Bilateral Cross-Modality Graph Matching Attention for Feature Fusion in Visual Question Answering

Jianjian Cao†, Xiameng Qin†, Sanyuan Zhao†, and Jianbing Shen*, Senior Member, IEEE

Abstract—Answering semantically complicated questions according to an image is challenging in a visual question answering (VQA) task. Although the image can be well represented by deep learning, the question is always simply embedded and cannot well indicate its meaning. Besides, the visual and textual features have a gap for different modalities, it is difficult to align and utilize the cross-modality information. In this article, we focus on these two problems and propose a graph matching attention (GMA) network. First, it not only builds graph for the image but also constructs graph for the question in terms of both syntactic and embedding information. Next, we explore the intramodality relationships by a dual-stage graph encoder and then present a bilateral cross-modality GMA to infer the relationships between the image and the question. The updated cross-modality features are then sent into the answer prediction module for final answer prediction. Experiments demonstrate that our network achieves the state-of-the-art performance on the GQA dataset and the VQA 2.0 dataset. The ablation studies verify the effectiveness of each module in our GMA network.

Index Terms—Graph matching attention (GMA), relational reasoning, visual question answering (VQA).

I. INTRODUCTION

Given a visual image, the goal of visual question answering (VQA) is to correctly answer a relative natural language question. It requires a comprehensive cross-modality understanding and visual-semantic reasoning between images and questions. VQA plays an important role and has a wide potential application in the future life, such as virtual assistants and autonomous agents. Therefore, it has aroused extensive interest from both vision and natural language communities. With the explosive progress of deep neural networks (DNNs), a large amount of deep learning methods, i.e. BUTD [1] and Pythia [2], have been developed for better representation learning. These feature representations have been introduced into VQA and efficiently improved the performance of existing algorithms. In recent years, a plenty of VQA studies focused on relational reasoning. The attention mechanism [3], [50], which is capable to encapsulate the most relevant information from the image or the question, has been adopted for relation reasoning. BAN [4] and DFAF [5] designed a variety of attention to explore the implicit relationship between the image and question domain, and their effects are remarkable. These VQA frameworks adopted simple attention mechanisms, learning the “black-box” relationships in an agnostic dimensional feature space without underlying restrictions. The arbitrariness increased the risk of overfitting and limited high-level relational reasoning. Besides, some recently works [6], [7] concentrated on learning an explicit visual-semantic relationship in an image via visual relation detection (VRD) [8], [9]. To some extent, this kind of frameworks put forward a new possible direction for solving the relational reasoning in the VQA task, but they increased uncertainty of the model, making the performance heavily dependent on the relationship learning algorithms. In addition, predominant approaches of VQA focused on learning a joint cross-modality representation to represent the alignment between the image region and the corresponding question. Specifically, bilinear fusion [10] exploited bilinear pooling to capture pairwise interactions among the modalities. Some sophisticated schemes, including MCB [11], MLB [12], and MUTAN [13], which designed different linear fusion strategies, demonstrated better performances and less computational burden than bilinear fusion. Nevertheless, modern fusion methods carried out feature alignment in the absence of modal structure (e.g., positional relationships of the image regions and grammatical relationships of the question) and output one monolithic vector unable to describe fine-grained information for both visual and language semantics. As a result, the alternatives almost produced a suboptimal solution leading to imperfect cross-modality feature representation.

The graph method is generally known as using a topological data structure to express complex relationships between nodes. In VQA, several graph-based approaches [14], [15] were proposed and shown great improvement in visual understanding [16]. They leveraged topology to the subtle context information in order to remedy the defect of Euclidean distance in relationship inference. The drawbacks lie in the ignorance of the syntactic relationship of the language modality and mere consideration of visual information. This is a serious problem for VQA model understanding.

Manuscript received 1 December 2020; revised 24 July 2021 and 31 October 2021; accepted 4 December 2021. Date of publication 7 February 2022; date of current version 1 March 2025. This work was supported by the National Natural Science Foundation of China under Grant 61902027. (Corresponding author: Sanyuan Zhao.)

Jianjian Cao and Sanyuan Zhao are with the Department of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: caojianjianbit@gmail.com; zhao-sanyuan@bit.edu.cn).

Xiameng Qin is with Baidu Inc., Beijing 100193, China (e-mail: qinxiameng@baidu.com).

Jianbing Shen is with the State Key Laboratory of Internet of Things for Smart City, Department of Computer and Information Science, University of Macau, Macau, China (e-mail: shenjianbingcz@gmail.com).

Digital Object Identifier 10.1109/NNLS.2021.3135655

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
In this article, we propose a GMA network to solve the problems of unsuitable question description and the alignment between the visual and textual modality information. As shown in Fig. 1, our network constructs two graph data structures, to depict the positional geometry of image regions and syntactic structure of the question. In the view of the topological information, the graph patterns for two modalities bring the convincing relational reasoning in cross-modality understanding. To solve the shortcoming of the traditional attention mechanisms in VQA, we propose the GMA module to develop a more intuitive relational reasoning. Based on the visual and textual graphs, the GMA module guides to align the attention across the two modalities. We first explore information relationships inside the visual and textual modalities by a dual-stage graph encoder to represent the relationships of the nodes. The graph encoder in our GMA produces meaningful internal relationships in the image and the question, separately, improving the attention interpretation. Then, the intramodality features are forward into the GMA modules to align and pass information between visual and textual features in a bilateral manner. By practicing the graph matching between the visual and texture graph patterns, each node is capable to adopt a feature fusion inference from the multimodal representations, rather than a monolithic feature in previous works. Furthermore, the GMA contributes to the performance with the participation of multimodality structural information. Finally, each graph feature is weighted by the bilateral cross-modality attention and sent to the answer prediction module to calculate the score for the potential answers.

Specifically, as shown in Fig. 1, the GMA utilizes the graph convolutional networks (GCNs) to successively capture the explicit and implicit relations among the nodes inside each modal (the red arrows in the GMA module). These relational features reveal fine-grained intramodality concepts and provide a holistic view of modal interpretation. Next, a pairwise matching (the yellow dashed arrows linking visual and question graph nodes) is applied to intermodality nodes and attaches the semantic alignment. The graph matching operations dynamically update the nodes and pass information flows within each modality to capture higher order intermodality relations. In this way, every node aggregates the attention-weighted information from the other instances under the guidance of graph structure. Besides, the GMA modules can be stacked and explore a deeper level semantic relationship.

