Sun Q, Fu YW, Xue XY. Learning a mixture of conditional gating blocks for visual question answering. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 39(4): 912-928 July 2024. DOI: 10.1007/s11390-024-2113-0

Learning a Mixture of Conditional Gating Blocks for Visual Question Answering

Qiang Sun[1, 2] (孙　强), Yan-Wei Fu[3] (付彦伟), *Member, ACM*, and Xiang-Yang Xue[4] (薛向阳), *Member, CCF*

[1] *School of Statistics and Information, Shanghai University of International Business and Economics, Shanghai 201620, China*

[2] *Academy for Engineering and Technology, Fudan University, Shanghai 200433, China*

[3] *School of Data Science, Fudan University, Shanghai 200433, China*

[4] *School of Computer Science, Fudan University, Shanghai 200433, China*

E-mail: sunq18@fudan.edu.cn; yanweifu@fudan.edu.cn; xyxue@fudan.edu.cn

Received December 26, 2021; accepted January 24, 2024.

Abstract As a Turing test in multimedia, visual question answering (VQA) aims to answer the textual question with a given image. Recently, the "dynamic" property of neural networks has been explored as one of the most promising ways of improving the adaptability, interpretability, and capacity of the neural network models. Unfortunately, despite the prevalence of dynamic convolutional neural networks, it is relatively less touched and very nontrivial to exploit dynamics in the transformers of the VQA tasks through all the stages in an end-to-end manner. Typically, due to the large computation cost of transformers, researchers are inclined to only apply transformers on the extracted high-level visual features for downstream vision and language tasks. To this end, we introduce a question-guided dynamic layer to the transformer as it can effectively increase the model capacity and require fewer transformer layers for the VQA task. In particular, we name the dynamics in the Transformer as Conditional Multi-Head Self-Attention block (cMHSA). Furthermore, our question-guided cMHSA is compatible with conditional ResNeXt block (cResNeXt). Thus a novel model mixture of conditional gating blocks (McG) is proposed for VQA, which keeps the best of the Transformer, convolutional neural network (CNN), and dynamic networks. The pure conditional gating CNN model and the conditional gating Transformer model can be viewed as special examples of McG. We quantitatively and qualitatively evaluate McG on the CLEVR and VQA-Abstract datasets. Extensive experiments show that McG has achieved the state-of-the-art performance on these benchmark datasets.

Keywords visual question answering, Transformer, dynamic network

1 Introduction

Humans have the ability to holistically understand images. We can answer questions about images such as identifying the objects and inferring high-level attributes, spatial positions, or relationships of these objects. By narrowing down the scope of these questions above, the corresponding problems can be formulated as various classical multimedia and computer vision research tasks, such as object detection, or object attribute learning. For these tasks, tremendous advancements have been made by deep neural networks recently. Thus it is desirable to further advance the research toward a general multimedia system that can answer arbitrary language questions about images. As one of the most challenging tasks, visual question answering (VQA)[1] is defined as automatically inferring answers to the questions given the image. The VQA task demands a generic model in holistically understanding both the image and question sentences.

With great computational efficiency and scalability, the Transformer architecture has been successfully scaled and is topical in natural language processing (NLP) tasks[2, 3]. Such efficiency of Transformer has also inspired recent vision studies on

Regular Paper

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62176061 and the Science and Technology Commission of Shanghai Municipality under Grant No. 22511105000.

©Institute of Computing Technology, Chinese Academy of Sciences 2024

classification[4], detection[5], and segmentation[6]. Unfortunately, due to the prohibitive computation cost of transformers with deep layers, most of the transformer-based VQA models[7, 8] only employ transformers at the high-level stages to further process the image features extracted from the pre-trained backbones, not in an end-to-end manner. To explore the feasibility of end-to-end transformers in VQA, PixelBERT[9] connects the convolutional neural network (CNN) visual backbone to the transformer to align the pixel and language in an end-to-end way for the first time. This simple and direct solution has shown that end-to-end training has advantages in the performance of the model, although a huge computation resource is needed for training. For the efficiency of the end-to-end transformer model in VQA, Vision-and-Language Transformer (ViLT)[10] replaces the CNN visual backbone with a lightweight module, linear projection of flattened patches, to reduce the computation cost of the visual backbone. However, the CNN-free architecture has made ViLT lose the feature of translation invariance in vision.

Recently, benefited from adaptability, interpretability, and model capacity, dynamic networks[11–14] have been extensively studied for visual reasoning in VQA. Generally, it is roughly characterized as neural module networks and dynamic parameter networks. The former ones[11, 15, 16] generate the layout of neural modules according to the question and execute the visual and textual input on it to predict the answer. In contrast, the latter ones[12, 17, 18] propose dynamic blocks, such as dynamic kernel[18] and layer[12], which are designed as the basic cells for the networks and are normally trained in an end-to-end manner.

A natural question arises: can we enhance the adaptability, interpretability, and model capacity of the transformer-based VQA model by incorporating dynamic parameter networks? The answer is affirmative. First, VQA is a task that demands varying reasoning abilities based on the question type. The utilization of dynamic parameter networks can make the model more flexible in adapting to different question types. Second, the dynamic parameter network offers better interpretability by enabling analysis of the activation or representation of the dynamic parameters. Third, the incorporation of dynamic parameter networks can significantly increase the model capacity, reducing the number of required transformer layers for the VQA model and making end-to-end training possible.

With this aim in mind, we propose a novel conditional gating transformer network for VQA tasks, learned in an end-to-end manner. Essentially, our model learns a "dynamic" transformer by the Conditional Multi-Head Self-Attention Block cMHSA. Particularly, cMHSA is a conditional multi-head self-attention block with a bottleneck and residual connection. The conditional channel gating layer, placed before the ReLU layer, brings the dynamic parameters to cMHSA. The channel gating layer serves as a dynamic parameter implementation, providing efficiency in reducing memory footprint and computation cost. The integration of the conditional layer and ReLU layer has the potential to increase the diversity of activation functions and improve the adaptability of the model. The dynamic parameters learned in the gating layer can be utilized to reflect the various processes adopted by the model.

We also enhance the CNN backbone by incorporating the conditional layer in the ResNeXt[19] block, resulting in cResNeXt. cMHSA and cResNeXt are designed to work seamlessly together as shown in Fig.1. As a result, we present a mixture of conditional gating blocks (McG), which combines cResNeXt and cMHSA. By incorporating both CNN and Transformer components, McG gets the best of both worlds. The CNN backbone excels at modeling local patterns and translation invariance, while the Transformer is highly effective at modeling global feature dependencies. This combination results in a more robust and versatile model that can effectively handle a wide range of questions.

