also an Advisor and Chief Scientist of the Digital Science Institute, The University of Sydney, Sydney, NSW, Australia. He mainly applies statistics and mathematics to artificial intelligence and data science, and his research is detailed in one monograph and over 200 publications in prestigious journals and proceedings at leading conferences.

Dr. Tao is a fellow of Australian Academy of Science, American Association for the Advancement of Science (AAAS), and the Association for Computing Machinery (ACM). He received the 2015 Australian Scopus-Eureka Prize, the 2018 IEEE ICDM Research Contributions Award, and the 2021 IEEE Computer Society McCluskey Technical Achievement Award.