The main contributions of our work are fourfold.
1) We propose to construct graphs for both the image regions and the questions, respectively, to explore their structured intramodality information and unify their representation for further alignment.
2) We design a novel GMA module that contains a dual-stage graph encoder and a bilateral cross-modality GMA. Inheriting graph learning, this module introduces the graph matching operation to solve the cross-modality alignment and feature fusion problem and learns intermodality and intramodality relations.
3) By cascading several GMA modules, the network can learn the deeper relations and achieve high-level relational reasoning, which makes the VQA inference more reasonable.
4) We conduct ablation studies and compare our network with state-of-the-art methods on the famous GQA and VQA 2.0 datasets. The results convince the effectiveness of our proposed GMA module.

II. RELATED WORK
A. Visual Question Answering
It is a challenging problem, which involves two fields of computer vision and natural language processing (NLP). The VQA task [17] requires that the trained model can answer any natural language questions of a given image. In real application scenarios, it usually becomes a multimedia answering task [18] and can also be directly applied to the multimodal dialogue system [19]. Current algorithms [1], [4], [20], [21] for VQA task are usually designed as a multilabel classification problem, which makes the neural network infer answers in a limited set of classes. The current mainstream VQA framework is usually composed of image encoder, question encoder, multimodal features fusion module, and answer predictor. Feature learning is an essential part of VQA algorithms. Due to the success of deep representation learning, BUTD [1] derived from Faster RCNN [23], and the performance of VQA has been greatly improved. Recently, the researchers pay more attention to the relational reasoning of VQA task. Specifically, Murel [24] proposed to stack several layers of attention to gradually focus on the most important regions. ReGAT [25] proposed graph-based attention mechanisms to learn the relational semantics between the visual and language modalities. Stacked attention- [20] and coattention-based [21] approaches inferred visual and textual attention distributions for each modality. BAN [4] inferred a bilinear attention distribution between two modalities. DFAF [5] integrated cross-modality self-attention and coattention mechanisms to achieve effective information flows within and between two modalities. Besides, some recent algorithms attempted to solve the problem of multimodality feature fusion in VQA. MCB [11], MLB [12], and MUTAN [13] designed different bilinear fusion strategies to fuse the cross-modality features. In addition, some popular...
methods [26], [27] begin to design different data augmentation algorithms to enhance the effect of VQA model. Those methods have proved useful for the VQA task, but there are still some unsolved issues, such as the alignment between the two modalities and the absence of the structural information of each modality. In this article, we construct two graphs to represent the geometry relationship of image regions and the syntactic structure of its question. A GMA module is designed to successively align and fuse the visual and textual features, in order to learn the semantic representation for final answer prediction.

### B. Relational Reasoning

Recently, the deep learning community began to deal with complex relational reasoning problems, such as relationship detection [22], [33], object recognition [28], and abstract reasoning [29]. Yu et al. [30] proposed the HDWE model to address click feature prediction from visual features for fine-grained image recognition. Yu et al. [31] proposed the SPE-VLAD layer to reflect the structural information of the images for place recognition. Andreas et al. [32] proposed multistep reasoning models NMNs that built question-specific layouts. Based on NMNs, Stack-NMN [34] algorithm was proposed to practice more effective multistep reasoning. FiLM [35] modulated an image representation with the given question via conditional batch normalization and was extended in [36] with a multistep reasoning procedure, where both modalities can modulate each other. MAC [37] performed multistep reasoning while recording information in its memory. QGHC [38] predicted question-dependent convolution kernels to modulate visual features. In addition, recently proposed methods [4], [5] applied various attention mechanisms in reasoning framework to enhance the learning of implicit relational semantics. Given a question, these models assigned an importance score to each region and employed them to weight-sum pool the visual representations. The attention mechanism has been proven to be effective in solving the problem of relational reasoning in the VQA task, but only learning the implicit relations is insufficient. It is necessary for the VQA model to learn explicit relations according to the structural information of each modality. In order to enhance the relational reasoning ability, we propose a novel GMA mechanism to learn the relational semantics between the image and the question by combining with the designed graph matching method.

### C. Deep Graph Matching

Graph matching [39] is a fundamental problem that aims to establish the node correspondences among multiple graphs and has been widely studied in computer vision areas. To find the maximal similarity between the matched graphs, both the unitary similarity between nodes and pairwise similarity edges from separate graphs are integrated in graph matching methods [40], [41]. Naturally, graph matching can encode the high-order geometric information in the matching process so that it can be used to model complex relationships and more diverse transformations. Because of the expressiveness and robustness, graph matching relates to many computer vision problems, such as visual tracking, motion recognition, and image classification. Recently, researchers began to use the deep features in geometric and semantic visual matching tasks. GMNs [42] proposed graph matching networks for learning the similarity of graph-structured objects. CLKGM [43] proposed a topic entity graph to represent the knowledge graph context information of an entity and estimated the similarity of two graphs by using the attentive-matching method. DLGM [44] proposed an end-to-end graph matching model that integrates node feature extraction, node/edge affinities learning, and matching optimization together. PCAGM [45] explored GCNs for graph matching task. In this article, we involve the graph matching algorithms to combine the visual features of the image with the semantic features of the question so as to align the two different feature subspaces and learn the high-level cross-modality relations to reasoning the answers of the questions.

### III. OUR METHOD

As shown in Fig. 2, the proposed GMA network consists of a graph construction module, stacked GMA modules, and
an answer prediction module. First, we build graph structures for both the image and the question to maximize retention of intramodality relationships in the graph construction module (Section III-A). Then, the GMA module (Section III-B) explores the cross-modality relationships by carrying bilateral GMA between the visual and the textual graph. Finally, the reasoning features from the GMA are aggregated with the textual feature and fed into the answer prediction module (Section III-C) for the final answer.

A. Graph Construction

1) Visual Graph Construction: In order to fully express the explicit semantic relationship between objects in visual modality, we construct the visual graph for each image. Given an image \( I \), we build its visual graph \( \mathcal{G}_1 \) according to the object regions and their relationships

\[
\mathcal{G}_1 = (V^m, E^m), \quad \{V^m = \{v^m_i\}_{i=1}^{K_1}, E^m = \{e^m_{ij} \in \mathbb{R}^{K_1 \times K_1}\} \quad (1)
\]

where \( V^m \) represents the set of graph nodes, \( E^m \) is the graph edges, \( v^m_i \) corresponds to a visible object in \( I \), and \( K_1 \) is the number of visual graph nodes. The visual feature \( V^m \) can be collected by concatenating the RoI-pooling feature \( R = \{r_i\}, i \in [1, \ldots, K_1] \) from the Faster R-CNN detector [23] and the normalized bounding box coordinates \( B = \{b_i\}, i \in [1, \ldots, K_1] \). In order to build the edge connections, dominant literature leveraged a fully connected (FC) graph to learn arbitrary implicit relations under complex attention mechanisms. However, FC architecture loses explicit topology information and inlines to rapidly crash in oversmoothing during deep graph network training. Therefore, the graph edges \( E^m \) are developed as binary connections, where \( e^m_{ij} \in \{0, 1\} \) denotes whether the IoU of the object \( i \) and \( j \) exceeds a threshold. Since the visual objects in an image are independent and \( \mathcal{G}_1 \) relies on their spatial relations, the visual topology is sparse.