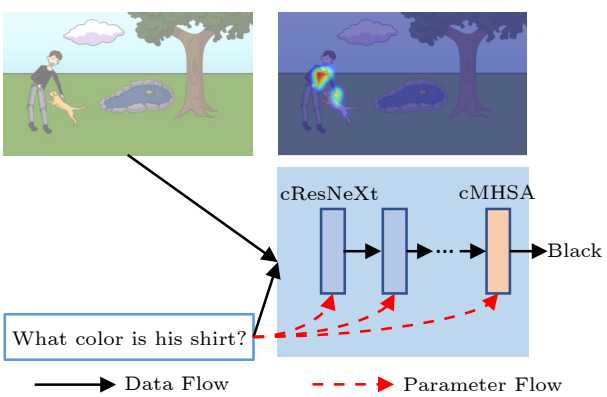


Fig.1. Concept illustration. Our approach uses the question to generate channel gates as dynamic parameters for cResNeXt and cMHSA in the network. The image and question are fed into the customized network to predict the answer.

McG has achieved state-of-the-art performance on both the CLEVR[20] and VQA-Abstract[1] datasets. We make a comprehensive study on different condi-

tional layers (conditional sigmoid, softmax, affine, and DyReLU[21]) and different settings of cResNeXt and cMHSA.

To summarize, our contributions are as follows:

1) introduction of a novel conditional gating transformer network for VQA, incorporating cMHSA to improve adaptability to different questions,

2) proposal of a novel VQA approach that combines the strengths of transformers and CNN by learning a mixture of conditional gating blocks of cMHSA and cResNeXt,

3) adoption of dynamic parameters to enhance the capacity of McG, which requires only a few transformer layers and enables feasible end-to-end training at a moderate computation cost,

4) improved model interpretability of McG through analysis of gating values and attention maps, and

5) demonstration of state-of-the-art performance on the CLEVR and VQA-Abstract benchmarks through extensive experiments.

2 Related Work

2.1 Visual Question Answering

The goal of visual question answering is to answer open-ended questions with given images. To facilitate advancements in this field, numerous datasets and methodologies have been put forth.

From the perspective of datasets, VQA benchmarks can be classified into two categories: real-image based and synthesized-image based. The VQA benchmarks with real images[1, 22, 23] are built on the images that contain multiple objects and rich contextual information, which require strong abilities in both perception and reasoning[24–26]. The VQA benchmarks with synthesized images are built upon the clipart pictures[1, 22] or synthesized images[20]. They are more focused on multi-step reasoning at high-level features. The pictures in VQA-Abstract[1, 22] contain clipart dolls in both indoor and outdoor scenes. CLEVR[20] has images composed of various objects in different shapes and colors.

From the perspective of methods, VQA models can be broadly categorized into static and dynamic networks. In static networks, the VQA models first extract the visual and textual features through off-the-shelf backbones[27], and then fuse them through high-order interaction[28, 29] or attention mechanism[24, 30]. The causal effect inference[31–33] has been utilized for removing the language bias for the VQA models. The most prevalent method now is the vision-language transformer[7, 8, 34–36], which dominates the VQA[7, 8, 34], image caption[35] and crossmodal retrieval[36] tasks. In contrast, the dynamic networks[11, 12, 15, 37] customize the visual backbone by the textual features, which can influence the visual feature extraction from low-level stages. In our paper, McG adopts the advantage of the Transformer, CNN, and dynamic networks. Our model is validated on the CLEVR and VQA-Abstract datasets, and we show the reasoning ability of our McG model.

2.2 Transformer

Transformer[2] uses attention to capture global dependencies among features for contextualization and has achieved great success in both NLP[3, 38, 39] and CV[4–6, 40] domains. Cross-modality transformers were introduced into VQA, such as ViLBERT[7] and LXMERT[8]. Concretely, ViLBERT and LXMERT treat the pre-extracted high-level visual features as "visual words", and perform both inter-modality and intra-modality attention on the vision and language features. However, using the pre-extracted visual features makes the transform not able to benefit from the representation power of end-to-end learning. Recently, end-to-end transformers[9, 10] have been proposed in VQA. Particularly, Pixel-BERT[9] connects the visual backbone to the transformer for end-to-end training but lead to a large computation cost. ViLT[10] treats the image as multiple patches, and linearly projects them into "visual words" to decrease the computation cost of the visual backbone. Pixel-BERT and ViLT use six or 12 transformer layers to contextualize the visual and language embeddings. In contrast, our proposed McG model utilizes only two transformer layers and increases model capacity by incorporating dynamic parameters through conditional layers. Rather than treating questions as merely input data, McG treats them also as control messages that influence network parameters, which has shown to be effective for visual reasoning. Moreover, McG can be trained from scratch without pretraining.

2.3 Dynamic Networks

Dynamic networks can enhance the adaptability, interpretability, and capacity of networks without substantially increasing the model size by dynamic architecture and dynamic parameter paradigms. In the

dynamic-architecture models, early-exit[41, 42], skip-layer[43, 44], mixture-of-experts methods[45–47] and composition of neural modules[11, 15] are widely used. However, due to the discrete nature of dynamic-architecture models, the training process is hard and tricky. In the dynamic-parameter models, a hyper network[48] is adopted to generate or adjust the parameters of the main network. The whole network is trained in an end-to-end manner. Similar to the dynamic parameter model, weight-sharing neural architecture search also aims to optimize the super network; however, it optimizes the network as a whole not conditioned on the input at the instance level.

Dynamic-architecture models in VQA are mainly proposed to build a composition network based on predefined neural modules[11, 15, 37], which usually requires the prior domain knowledge of basic operation blocks and the ground truth layout annotation of the training examples. Specifically, QGMRN[13] builds a super module network based on ResNeXt and discretely routes on ResNeXt, requiring neither prior domain knowledge nor ground truth layout annotations. Compared with QGMRN, we also adopt the ResNeXt architecture in the early stage. However, we use cMHSA in the last stage, which is critical in visual reasoning. Moreover, McG is based on channel gating rather than discrete routing.

Dynamic-parameter models in VQA use the hyper-network to infer the parameters of a full connection layer[17, 49], a convolution kernel[18] or a linear modulation layer[12]. Specifically, FiLM[12] successfully uses the channel-wise linear modulation layer to influence the whole network. The most related work to the proposed McG is FiLM. Compared with FiLM, the pioneering work for channel-wise condition layer in CNN, McG extends the channel gating layer to the MHSA block. Furthermore, it is demonstrated that the deprecated sigmoid function in FiLM is the most suitable for the conditional layer in McG.