2) Question Graph Construction: Most of the existing VQA algorithms merely adopted question embedding as the semantic description of the questions and did not incorporate the syntactic patterns, which limited the ability of the VQA model to understand complex questions. Motivated by this, we propose to enhance the question representation by both the syntactic structure and internal semantic relevance by a question graph. We employ the dependency parsing algorithm [46] to analyze the grammar of question \( Q \) and output the syntax dependency tree. To be specific, the grammar of question \( Q \) is analyzed by dependency parsing and depicted by a syntax dependency tree. This tree is regarded as the question graph

\[
\mathcal{G}_2 = (V^n, E^n), \quad \{V^n = \{v^n_i\}_{i=1}^{K_2}, E^n = \{e^n_{ij} \in \mathbb{R}^{K_2 \times K_2}\} \quad (2)
\]

where \( K_2 \) is the number of words in \( Q \), each graph node \( v^n_i \in V^n \) indicates a word, and each edge \( e^n_{ij} \in E^n \) represents the syntax relations. \( \mathcal{G}_2 \) is a directed graph and its edges are also binary connections. In addition, \( v^n_i \) is associated with vector embedding, for example, produced by question \( Q \) going through a pretrained GloVe [47] and a following bidirectional dynamic gated recurrent unit (Bi-GRU). The embedding operation maps the word to a higher dimensional feature space for better contextual representation. The question graph fundamentally supports an effective semantic aggregation in the following GMA module that is shown in Section III-B.

B. Graph Matching Attention

Rather than fusing the visual and question features directly in previous works, the proposed GMA learns the bilateral matching relationships between the visual and textual graphs and aligns them by passing the cross-modal information to each other. Inside the GMA module, each graph node can be updated after receiving the cross-modal attention. The GMA module can be stacked for deeper level of semantic relationship features. Thus, it is convenient for GMA model to carry out reasonable cross-modality semantic inference. As shown in Fig. 3, the GMA module consists of two submodules, i.e., the dual-stage graph encoder and the bilateral matching attention module. The details are described as follows.

1) Dual-Stage Graph Encoder: The inputs of GMA module are the node features and the edge matrices of both visual graph and question graph, which have been built in Section III-A. The visual graph and question graph are constructed from two different modal spaces and have different topological structures, so there is a gap between them. Our dual-stage graph encoder transforms both graph features into the same graph matching subspace. Inside the encoder, we apply cascaded graph convolutional layers to successively infer the explicit and implicit relations inside each graph. The meaning of explicit stage is information aggregation and relational reasoning between graph nodes based on the clear spatial relationship in visual graph and syntactic relationship in question graph. However, the implicit stage infers according to the similarity relationship between graph node features, which further makes up for the insufficient reasoning of explicit stage. As shown in Fig. 3, the graph encoder contains two parallel branches to encode the visual graph and question graph separately. Each branch is composed of one FC layer and two graph convolutional layers.

In the explicit relationship inference stage, we first utilize the FC layer \( FC_{11} \) and \( FC_{21} \) to transform the features of two graph nodes \( V^m \) and \( V^n \) into feature matrix \( X \in \mathbb{R}^{K_1 \times d} \) and \( Y \in \mathbb{R}^{K_2 \times d} \), respectively

\[
X = \{x_i\}, \quad Y = \{y_j\}, \quad \{i \in [1, \ldots, K_1], j \in [1, \ldots, K_2]\}
\]

(3)

where \( d \) denotes the dimension of each node features. \( x_i \in \mathbb{R}^{1 \times d} \) indicates the encoded features of node \( i \) in the visual graph and \( y_j \in \mathbb{R}^{1 \times d} \) is the encoded features of node \( j \) in the question graph. Next, two graph convolutional layers (Gconv\(_{11}\) and Gconv\(_{21}\)) aggregate node features \( (X \) and \( Y ) \) according to the graph edge information \( E^m \) and \( E^n \), respectively. The parameters of Gconv\(_{11}\) and Gconv\(_{21}\) are shared. The Gconv\(_{11}\) layer for visual graph is formulated as

\[
X' = \sigma(W_1(X + W_2(L^n X)))
\]

(4)

where \( W_1 \) and \( W_2 \) are the layer-specific trainable weight matrix, \( \sigma(\cdot) \) denotes a nonlinear activation function, and \( L^n \) is a symmetric normalized Laplacian of size \( K_1 \times K_1 \), which
could be defined as

\[ L^m = (D^m)^{-1/2}E^m(D^m)^{1/2} \]

\[ D^m = \text{diag} \left( \sum_{j=1}^{K_1} e_{ij}, \ldots, \sum_{j=1}^{K_1} e_{ij}, \ldots, \sum_{j=1}^{K_1} e_{K_1,j} \right) \]

where \( D^m \) is a diagonal matrix and \( e_{ij} \) belongs to \( E^m \). A symmetric normalized Laplacian matrix is a variant of Laplacian matrix, which can effectively aggregate information between graph nodes according to the edge matrix of graph. Then, the aggregated question graph features \( Y' \in \mathbb{R}^{K_2 \times d} \) can be obtained by the Gconv\(_{21} \) layer in the same way. In this way, we obtain the explicit intramodality representation \( X' \) and \( Y' \).

Given \( X' \) and \( Y' \), the successive implicit relationship inference stage constructs connections between nodes by comparing their similarities. In this way, the intramodality messages among the graph nodes can be propagated. Depending on this propagation, our GMA module is capable of learning semantic relations inside the image and the question implicitly. In the adjacency layers (denoted as Adj and shown in Fig. 3), the adjacency matrices \( A^m \) of visual graph and \( A^n \) of question graph are learned by calculating the similarity inside the graph node features \( X' \) and \( Y' \), respectively. The adjacency layer for the visual graph can be formulated as

\[ a^m_{ij} = \exp \frac{S_{IM}(x'_i, x'_j)}{\sum_{j' \in K_1} \exp \frac{S_{IM}(x'_i, x'_{j'})}{2} } \]

\[ S_{IM}(x'_i, x'_{j'}) = \| x'_i - x'_{j'} \|_2^2 \]

where \( a^m_{ij} \in A^m \), \( x'_i, x'_{j'} \in X' \) and \( i, j \in [1, K_1] \). The function \( S_{IM} \) calculates the Euclidean distance between two nodes in a graph. The adjacency matrix \( A^m \in \mathbb{R}^{K_1 \times K_1} \) can be calculated in the same manner. In order to improve the understanding and reasoning ability of the VQA model, another two graph convolutional layers (Gconv\(_{12} \) and Gconv\(_{22} \)) transfer semantic relation information into the feature of each graph node according to the adjacency matrices. The parameters of Gconv\(_{12} \) and Gconv\(_{22} \) layers are also shared, and Gconv\(_{12} \) is calculated by

\[ X'' = X' + W_3 (A^m X') \]

where \( W_3 \in \mathbb{R}^{d \times d} \) is the layer-specific trainable weight matrix and \( X'' \in \mathbb{R}^{K_1 \times d} \) is the visual knowledge, which is outputted from the graph encoder. In a similar way, the textual knowledge \( Y'' \in \mathbb{R}^{K_2 \times d} \) is calculated as

\[ Y'' = Y' + W_3 (A^n X'). \]

By the dual-stage graph encoder, the two modality graphs can be termed as

\[ G_1' = \{ X'', E^m, A^m \} \]

\[ G_2' = \{ Y'', E^n, A^n \} \]

2) Bilateral Matching Attention: After the dual-stage graph encoder module, the bilateral matching attention module is proposed for cross-modal information interaction between two graphs. The nodes of the visual graph represent the visual features of all objects detected in the image, while the nodes of the question graph are the related word features of some specific objects in the question sentence. The two different modal features need to be aligned first and then further fused, so as to achieve more accurate reasoning in the process of cross-modal interaction. Inspired by the idea of graph matching algorithm, the proposed bilateral matching attention module first performs the matching operation between two graphs and then the designed attention algorithm according to the matching scores to complete the cross-modal feature fusion and relational reasoning. The proposed matching attention submodule consists of a graph matching layer and a bilateral cross-modality attention layer. In the graph matching layer, an affinity matching function \( f_a \) learns the graph matching score \( s_{ij} \) between the two nodes’ features \( x''_i \in X'' \) and \( y''_j \in Y''. \) The formula of the matching function \( f_a \) is defined as

\[ s_{ij} = f_a(x''_i, y''_j), \quad i \in [1, K_1], j \in [1, K_2]. \]
can be a bilinear mapping followed by an exponential function, which ensures all elements positive

\[
\begin{align*}
    s_{ij} &= \exp\left(\frac{x_i^T A_w y_j^T}{\tau}\right) \\
    &= \exp\left(\frac{x_i^T (A_w + A_w^T) y_j^T}{2\tau}\right)
\end{align*}
\]

\[
\{ \forall \ i \in [1, K_1], \ x_i'' \in \mathbb{R}^{1 \times d} \ \forall \ j \in [1, K_2], \ y_j'' \in \mathbb{R}^{1 \times d} \}
\] (13)

where \( \tau \) is a hyperparameter and \( A_w \in \mathbb{R}^{d \times d} \) denotes the learnable weights of the affinity function. From 13, the graph matching affinity matrix \( S = [s_{ij}] \in \mathbb{R}^{K_1 \times K_2} \) retains the matching score between two nodes coming from two different modalities. In the bilateral cross-modality attention submodule, we merge the cross-modality features in two ways (form \( X'' \) to \( Y'' \) and from \( Y'' \) to \( X'' \)) for interactive learning by applying a softmax function. It outputs two cross-modality attention maps, \( S_1 \in \mathbb{R}^{K_1 \times K_2} \) and \( S_2 \in \mathbb{R}^{K_1 \times K_1} \), which respectively represent the matching degree between the two cross-modality graph nodes. Therefore, the node features of the visual and textual modalities can be updated by the attention maps \( S_1 \) and \( S_2 \):

\[
\begin{align*}
    V'' &= W_4 (Y'' \oplus S_1 X'') \quad \text{(14)} \\
    V''' &= W_5 (X'' \oplus S_2 Y'') \quad \text{(15)}
\end{align*}
\]

where \( V'' \in \mathbb{R}^{K_1 \times d} \) and \( V''' \in \mathbb{R}^{K_2 \times d} \) are two updated graph nodes, and the operator \( \oplus \) represents matrix concatenation. \( W_4 \in \mathbb{R}^{d \times d} \) and \( W_5 \in \mathbb{R}^{d \times d} \) are the layer-specific trainable weight matrices.

### C. Answer Prediction Layer

After one or several stacked GMA modules, an answer prediction module concatenates the visual and textual modal information to predict the final answer. Graph node features \( V'' \) and \( V''' \) are integrated and generate a new representation \( H \in \mathbb{R}^{K_3 \times d} \) by

\[
H = \delta (V'' \oplus V''') \quad \text{(16)}
\]

where \( \delta \) denotes a nonlinear activation function, \( \oplus \) denotes the concatenation operation, and \( K_3 = K_1 + K_2 \). Then, \( H \) is operated by a maximum pooling and outputs the reasoning feature \( h \in \mathbb{R}^{1 \times d} \) along the node dimension. During question graph construction, we use Bi-GRU to embed question \( Q \) into question feature \( q \in \mathbb{R}^{1 \times d} \). \( q \) and \( h \) are merged by an elementwise production, and sent into a two-layer MLP to predict the scores of candidate answers. Formally,

\[
\text{scores} = \sigma (\text{MLP}(q \circ h)). \quad \text{(17)}
\]

\( N_a \) denotes the number of all possible answers, and the MLP has 2048 and \( N_a \) units in its first and second layer. In addition, we use the multilabel soft loss function to compute the soft target score of each class

\[
L(t, y) = \sum_i t_i \log \left( \frac{1}{1 + e^{-y_i}} \right) + (1 - t_i) \log \left( \frac{e^{-y_i}}{1 + e^{-y_i}} \right)
\]

(18)

where \( y \) is the output of our model, \( t = \text{(number of votes)/nc} \) denotes the target score of each class, and \( nc \) is the total answer category of the corresponding dataset, which can be found in Section IV-B. The detailed process of our GMA framework is expressed in Algorithm 1.