3 Method

In this section, we detail our proposed McG, which leverages conditional Transformer and conditional CNN to reason on the images. Consistent with previous work[12, 13], McG consists of three components: the textual network, the visual network, and the classifier as illustrated in Fig.2. First, the textual network produces the question feature from the input question. Second, the question feature is mapped into the parameters of the conditional block in the visual network. Third, the question-customized visual network extracts the visual features from the raw image. Finally, the classifier utilizes both the visual feature

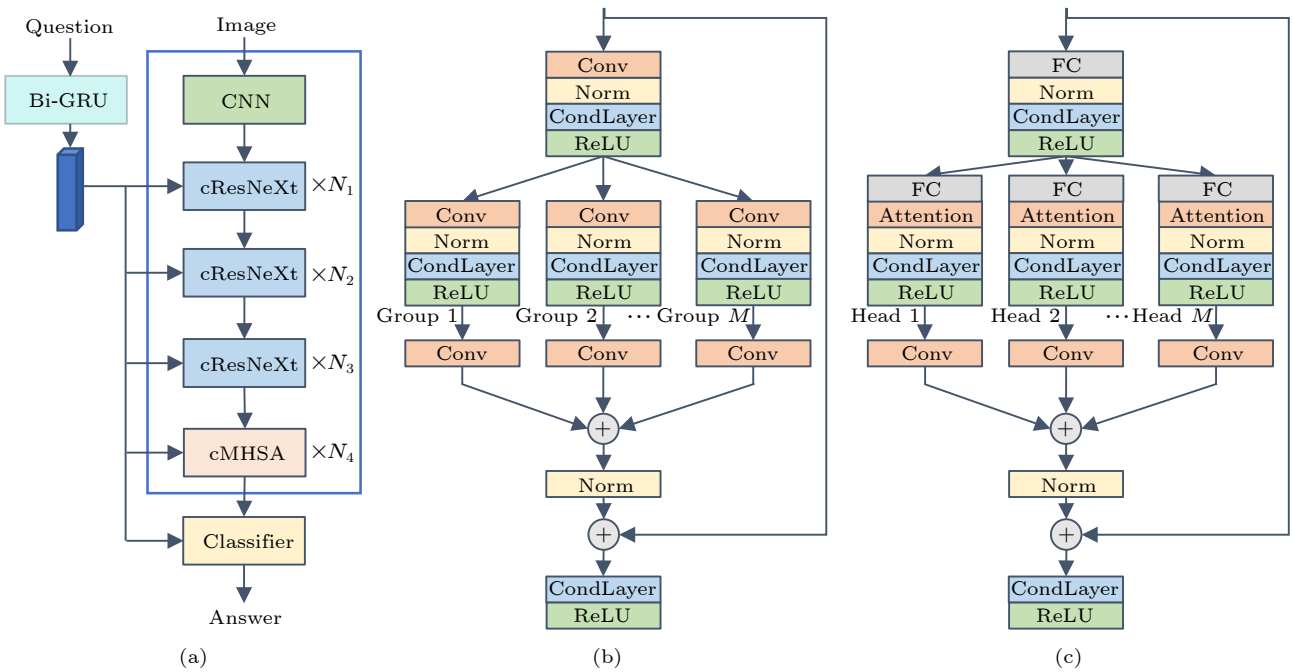


Fig.2. Illustration of McG. (a) McG framework. McG consists of cResNeXt and cMHSA modules in four stages. The number of blocks in each stage is denoted as N_1, N_2, N_3, and N_4, respectively. (b) cResNeXt structure. cResNeXt is a ResNeXt block with a conditional layer (CondLayer) before the ReLU layer. (c) cMHSA structure. cMHSA is a multi-head self-attention residual block with the conditional layer before the ReLU layer.

and the question feature to predict the answer.

3.1 Textual Network

The textual network composed of a word embedding layer (WE) and a bidirectional GRU (BiGRU)[50] is used to extract the question feature $\boldsymbol{q}$.

Given the question Q with a word sequence $w_1, w_2, \ldots, w_L$, the word embedding layer converts all the words into vectors.

$$\boldsymbol{E} = WE(Q),$$

where WE denotes the word embedding layer, $\boldsymbol{E} \in \mathbb{R}^{L \times d}$ denotes the word vectors, d denotes the dimension of word vectors and L denotes the length of the word sequence.

The word vectors are then sequentially fed into BiGRU. As a result, the last hidden state of BiGRU is taken as the question feature $\boldsymbol{q}$.

$$\boldsymbol{q} = BiGRU(\boldsymbol{E}).$$

3.2 Visual Network

3.2.1 *Visual Network Pipeline*

The visual network adopts a ResNeXt-like architecture[19], which contains a question-invariant CNN stem and a stack of conditional residual blocks based on the question feature.

CNN Stem. The CNN stem is adopted from ResNeXt-101 to extract the low-level visual features and reduce the feature map size. It takes the RGB image as input and outputs the feature map with a size of $(H/4) \times (W/4)$, where H and W represent the height and width of the image, respectively.

Stack of Conditional Blocks. The stack of conditional blocks is composed of two types of conditional blocks, i.e., cResNeXt and cMHSA. The integration of cResNeXt and cMHSA covers both CNN and the attention mechanism, which is critical in solving reasoning tasks. CNN is able to extract visual features by learning local patterns, while the attention mechanism is able to model the global dependencies among all image regions. The conditional layer is inserted before each rectified linear unit (ReLU) in the block to generate the instance-level activation function specified by the question feature. The stack of conditional blocks is divided into four stages, with each stage consisting of a number of conditional blocks as shown in Fig.2(a). The downsample strategy is applied at the beginning of the second and third stages to lower the feature map size for larger receptive fields and computation efficiency.

3.2.2 *cResNeXt*

cResNeXt is a ResNeXt block with the conditional layer placed before the Rectified Linear Unit (ReLU) layer. As shown in Fig.2(b), the input is sequentially fed to the bottleneck, group convolution, and short-cut connection.

In the bottleneck step, the feature is translated to a low-dimensional space by a convolution layer with a kernel size of 1, followed by a batch normalization layer, a conditional layer, and a ReLU layer.

$$\boldsymbol{F} = ReLU(C_\psi(BN(Conv_{1\times1}(\boldsymbol{F}_{\text{in}})), \boldsymbol{q})),$$

where $ReLU$ denotes the rectified linear unit, C_ψ is the conditional layer with the gating function ψ, BN denotes the batch normalization layer, $Conv_{1\times1}$ is a convolution layer with kernel size of 1, $\boldsymbol{F}_{\text{in}}$ is the input visual feature map, and $\boldsymbol{q}$ is the question feature for the conditional layer.