### IV. EXPERIMENTS

In this section, we conduct ablation studies to verify the effectiveness of each module in our proposed graph match attention network and make a comparison with state-of-the-art methods on the VQA 2.0 [48] and the GQA datasets [49].

#### A. Dataset and Metrics

1) VQA 2.0 Dataset: VQA 2.0 dataset [48] is built on top of the widely used VQA 1.0 dataset [17] and it contains a total of 1 105 904 questions and 204 721 images collected from the COCO dataset [51]. Each question has ten answers labeled by ten AMT workers. The sizes of train split, the val split, and the test split are 443 757, 214 354, and 447 739, respectively. The dataset has two types of questions, i.e., the open-ended and the multiple-choice questions. We focus on the open-ended task for common comparisons. After preprocessing the dataset, we take the answers that appear more than nine times in the train split as candidate answers and generate 3129 candidate answers.
answers as the final classification category. We train our model on the train split and evaluate on the val split. When testing on test-dev, train and val splits and extra Visual Genome dataset [52] are combined for training. VQA 2.0 datasets report accuracy on three categories of questions: yes/no, number, and other. The ground-truth score of an answer is determined by voting among ten human annotators

\[
\text{ACC(ans)} = \min\left(1, \frac{\text{#humans provided ans}}{3}\right).
\]

2) GQA Dataset: It is a new dataset for the VQA task over real-world images [49]. Compared with VQA 2.0 dataset, GQA needs more reasoning skills, such as spatial understanding and multistep reasoning. In the GQA dataset, 52% of the questions need the trained model to understand the visual relationship and conduct relational reasoning. The effectiveness of our model can be verified directly by reasoning on these complex questions. Apart from the standard accuracy metric measured on several different categories of questions, the dataset also provides some other metrics to better evaluate the performance of the model. The consistency metric is used to measure the level of consistency in responses across different questions. The validity metric measures whether the model gives valid answers, ones that can be theoretically correct for the question. The plausibility metric measures whether the model responses are reasonable in the real world or not making sense. The distribution metric measures the difference between the predicted answer distribution and the ground truth.

B. Experimental Implementation

1) Feature Preprocessing: For visual feature extraction, we adopt Faster RCNN [23] pretrained in Visual Genome [52] to detect object proposals, extracting their bounding boxes \( B \in \mathbb{R}^{K_1 \times 4} \) and the RoI features \( R \in \mathbb{R}^{K_1 \times 2048} \). Those features contain 10–100 object proposals per image and we pad the number of object proposals to the fixed size \( K_1 = 100 \). In the visual graph construction module, in order to make the graph node features contain spatial information, \( B \) is normalized by the image size and concatenated with \( R \) to form the visual graph nodes \( V^m \). In question processing, we set the sentence length to \( K_2 = 14 \) and embed it as a sequence of 300-D word vectors by a pretrained GloVe embedding function. Next, this embedded feature is sent into a Bi-GRU [54] layer to produce a global feature \( q \in \mathbb{R}^{1 \times 2048} \). Simultaneously, we adopt the dependency parsing algorithm [46] to analyze the syntax dependency tree in the given question. Based on the constructed tree, we then put all the related words of each word in the corresponding question together as a node of the question graph. Therefore, each graph node has a variable number of words. Next, we map those words into a sequence of GloVe word embedding features and also use the Bi-GRU module to extract the features \( V^c \in \mathbb{R}^{K_1 \times 2048} \).  

2) Hyperparameters: In the visual graph construction module, we set the IoU threshold to 0.3 to link sparse image regions. In the GMA module, all the hidden dimensions \( d \) are set to 2048. The hyperparameter \( \tau \) is set as \( 1 / (2028)^{1/2} \). The final answer category \( n_c \) of our GMA model is set to 3129 classes on the VQA 2.0 dataset and 1834 classes on the GQA dataset. During training, the dropout layer is applied to the pretrained GloVe word embedding features and question features with a 0.25 probability and is applied to image features and final reasoning features with a 0.5 probability. The model is trained by Adamax optimizer for 35 epochs with a batch size of 256. The learning rate is initialized as \( 5e^{-4} \) and linearly adjusted by \( 2e^{-3} \) at epoch 4. After 25 epochs, the learning rate is half decreased every two epochs.

C. Ablation Study of GMA

In this section, we conduct ablation studies on the VQA 2.0 val split to verify the validity of each part of our GMA network. We also compare the effects of different hyperparameters on the GMA model by visualization. In addition, we conduct ablation studies on the GQA val split to verify the effectiveness of stacked GMA modules.

1) Comparison to Attention-Based Model: The goal of those experiments is to compare our approach with strong baselines for real VQA in controlled conditions. As shown in Table I, the default setting of our network contains only one GMA module (GMA-1). Also, we compare our GMA network with several famous attention-based methods, i.e., the BUTD [1], the BAN [4], and the DFAF [5]. For fair comparisons, we use the same bottom-up features to insure competitiveness. In terms of the attention mechanism comparison, our GMA-3 outperforms the BAN-12 by 1.43% and DFAF-8 by 0.87% on the accuracy metric. Our GMA model reaches a higher accuracy than those three attention-based

| Component | Setting | Accuracy |
|-----------|---------|----------|
| BUTD [1]  | Bottom-up | 63.37    |
| BAN [4]   | BAN-1   | 65.36    |
|           | BAN-4   | 65.81    |
|           | BAN-12  | 66.04    |
|           | DFAF-1  | 66.21    |
|           | DFAF-5  | 66.58    |
|           | DFAF-8  | 66.60    |
| Default   | GMA-1   | 66.80    |
| # of stacked blocks | GMA-2 | 67.34 |
|           | GMA-3   | 67.47    |
|           | GMA-4   | 67.41    |
| Part of attention | Bilateral Matching Attention | 66.36 |
| Part of graph encoder | only Explicit Stage | 66.67 |
|           | only Implicit Stage | 66.64 |
|           | Dual-Stage | 66.80 |
| Cross-graph feature fusion | Addition | 66.56 |
| Embedding | Concatenation | 66.80 |
| Bounding Box | Absolute Position | 66.67 |
| Embedding | Normalized Position | 66.80 |
methods on the VQA 2.0 val split. It demonstrates that our GMA network answers questions correctly and generates more effective relational reasoning.

2) Number of Stacked GMA Modules: In order to carry out deeper relationship reasoning, we stack several GMA modules. In this section, we investigate the influence of the number of the stacked GMA modules on the VQA 2.0 dataset. As shown in the fifth setting (“# of stacked blocks”) of Table I, compared with GMA-1, stacking multiple GMA modules improves the performance. For example, GMA-2 improves the results by 0.54%, and GMA-3 gains a further improvement of 0.67%. With the growing number of stacked GMA modules, GMA-4 turns up with a slightly performance degradation of 0.06%, compared with GMA-3. It can be interpreted as the emergency of oversmoothing when the message passes in deep GCNs. We also verify the effectiveness of the stacking operation on the GQA dataset, as shown in Table II. GMA-3 overperforms BUTD [1] and BAN-1 [4] by 2.99% and 1.0%, respectively. The large gaps validate the advantage of our GMA mechanism. Then, we design fine granularity studies on the part of graph encoder. As shown in the seventh component of Table I, the result of the explicit stage graph encoder was improved by 0.31% since the explicit structure information in the two modalities can be integrated into each graph node. Besides, merely using the implicit stage graph encoder, the performance improves 0.28%. Since the implicit relationships between each graph node have been learned, the performance can be promoted by the matching attention submodule. Moreover, the dual-stage graph encoder can be enhanced by 0.43%, indicating the complementarity between the explicit and the implicit graph structure information.

3) Components of GMA Module: In order to validate the effectiveness of each part of our GMA module, some modules are deleted from the GMA-1 model. We first delete the dual-stage graph encoder submodule and the result is reported in the sixth setting of Table I. Compared with the default setting, the performance without the dual-stage graph encoder decreases by 0.44%, reflecting the significant role of intramodality reasoning. Because the bilateral matching attention submodule can intuitively align features between

methods on the VQA 2.0 val split. It demonstrates that our GMA network answers questions correctly and generates more effective relational reasoning.

2) Number of Stacked GMA Modules: In order to carry out deeper relationship reasoning, we stack several GMA modules. In this section, we investigate the influence of the number of the stacked GMA modules on the VQA 2.0 dataset. As shown in the fifth setting (“# of stacked blocks”) of Table I, compared with GMA-1, stacking multiple GMA modules improves the performance. For example, GMA-2 improves the results by 0.54%, and GMA-3 gains a further improvement of 0.67%. With the growing number of stacked GMA modules, GMA-4 turns up with a slightly performance degradation of 0.06%, compared with GMA-3. It can be interpreted as the emergency of oversmoothing when the message passes in deep GCNs. We also verify the effectiveness of the stacking operation on the GQA dataset, as shown in Table II. GMA-3 overperforms BUTD [1] and BAN-1 [4] by 2.99% and 1.0%, respectively. The large gaps validate the advantage of our GMA mechanism. Then, we design fine granularity studies on the part of graph encoder. As shown in the seventh component of Table I, the result of the explicit stage graph encoder was improved by 0.31% since the explicit structure information in the two modalities can be integrated into each graph node. Besides, merely using the implicit stage graph encoder, the performance improves 0.28%. Since the implicit relationships between each graph node have been learned, the performance can be promoted by the matching attention submodule. Moreover, the dual-stage graph encoder can be enhanced by 0.43%, indicating the complementarity between the explicit and the implicit graph structure information.

3) Components of GMA Module: In order to validate the effectiveness of each part of our GMA module, some modules are deleted from the GMA-1 model. We first delete the dual-stage graph encoder submodule and the result is reported in the sixth setting of Table I. Compared with the default setting, the performance without the dual-stage graph encoder decreases by 0.44%, reflecting the significant role of intramodality reasoning. Because the bilateral matching attention submodule can intuitively align features between

methods on the VQA 2.0 val split. It demonstrates that our GMA network answers questions correctly and generates more effective relational reasoning.

2) Number of Stacked GMA Modules: In order to carry out deeper relationship reasoning, we stack several GMA modules. In this section, we investigate the influence of the number of the stacked GMA modules on the VQA 2.0 dataset. As shown in the fifth setting (“# of stacked blocks”) of Table I, compared with GMA-1, stacking multiple GMA modules improves the performance. For example, GMA-2 improves the results by 0.54%, and GMA-3 gains a further improvement of 0.67%. With the growing number of stacked GMA modules, GMA-4 turns up with a slightly performance degradation of 0.06%, compared with GMA-3. It can be interpreted as the emergency of oversmoothing when the message passes in deep GCNs. We also verify the effectiveness of the stacking operation on the GQA dataset, as shown in Table II. GMA-3 overperforms BUTD [1] and BAN-1 [4] by 2.99% and 1.0%, respectively. The large gaps validate the advantage of our GMA mechanism. Then, we design fine granularity studies on the part of graph encoder. As shown in the seventh component of Table I, the result of the explicit stage graph encoder was improved by 0.31% since the explicit structure information in the two modalities can be integrated into each graph node. Besides, merely using the implicit stage graph encoder, the performance improves 0.28%. Since the implicit relationships between each graph node have been learned, the performance can be promoted by the matching attention submodule. Moreover, the dual-stage graph encoder can be enhanced by 0.43%, indicating the complementarity between the explicit and the implicit graph structure information.

3) Components of GMA Module: In order to validate the effectiveness of each part of our GMA module, some modules are deleted from the GMA-1 model. We first delete the dual-stage graph encoder submodule and the result is reported in the sixth setting of Table I. Compared with the default setting, the performance without the dual-stage graph encoder decreases by 0.44%, reflecting the significant role of intramodality reasoning. Because the bilateral matching attention submodule can intuitively align features between

methods on the VQA 2.0 val split. It demonstrates that our GMA network answers questions correctly and generates more effective relational reasoning.

2) Number of Stacked GMA Modules: In order to carry out deeper relationship reasoning, we stack several GMA modules. In this section, we investigate the influence of the number of the stacked GMA modules on the VQA 2.0 dataset. As shown in the fifth setting (“# of stacked blocks”) of Table I, compared with GMA-1, stacking multiple GMA modules improves the performance. For example, GMA-2 improves the results by 0.54%, and GMA-3 gains a further improvement of 0.67%. With the growing number of stacked GMA modules, GMA-4 turns up with a slightly performance degradation of 0.06%, compared with GMA-3. It can be interpreted as the emergency of oversmoothing when the message passes in deep GCNs. We also verify the effectiveness of the stacking operation on the GQA dataset, as shown in Table II. GMA-3 overperforms BUTD [1] and BAN-1 [4] by 2.99% and 1.0%, respectively. The large gaps validate the advantage of our GMA mechanism. Then, we design fine granularity studies on the part of graph encoder. As shown in the seventh component of Table I, the result of the explicit stage graph encoder was improved by 0.31% since the explicit structure information in the two modalities can be integrated into each graph node. Besides, merely using the implicit stage graph encoder, the performance improves 0.28%. Since the implicit relationships between each graph node have been learned, the performance can be promoted by the matching attention submodule. Moreover, the dual-stage graph encoder can be enhanced by 0.43%, indicating the complementarity between the explicit and the implicit graph structure information.