In the group convolution step, the feature map is divided into different channel groups and convolved by different convolution kernels with a kernel size of 3. Then the feature map is fed to a batch normalization layer, a conditional layer, and a ReLU layer.

$$\boldsymbol{F} = ReLU(C_\psi(BN(GrpConv_{3\times3}(\boldsymbol{F})), \boldsymbol{q})), \quad (1)$$

where $GrpConv$ is the group convolution layer, and the stride will be set to 2 when the downsample strategy is applied.

In the short-cut connection step, the results of different groups are converted by a convolution layer to the output space and summed up followed by batch normalization as residual. Then the feature map is fed to a conditional layer and a ReLU layer.

$$\boldsymbol{F} = ReLU(C_\psi(DS(\boldsymbol{F}_{\text{in}}) + BN(Conv_{1\times1}(\boldsymbol{F})), \boldsymbol{q})),$$

where DS represents either an identity layer or a downsample layer depending on whether the downsample strategy is applied.

3.2.3 *cMHSA*

cMHSA improves the MHSA block of the bottleneck transformer[40] by injecting the conditional layer before ReLU. It consists of a bottleneck, a multi-head self-attention, and a short-cut connection as shown in

Fig.2(c).

In the bottleneck, a feature-wise fully connected (FC) layer is used to reduce the dimension of the input visual feature map. Then the batch normalization layer, the conditional layer, and the ReLU layer are applied to generate the new feature map $\boldsymbol{F}$.

$$\boldsymbol{F} = ReLU(C_{\psi}(BN(FC(\boldsymbol{F}_{\text{in}})), \boldsymbol{q})),$$

where FC is the fully connected layer and $\boldsymbol{F}_{\text{in}}$ is the input visual feature map.

In the multi-head self-attention step, the visual features $\boldsymbol{F}$ are first converted into queries $\boldsymbol{Q}^{(h)}$, keys $\boldsymbol{K}^{(h)}$, and values $\boldsymbol{V}^{(h)}$ in the h-th head.

$$\begin{aligned}\boldsymbol{Q}^{(h)} &= \boldsymbol{F} \cdot \boldsymbol{W}_q^{(h)},\\ \boldsymbol{K}^{(h)} &= \boldsymbol{F} \cdot \boldsymbol{W}_k^{(h)},\\ \boldsymbol{V}^{(h)} &= \boldsymbol{F} \cdot \boldsymbol{W}_v^{(h)},\end{aligned}$$

where $\boldsymbol{W}_q^{(h)}, \boldsymbol{W}_k^{(h)}, \boldsymbol{W}_v^{(h)} \in \mathbb{R}^{d_v \times d_h}$ are linear transformation weights, d_v is the dimension of visual features, n_h is the number of heads, and d_h is the dimension of attention heads.

Then position-aware attention is applied.

$$\widehat{\boldsymbol{F}}_i^{(h)} = \text{softmax}\left(\frac{\boldsymbol{Q}_i^{(h)} \cdot \boldsymbol{K}^{(h)\text{T}} + \boldsymbol{Q}_i^{(h)} \cdot \boldsymbol{P}^{\text{T}}}{\sqrt{d_h}}\right) \cdot \boldsymbol{V}^{(h)},$$

where $\boldsymbol{Q}_i^{(h)} \in \mathbb{R}^{d_h}$ is the i-th query feature from the queries $\boldsymbol{Q}^{(h)}$. $\boldsymbol{P}$ is the position embedding of the keys, and $\hat{\boldsymbol{F}}_i^{(h)}$ is the i-th output visual feature in the h-th attention head. In the absolute position setting, $\boldsymbol{P}$ is determined by the absolute spatial position of keys. In the relative position[51] setting, $\boldsymbol{P}$ is determined by the relative spatial position of keys to the query vector. To enhance the spatial reasoning ability, we adopt the relative position embedding by default.

Then, the optional downsample layer, the normalization layer, the conditional layer, and the ReLU layer are applied.

$$\widetilde{\boldsymbol{F}}^{(h)} = ReLU(C_{\psi}(BN(DS(\widehat{\boldsymbol{F}}^{(h)})), \boldsymbol{q})).$$

In the short-cut connection step, the results of different heads are converted by a feature-wise fully connected layer to the output space and merged as a residual, followed by the conditional and normalization layer. An optional downsample layer is applied to the original input feature map $\boldsymbol{F}_{\text{in}}$ depending on whether the downsample strategy is applied.

$$\boldsymbol{F} = ReLU(C_{\psi}(DS(\boldsymbol{F}_{\text{in}}) + BN(\sum_{h=1}^{n_h} FC(\widetilde{\boldsymbol{F}}^{(h)})), \boldsymbol{q})).$$

3.2.4 Conditional Layers

In Fig.3, the conditional layer utilizes the question feature $\boldsymbol{q}$ to generate the dynamic parameters $\phi(\boldsymbol{q})$ by a fully connected network ϕ. Then gating function ψ is applied to the visual feature $\boldsymbol{V}$ with the generated parameters.

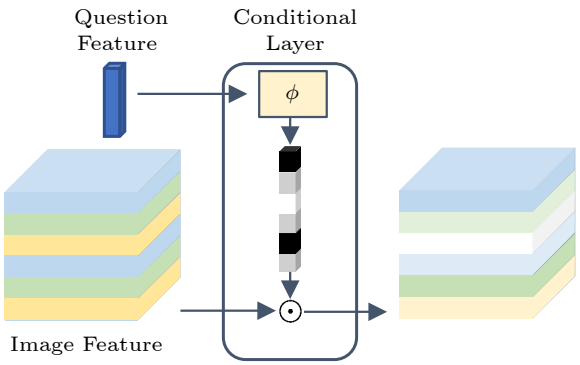


Fig.3. Channel-wise conditional gating. In the conditional layer, the dark cell denotes 1, the white cell denotes 0, and the gray cell denotes a value between 0 and 1.

$$C_{\psi}(\boldsymbol{V}, \boldsymbol{q}) = \psi(\boldsymbol{V}; \phi(\boldsymbol{q})).$$

For the gating function ψ, we define a series of implementations, including 2-way DyReLU[21], conditional affine, conditional sigmoid, and conditional 2-way softmax. The corresponding calculations are shown below.

$$\begin{aligned}C_{\text{DyReLU}}(\boldsymbol{V}, \boldsymbol{q}) &= \max_{i \leqslant 2} \boldsymbol{V} \odot (\boldsymbol{q} \cdot \boldsymbol{W}_{\alpha_i}) + \boldsymbol{q} \cdot \boldsymbol{W}_{\beta_i},\\ C_{\text{affine}}(\boldsymbol{V}, \boldsymbol{q}) &= \boldsymbol{V} \odot (\boldsymbol{q} \cdot \boldsymbol{W}_{\alpha}) + \boldsymbol{q} \cdot \boldsymbol{W}_{\beta},\\ C_{\text{sigmoid}}(\boldsymbol{V}, \boldsymbol{q}) &= \boldsymbol{V} \odot \text{sigmoid}(\boldsymbol{q} \cdot \boldsymbol{W}_{\alpha}),\\ C_{\text{softmax}}(\boldsymbol{V}, \boldsymbol{q}) &= \boldsymbol{V} \odot \text{softmax}(\boldsymbol{q} \cdot \boldsymbol{W}_{\alpha}),\end{aligned}$$

where $\odot$ denotes element-wise multiplication, and $\boldsymbol{W}_{\alpha}$ and $\boldsymbol{W}_{\beta}$ are linear projection weights for dynamic parameters generating.

As illustrated in Fig.4, the 2-way DyReLU activation, as well as the combination of conditional affine, sigmoid, and 2-way softmax with ReLU, are capable of producing various activation functions according to the question input. Obviously, the freedom of parameters decreases by the order of 2-way DyRLU, affine + ReLU, sigmoid + ReLU, and 2-way softmax + ReLU. Our model McG has the capability to utilize all types of conditional layers in both the CNN and Transformer modules. It is worth noting that the FiLM

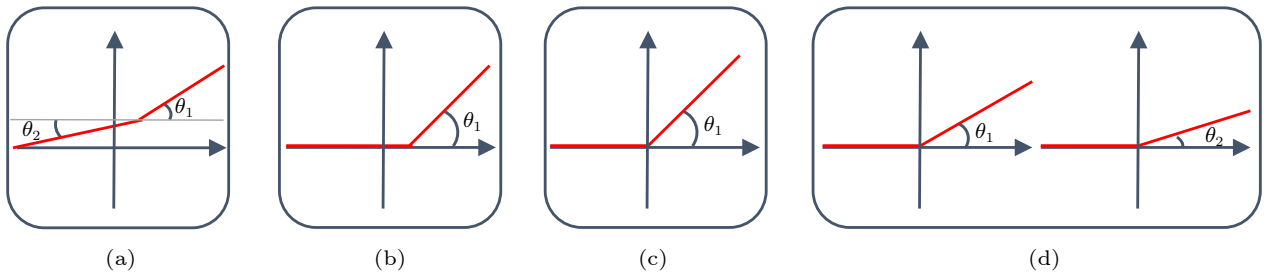


Fig.4. Efforts of conditional layers with ReLU. (a) 2-Way DyReLU, a maxout ReLU with two lines. (b) Affine+ReLU, a shiftable ReLU with the scale and bias parameters. (c) Sigmoid + ReLU, a ReLU with scale parameter $\tan\theta_1 \in [0, 1]$. (d) Softmax +ReLU, a ReLU with scale parameters $\tan\theta_1$, $\tan\theta_2$, where $\tan\theta_1 + \tan\theta_2 = 1$.

model can be regarded as a special case where the affine + ReLU combination is deployed solely in the CNN-based network.

3.3 Classifiers

We develop two classifiers for different datasets: a simple classifier for the CLEVR dataset, and a question-guided classifier for the VQA-Abstract dataset.

Simple Classifier. The simple classifier first attaches to the visual feature map with a 2-dimensional position map $\boldsymbol{M}$ to explicitly provide the absolute position information. Then, the feature map is fed to a 1×1 convolution layer and a max-pooling layer to generate the visual feature $\boldsymbol{v}$. Finally, the concatenation of the visual feature and question feature is fed to a 1-hidden-layer multilayer perceptron (MLP) to predict the answer score $\boldsymbol{s}$.

$$\boldsymbol{v} = pool_{\max}(Conv_{1\times1}([\boldsymbol{F}, \boldsymbol{M}])),$$

$$\boldsymbol{s} = MLP([\boldsymbol{v}, \boldsymbol{q}]).$$

Question-Guided Classifier. The question-guided classifier adopts a question-guided attention layer to replace the max-pooling layer in the simple classifier. The other layers are the same as the simple classifier.

4 Experiments

4.1 Datasets

CLEVR. CLEVR[20] is a synthesized dataset with scenes of objects and corresponding questions for visual reasoning. The objects in the scene have three shapes (cube, sphere, and cylinder), two sizes (small and large), two materials (shiny and matte), and eight colors. Additionally, there are four spatial relationships between two objects: left, right, behind, and in front. The dataset contains 70k images and 700k questions in the training split, 15k images and 150k questions in the validation split, and another 15k images and 150k questions in the test split. The synthesized questions concern the query and comparison of the attribute, number, and spatial relationship. The metric is based on the accuracy in "Count", "Exist", "Compare Number", "Query Attribute", "Compare Attribute" categories, and "Overall"[20].

VQA-Abstract. VQA-Abstract[1] is a dataset containing cartoon indoor and outdoor abstract scenes. The scene image is generated by 20 paper dolls, 100 objects, and 31 animals. The doll can adopt eight expressions and adjust its limbs to fit the abstract scene. The dataset has 20k images and 60k questions in the training split, 10k images and 30k questions in the validation split, and another 20k images and 60k questions in the test split. The questions are categorized as "Yes/No", "Number", and "Other" according to the answer types. The predicted answer is evaluated as an accuracy compared with 10 human-annotated answers.

4.2 Implementation

Model. In the textual network, we set the word vector dimension to 200, and the hidden dimension of the bidirectional GRU to 256. In the visual network, we implement the conditional block stack with three cResNeXt stages and one cMHSA stage. Each stage contains two conditional blocks with channel-wise sigmoid gating by default, respectively. For the classifier, we adopt a simple classifier for the CLEVR dataset and a question-guided classifier for the VQA-Abstract dataset. The group number and head number are set to 32 and 4 by default, respectively.

Training. For the CLEVR dataset, the model is trained with a batch size of 64 for 100 epochs. The learning rate is set to 0.000 3 with a weight decay of 1.0×10^{-5} by the Adam optimizer. The images are resized to 224×224 and data augmentation with pad four and random cropping is applied.