3) Components of GMA Module: In order to validate the effectiveness of each part of our GMA module, some modules are deleted from the GMA-1 model. We first delete the dual-stage graph encoder submodule and the result is reported in the sixth setting of Table I. Compared with the default setting, the performance without the dual-stage graph encoder decreases by 0.44%, reflecting the significant role of intramodality reasoning. Because the bilateral matching attention submodule can intuitively align features between
layers. Also, the addition strategy simply fuses the features of two graphs by addition operation and also applies the full connection layers. The final results show that the concatenation strategy for cross-graph feature fusion is better than the addition strategy. Besides, we also test several embedding methods of the bounding box to improve the performance of our GMA model. In the ninth component of Table I, for the bounding box embedding, the normalized position embedding can get the best performance compared with no bounding box embedding and absolute position embedding. At last, we investigate the hyperparameters settings (as shown in Fig. 4). The model is trained on the VQA 2.0 train split for 35 epochs with a batch size of 256 and evaluated on the val split. Fig. 4(a) shows the performance of IoU threshold used in the graph construction module (III-A), and Fig. 4(b) shows the result of the dimension $d$ in all hidden layers. It can be found that the network performs best when the IoU threshold is set to 0.3 and dimension $d$ is set to 2048.

D. Comparison With State-of-the-Art Methods

1) Performance on the VQA 2.0 Dataset: Table III shows the comparison results of our proposed network and the state of the arts on the VQA 2.0 test split. The results demonstrate that our GMA mechanism improves the performance over the baseline BUTD [1] by 4.49% in terms of test-std accuracy. We also compare our model with some excellent attention-based approaches, such as SAN [20], FDA [55], Co-attention [21], RelAtt [56], DCN [57], RAF [58], and BAN + Glove [4]. In Table III, our GMA outperforms these methods in all the metrics, which proves that the proposed GMA mechanism is more effective in the VQA task. Since our model builds graph structure for both the image and the question representation, we compare our model with the graph- and tree-based approaches, such as GraphLearner [14], SceneGCN [15], and VCTREE-HL [59], which merely built graph on visual modality. Our GMA method overwhelms these methods, which verifies that the cross-modality graph-based representation is feasible. Finally, we make a comparison with state-of-the-art relationship learning methods, including Free_VQA [61], BLOCK [62], SEADA [65], MuRel [24], Counter [63], and DF [64]. The results show that our accuracy score on the test-dev split overwhelms these methods.

2) Performance on the GQA Dataset: Table IV shows the results of our network on the GQA test split. We compare our GMA model with both baselines as well as state-of-the-art methods. The baselines include local prior, global

|                | Global Prior | Local Prior | CNN+LSTM | BUTD     | MAC      | SceneGCN | LCGN | GRNs | Ours(GMA) |
|----------------|--------------|-------------|----------|----------|----------|----------|------|------|-----------|
| Open           | 16.52        | 16.99       | 31.8     | 34.83    | 38.91    | 40.63    | -    | 41.24| 42.3      |
| Binary         | 42.99        | 47.53       | 63.26    | 66.64    | 71.23    | 70.33    | -    | 74.93| 73.2      |
| Distribution   | 130.86       | 21.56       | 7.46     | 5.98     | 5.34     | 6.43     | -    | 5.76 | 5.56      |
| Validity       | 89.02        | 84.44       | 96.02    | 96.18    | 96.16    | 95.9     | -    | 96.14| 96.41     |
| Plausibility   | 75.34        | 84.42       | 84.25    | 84.57    | 84.48    | 84.23    | -    | 84.68| 85.05     |
| Consistency    | 51.78        | 54.34       | 74.57    | 78.71    | 81.59    | 83.49    | -    | 87.41| 83.95     |
| Accuracy       | 28.93        | 31.31       | 46.55    | 49.74    | 54.06    | 54.56    | 56.10| 57.04| 57.26     |
prior, and CNN + LSTM, which are introduced by the GQA dataset [49]. Specifically, based on the question group, the local and global prior models return the most common answer for each group. The CNN + LSTM model is simple fusion approaches between the text and the image. As we can see in Table IV, the performance of our GMA model is much better than those baselines. In terms of accuracy metric, our GMA model is 28.33% higher than that of local prior model, 25.95% higher than that of global prior model, and 10.71% higher than the CNN + LSTM model. The results verify that our method is effective in solving VQA tasks on the GQA dataset.

Next, we compare with several attention-based approaches, including BUTD [1] and MAC [37]. Our GMA model exceeds the BUTD model by 7.52% and the MAC model by 3.20%, which demonstrates that our graph match attention mechanism is available for cross-modality alignment and can effectively carry out feature fusion between different modalities. Finally, we compare our GMA model with several state-of-the-art graph-based methods, such as SceneGCN [15], LCGN [53], and GRNs [66]. These methods applied graph message passing strategy and performed relational reasoning. Our model works better than SceneGCN and LCGN in all metrics, with the overall accuracy 2.7% and 1.16% higher than the two, respectively. Besides, the performance of our GMA network is comparable with GRNs [66].