For the VQA-Abstract dataset, the model is

trained with a batch size of 64 for 30 epochs. The initial learning rate is set to 0.000 3 and a warm-up strategy is adopted. The learning rate decays at the 25th and 29th epochs with a decay rate of 0.2. The images are resized to 420×240, When evaluated on test split, the validation data is used as training examples.

4.3 Results

4.3.1 *Performance on CLEVR*

We compare our McG model with several strong baselines on the test split of the CLEVR dataset in Table 1. In all the tables of the paper, the best accuracy is highlighted in bold. We briefly introduce these methods as below.

- Human[11]: the human performance is measured on a 5.5k subset of CLEVR questions.
- LSTM[11]: the hidden state of LSTM is used as the question feature to predict the answer.
- CNN+LSTM[11]: CNN extracted image feature and LSTM extracted question feature are fed to the MLP for answer predicting.
- CNN+LSTM+SA[52]: the stack attention model with question feature as query and visual features as values.
- QGHC[18]: the group convolution kernel parameters are hybrid with question-generated and question-invariant.
- MAC[53]: a memory, attention, and composition model with read and write operation.
- FiLM[12]: a question-generated feature-wise linear modulation layer used in CNN block for visual reasoning.
- NS-CL[54]: a neuro-symbolic concept learner that learns visual concepts and words.
- N2NMN[15]: a neural module network generated by a layout policy generator with the question.
- PG+EE[11]: an execution engine executes the program generated by the program generator.
- TbD[16]: a neural module network with explicit visual reasoning primitives.
- CNN+LSTM+RN[52]: a simple relation network based on all feature pairs in the CNN feature map.
- BRN[13]: a discrete routing network on the branch modules depending on the question feature.
- FRN[13]: a discrete routing network on the convolution filters depending on the question feature.
- ViLT[10]: a vision and language transformer with image patches as input and without any convolution network.

In Table 1, the first row shows the performance of Human on the subset of CLEVR in the previous work[11]. It is only 92.6% because CLEVR contains many highly compositional questions that require multi-step reasoning. Rows 2–8 show the performance of models trained on the extracted visual features (e.g., ResNet101[27]) without program annotation. The compositional attention network MAC, the

Table 1. Accuracy of McG and Other State-of-the-Art Models on CLEVR Test Split

Model	Overall	Count	Exist	Compare Number	Query Attribute	Compare Attribute
Human[11]	92.6	86.7	96.6	86.5	95.0	96.0
LSTM[11]	46.8	41.7	61.1	69.8	36.8	51.8
CNN+LSTM[11]	52.3	43.7	65.2	67.1	49.3	53.0
CNN+LSTM+SA[52]	76.6	64.4	82.7	77.4	82.6	75.4
QGHC[18]	86.3	78.1	91.7	80.7	89.4	86.8
MAC[53]	98.9	97.2	99.5	99.4	99.3	99.5
FiLM[12]	97.7	94.3	99.1	96.8	99.1	99.1
NS-CL[54]	98.9	98.2	98.9	99.0	99.3	99.1
N2NMN[15]+	83.7	68.5	85.7	84.9	90.0	88.7
PG+EE[11]+	96.9	92.7	97.1	98.7	98.1	98.9
TbD[16]+	99.1	97.6	99.2	99.4	99.5	99.6
CNN+LSTM+RN[52]*	95.5	90.1	97.8	93.6	97.9	97.1
FiLM[12]*	97.6	94.3	99.3	93.4	99.3	99.3
BRN[13]*	98.7	97.4	99.6	95.1	99.7	99.7
FRN[13]*	98.9	97.7	**99.9**	94.5	99.8	**99.9**
ViLT*[10]	98.4	96.2	99.6	94.6	99.6	99.6
McG*(ours)	**99.7**	**99.3**	99.8	**99.5**	**99.9**	99.8

Note: "*" denotes training by raw pixels of images in an end-to-end way. "+" denotes using the program annotation of the question. "Count", "Exist", "Compare Number", "Query Attribute", "Compare Attribute" columns denote the accuracy on different question categories, and "Overall" represents the average accuracy across all categories.

neural symbolic network NS-CL, and the conditional network FiLM show very strong performances of around 98%. The rows with "+" show the performances of neural module models that are trained with annotated programs of questions. With the prior domain knowledge in the program annotations, TbD can achieve a high score of 99.1%. The rows with "*" represent the models that are trained on the raw pixels of images. These models can utilize the question feature in the low-level visual extracting stages. The CNN-based FiLM, BRN, and FRN have achieved a strong performance of about 98%. In addition, ViLT as a convolution-free vision and language transformer achieves an accuracy of 98.4%. It is obvious that McG achieves a new state-of-the-art accuracy of 99.7% overall on the CLEVR test split. In particular, McG achieves the first place in three categories ("Count", "Compare Number", and "Query Attribute") and the second place in two categories ("Exist" and "Compare Attribute"). This indicates the strong visual reasoning abilities of McG. Compared with the second-best model TbD, McG reduces the error rate from 0.9% to 0.3%, which implies the advantage of McG. To emphasize, TbD utilizes the ground program annotation as extra supervised information, while does not use any auxiliary information. This further implies that McG can learn visual reasoning ability solely by image, question, and answer triplets. For the dynamic routing networks BRN and FRN, McG outperforms them by 1.0% and 0.8%, respectively. Compared with the pioneer conditional network FiLM, which adopts a conditional channel-wise affine layer within the residual CNN block for the first time, McG outperforms it with a margin of 2.1%. This implies that a mixture of conditional CNN and MHSA blocks is better than a network made of pure conditional CNN blocks.

4.3.2 *Performance on VQA-Abstract*

In Table 2, we compare the McG model with the baselines on the test split of the VQA-Abstract dataset. A simple introduction to these models is listed below.

- LSTM blind [1]: an LSTM model predicting the answer only based on the question.
- LSTM+CNN [1]: a model predicting the answer based on the LSTM extracted question feature and CNN extracted visual feature.
- MRL [55]: a multi-modal residual learning network on both visual and textual features.
- DualNet [56]: a model fusing visual and textual features in a common space.
- Graph VQA [57]: a model with graph representation for both the image and the question, and GRU-based visual reasoning.

Table 2. Accuracy on VQA-Abstract Test Split

Model	Overall	Yes/No	Other	Number
LSTM blind[1]	57.19	76.88	38.79	49.55
LSTM+CNN[1]	65.02	77.45	56.41	52.54
MRL[55]	62.56	79.10	48.90	51.60
DualNet[56]+	68.87	80.00	57.90	61.10
Graph VQA[57]+	70.42	81.26	56.28	**76.47**
McG(ours)	**70.80**	**82.44**	**58.54**	69.82

Note: "+" denotes using the annotated scene composition annotation as an auxiliary input. "Yes/No", "Number", and "Other" denote the accuracy on different question categories, and "Overall" represents the average accuracy across all categories.

McG achieves an accuracy of 70.80% on the VQA-Abstract test split, which overbeats all the baselines. Concretely, McG gains the best scores of 82.44% and 58.54% on the "Yes/No" and "Other" category, respectively. However, McG is 6.65% behind the Graph VQA model on the "Number" questions. This is reasonable since the Graph VQA model utilizes the ground truth scene layout information, including the occurrence of object, expression, and pose of the doll as the input of the model during both training and testing. The scene layout information is very valuable during visual reasoning, especially for the counting questions. McG outperforms the "LSTM+CNN" model with a gap of 5.78% on the overall accuracy. This indicates the effect of the conditional block compared with the plain CNN and LSTM modules.

4.4 Ablation Study

We conduct the ablation study on the CLEVR and VQA-Abstract validation split into three levels. At the architecture level, we propose the McG variants with different conditional blocks (cResNeXt and cMHSA) combinations in the stack. At the module level, we implement the conditional block with different conditional layers and designs. In the hyper-parameter level, we train the McG variants with different group numbers, head numbers, kernel sizes, and block numbers. The accuracy is measured on the validation split of the CLEVR and VQA-Abstract datasets in the ablation study part.

4.4.1 Architecture-Level Study

The conditional block stack in McG contains four stages, and each stage contains either the CNN blocks or MHSA blocks. Furthermore, each block can be designed as conditional or static. At the architecture level, we study the combination of different blocks as shown in Table 3.

Table 3. Accuracy of Architecture with Different Blocks

Model	Block Type	Conditional	CLVER	VQA-Abstract
McG	cccc	yyyy	97.50	64.84
McG	ccct	yyyy	**99.73**	**67.15**
McG	cctt	yyyy	99.70	66.47
McG†	cttt	yyyy	99.31	63.09
McG†	tttt	yyyy	97.66	61.67
McG	ccct	yyyn	99.67	66.37
McG	ccct	nnny	98.86	65.96
McG	ccct	nnyn	99.56	66.60
McG	ccct	nynn	99.66	65.36
McG	ccct	ynnn	98.91	65.21
McG	ccct	nnnn	66.59	64.15

Note: "†" denotes McG downsample at the first and the second stages. In the "Block Type" column, "c" denotes cResNeXt, and "t" denotes cMHSA. In the "Conditional" column, "y/n" indicates whether the corresponding block uses the conditional layer. The accuracy is measured on the validation split of the CLEVR and VQA-Abstract datasets.

Study on the Combination of cResNeXt and cMHSA. In rows 1–5 of Table 3, we implement several McG variants by using different combinations of cResNeXt and cMHSA in the 4-stage stack. In the "Block Type" column, we use four letters to represent these variants, and each letter denotes the block type in the corresponding stage: "c" means cResNeXt, "t" means "cMHSA". "cccc" denotes a pure cResNeXt stack in the visual network. "ccct", "cctt", and "cttt" denote the hybrid combination of cResNeXt and cMHSA in the 4-stage stack. "tttt" denotes a pure cMHSA stack in the visual network. In our implementation, the models "cccc", "ccct", and "cctt" downsample at the second and third stages as a default setting. Meanwhile, the models "cttt" and "tttt" downsample at the first and second stages to make it feasible for training. According to the results, the "ccct" model outperforms the others, achieving an impressive accuracy rate of 99.73% and 67.15% on the validation split of the CLEVR and VQA-Abstract datasets, respectively. These findings indicate that the optimal approach for the VQA task involves using cResNeXt to model local patterns in the low-level stages, and cMHSA to model global dependencies in the high-level stages.

Study on Combination of Conditional and Static Blocks. The strategy used in each stage of McG can be either conditional or static (without using conditional layers). In column "Conditional" of Table 3, we adopt a 4-letter code to denote whether each of the four stages uses conditional layers: "y" for used, "n" for not used. To investigate the impact of the conditional and static strategy, we develop several variants of McG with "ccct" architecture in rows 6–11 of Table 3. "yyyn" denotes that cMHSA in the fourth stage do not use the conditional layer. "nnny" denotes only cMHSA in the fourth stage using the conditional layer. "nnyn", "nynn", and "ynnn" represent only cResNeXt in the third, second, and first stages using the conditional layer, respectively. "nnnn" means all the blocks do not use the conditional layer. By comparing "yyyn" and "nnny" with the default "yyyy" setting, we can see a drop of 0.06% and 0.87% on CLEVR, which demonstrates that both the conditional layer in cResNeXt and cMHSA are important for McG. For the comparison of "nnyn", "nynn", and "ynnn", we can see the conditional layer in the second stage is the most important if only one stage uses the conditional layer. The accuracy of "nynn" achieves 99.66% on CLEVR, which is only 0.07% behind the default McG. When comparing the "nnnn" setting with the default "yyyy" setting, we find the performance decreased by 33.14% and 3% on CLEVR and VQA-Abstract datasets, respectively. The performance decline on VQA-Abstract is less significant due to the higher model capacity of the question-guided classifier, which helps to alleviate the impact of losing conditional layers. The performance drop has further demonstrated the importance of conditional layers in McG.

To be noticed the conditional layers are very efficient in the McG variants as shown in Fig.5. The number of parameters in the McG model increases by 2%–10% when only one stage incorporates a conditional layer, and by 22% when all stages adopt conditional layers. Meanwhile, the metric floating point operations per second (FLOPS) increases by a mere 0.1% when all stages incorporate conditional layers.

4.4.2 Module-Level Study

At the module level, we investigate the effects of conditional layer type and position of the conditional layer, as well as the effects of position embedding type in cMHSA as shown in Table 4.