E. Visualization

In this section, we visualize the qualitative results of GMA model on the GQA dataset, as shown in Fig. 5. Each row in the figure corresponds to the prediction results of GMA model with different questions on the same image. The left side of each image represents the question, and the bottom side represents the ground truth and predicted answer. According to the calculated GMA map, we visualize each word in the question and the three most relevant objects in the image. From Fig. 5, the GMA can effectively understand complex questions, find the target objects associated with

![Qualitative results of our GMA network on the VQA 2.0 dataset. GMA #1, GMA #2, and GMA #3 denote the intermediate results of the GMA-3 model and predictions are the final answers to the output of the whole model.](image1)

![Comparison of visualization results with state-of-the-art methods on the GQA dataset. Our (GMA) method is compared with SceneGCN and LCGN by using the same images and questions. The bottom of each image represents the prediction answers.](image2)
the question in the image, and then can correctly answer the question through effective relational reasoning. To better illustrate the effectiveness of our GMA mechanism, we also visualize the learned information in each GMA module. Fig. 6 shows the results of our trained GMA-3 model on the VQA 2.0 dataset. It finds the most relevant nodes in the question graph according to the index of the last global pooling layer (corresponding to the orange words in the input questions). We illustrate the top 3 visual objects in a visual graph, which has a high correlation value (objects drawn by the red rectangles), depending on the graph match attention maps. According to the edge and affinity matrices of the visual graph, we visualize the explicit relationships (the white solid line) and the implicit relationships (the white dotted line) between the objects in a given image. Our GMA model can find textual keywords in the questions and accurately locate visual objects in the images. As the GMA layers are stacked, it learns much deeper relational semantics for answering questions. At last, we visualize the comparison between GMA and state-of-the-art methods on the GQA dataset in Fig. 7. The comparison results demonstrate that our GMA network correctly solves some questions by deeper relationship reasoning.

V. CONCLUSION

In this article, we have proposed a new GMA module to solve the problem of cross-modality feature fusion and relational reasoning in the VQA task by using a graph matching network. Our network not only uses the whole question embedding feature but also analyzes the syntax structure of question and constructs the question graph. We apply the graph-structure representation in both the visual and textual modalities. Then, the GMA module learns the relationships inside the visual and textual features and propagates the cross-modality information mutually. By stacking multiple blocks of GMA, the inference ability will be enhanced on the VQA benchmarks. The experimental data and the attention-based visualization provide strong evidence that our model has the capability in multimodal feature fusion and relation reasoning.

REFERENCES

[1] P. Anderson et al., “Bottom-up and top-down attention for image captioning and question answering,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 6077–6086.

[2] Y. Jiang, V. Natarajan, X. Chen, M. Rohrbach, D. Batra, and D. Parikh, “Pythia v0.1: The winning entry to the VQA challenge 2018,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2018.

[3] A. Vaswani et al., “Attention is all you need,” in Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS), 2017, pp. 5998–6008.

[4] I.-H. Kim, J. Jun, and B.-T. Zhang, “Bilinear attention networks,” in Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS), 2018, pp. 1–11.

[5] P. Gao et al., “Dynamic fusion with intra- and inter-modality attention flow for visual question answering,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 6639–6648.

[6] Z. Yang, J. Yu, C. Yang, Z. Qin, and Y. Hu, “Multi-modal learning with prior visual relation reasoning,” in Proc. ACM Multimedia (MM), Dec. 2018.

[7] P. Wang, Q. Wu, J. Cao, C. Shen, L. Gao, and A. V. D. Hengel, “Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 1960–1968.

[8] M. Rabah, R. Herzig, J. Berant, G. Chechik, and A. Globerson, “Differentiable scene graphs,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Mar. 2020, pp. 1488–1497.

[9] R. Yu, A. Li, V. I. Morariu, and L. S. Davis, “Visual relationship detection with internal and external linguistic knowledge distillation,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 1974–1982.

[10] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell, “Compact bilinear pooling,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 317–326.

[11] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, “Multimodal compact bilinear pooling for visual question answering and visual grounding,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2016, pp. 1–12.

[12] J. H. Kim, K. W. On, W. Lim, J. Kim, J. W. Ha, and B. T. Zhang, “Hadamard product for low-rank bilinear pooling,” in Proc. ICLR, 2017, pp. 1–14.

[13] H. Ben-younes, R. Cadene, M. Cord, and N. Thome, “MUTAN: Multimodal Tucker fusion for visual question answering,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2612–2620.

[14] W. Norcliffe-Brown, E. Vafeias, and S. Parisot, “Learning conditioned graph structures for interpretable visual question answering,” in Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS), 2018, pp. 8344–8353.

[15] Z. Yang, Z. Qin, J. Yu, and Y. Hu, “Scene graph reasoning with prior visual relationship visual question answering,” in Proc. ACM Multimedia (MM), Dec. 2019.

[16] S. Yang, G. Li, and Y. Yu, “Graph-structured referring expression reasoning in the wild,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 9952–9961.

[17] S. Antol et al., “VQA: Visual question answering,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 2425–2433.

[18] L. Nie, M. Wang, Z. Zha, G. Li, and T.-S. Chua, “Multimedia answering: Enriching text QA with media information,” in Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. (SIGIR), 2011, pp. 695–704.

[19] W. He et al., “Multimodal dialogue systems via capturing context-aware dependencies of semantic elements,” in Proc. 28th ACM Int. Conf. Multimedia, Oct. 2020, pp. 2755–2764.

[20] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, “Stacked attention networks for image question answering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 21–29.

[21] J. Lu, J. Yang, D. Batra, and D. Parikh, “Hierarchical question-image co-attention for visual question answering,” in Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS), 2016, pp. 289–297.

[22] Q. Meng, W. Wang, T. Zhou, J. Shen, Y. Jia, and L. Van Gool, “Towards a weakly supervised framework for 3D point cloud object detection and annotation,” IEEE Trans. Pattern Anal. Mach. Intell., early access, Mar. 3, 2021, doi: 10.1109/TPAMI.2021.3063611.2021.

[23] R. Yu, A. Li, V. I. Morariu, and L. S. Davis, “Visual relationship detection with prior visual relation reasoning,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2020, pp. 379–396.

[24] L. Chen, X. Yan, J. Xiao, H. Zhang, S. Pu, and Y. Zhuang, “Counterfactual samples synthesizing for robust visual question answering,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 10800–10809.

[25] M. Zhang, X. Liu, W. Liu, A. Zhou, H. Ma, and T. Mei, “Multi-granularity reasoning for social relation recognition from images,” in Proc. IEEE Int. Conf. Multimedia Expo. (ICME), Jul. 2019, pp. 1618–1623.

[26] R. C. Moore, “Abstract reasoning about knowledge and action,” Readings Artif. Intell., vol. 41, no. 2, pp. 473–477, 1981.

[27] J. Yu, M. Tan, H. Zhang, D. Tao, and Y. Rui, “Hierarchical deep click feature prediction for fine-grained image recognition,” IEEE Trans. Pattern Anal. Mach. Intell., early access, Jul. 30, 2021, doi: 10.1109/TPAMI.2021.3093611.

[28] J. Yu, C. Zhu, J. Zhang, Q. Huang, and D. Tao, “Spatial pyramid-enhanced NetVLAD with weighted triplet loss for place recognition,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 2, pp. 661–674, Feb. 2020.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