Conditional Layer Types. We implement the conditional layers with different gating functions, including DyReLU, affine, sigmoid, and softmax. As shown

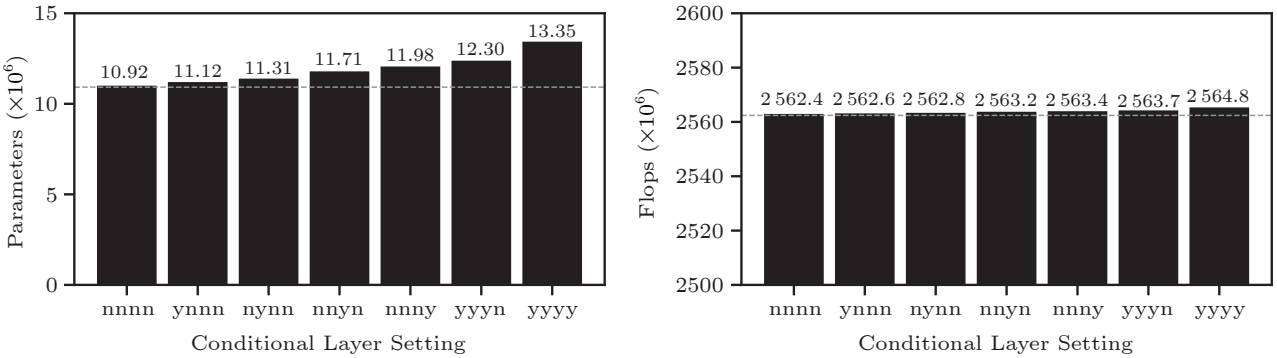


Fig.5. Model parameters and FLOPS for the McG variants on the VQA-Abstract dataset, where "y" and "n" denote whether the corresponding stage adopts the conditional layer. (a) Parameters for McG variants. (b) FLOPS for McG variants.

Table 4. Accuracy of Different Conditional Layer Types

Conditional Layer Type	CLEVR	VQA-Abstract
Affine	99.67	65.49
DyReLU	99.65	66.53
Softmax	99.64	65.41
Sigmoid	**99.73**	**67.15**

in Fig.4, these conditional layers can lead to activation functions with different freedom degrees. The freedom degree decreases in the following order: 2-way DyReLU, affine+ReLU, sigmoid+ReLU, to 2-way softmax+ReLU. Table 4 shows that the sigmoid conditional layer, which has a moderate freedom degree, performs better than all the other conditional layer types on both the CLEVR and VQA-Abstract datasets. The variants of McG with the affine, DyReLU, and softmax gating function still achieve remarkable scores on both the CLEVR and VQA-abstract, which implies that the conditional block design in McG is very general and practical.

Conditional Layer Position. As shown in Figs.2(b) and 2(c), there are three conditional layers in the standard conditional block, which are represented as the position 1–3 according to the execution order. To investigate which position is the most important for the conditional block, we develop three variants, each containing one conditional layer while leaving the others as plain ReLU. The first three rows in Table 5 show that the second and third conditional layers are the most significant for the CLEVR and VQA-Abstract dataset, respectively. The variants at three different positions have all dropped slightly, especially in the CLEVR dataset. On the one hand, this implies the robustness of McG. On the other hand, this also shows that the VQA-Abstract dataset needs a large model capacity.

Table 5. Accuracy of Different Conditional Block Variants

Model Setting	CLEVR	VQA-Abstract
Conditional layer position 1	99.70	66.29
Conditional layer position 2	99.72	66.57
Conditional layer position 3	99.68	66.58
cMHSA absolute position emb	99.48	67.07
cMHSA no position emb	99.24	67.04
Default	**99.73**	**67.15**

Position Embedding in cMHSA. Position embedding is critical in capturing the spatial relation between two visual objects. To enhance the spatial reasoning ability of cMHSA, we adopt the relative position embedding in the attention procedure by default. We implement two variants: "cMHSA absolute position emb" means using absolute position embedding in attention. "cMHSA no position emb" denotes not using any position embedding in attention. Row 4 of Table 5 shows that the performance of cMHSA with absolute position embedding decreases on both the CLEVR and the VQA-Abstract datasets. This suggests that the relative spatial reasoning ability is more critical than the absolute spatial reasoning ability in VQA. In addition, the "cMHSA no position emb" model drops 0.49% and 0.11% on CLEVR and VQA-Abstract, respectively. This indicates spatial reasoning ability is more required in CLEVR since it has a lot of questions on spatial relationships.

4.4.3 *Hyperparameter-Level Study*

Branch Number. Both cResNeXt and cMHSA employ a branch architecture, with groups in cResNeXt and heads in cMHSA, respectively. The model capacity increases as the number of groups or the number of heads decreases. According to the results in Table 6, the best model for CLEVR has 32 groups and four heads, while the best model for VQA-Abstract has 16 groups and four heads. This further suggests that the

Table 6. Accuracy of Different Branch Numbers in McG

Number of Groups	Number of Heads	CLEVR	VQA-Abstract
1	1	98.63	66.35
16	4	99.69	**67.41**
16	8	99.72	66.80
32	4	**99.73**	67.15
32	8	99.70	67.09
64	4	99.61	66.97
64	8	99.56	66.55

model capacity needed for the VQA-Abstract dataset is larger than that for the CLEVR dataset.

Kernel Size. The cResNeXt module defaults to a kernel size of three for the group convolution operation, as illustrated in (1). We investigate the influence of kernel size by developing two variants. Row 2 in Table 7 represents setting the kernel size to five for all the groups. Row 3 represents setting the kernel size to three in the first half group and five in the second half group. Table 7 shows that the model with the kernel size of three achieves the best performance on both the CLEVR and VQA-Abstract datasets with a slight gap. The findings indicate that employing a kernel size of three generates a small receptive field, which enhances the model's ability to capture finer-grained visual features. Consequently, this results in superior performance on the VQA task.

Block Number. We adopt 1–3 blocks per stage to study the influence of block number on the model performance. For the CLEVR dataset, the model with one block per stage achieves an overall accuracy of 99.15% on CLEVR as shown in Table 8, which can beat the previous state-of-the-art models. It can be noticed that the model performance drops slightly when the block number of a stage increases from two to three, which indicates that the model with two blocks a stage reaches an optimal capacity for CLEVR. Similarly, the two blocks per stage strategy is also the best choice for the VQA-Abstract dataset.

Table 7. Accuracy of Different Kernel Sizes in McG

Kernel Size	CLEVR	VQA-Abstract
3	**99.73**	**67.15**
5	99.64	66.83
3, 5	99.65	66.88

Table 8. Accuracy of Different Block Numbers in Stages

Stage 1	Stage 2	Stage 3	Stage 4	CLEVR	VQA-Abstract
1	1	1	1	99.15	65.96
2	2	2	2	**99.73**	**67.15**
3	3	3	3	99.65	65.86

4.5 Visualization

4.5.1 *Distribution of Gate Value*

To study the distribution of gate value for McG on the CLEVR validation dataset, we draw the mean, standard deviation (std) plot, and histogram in Fig.6. The gate value mean decreases while the standard deviation increases as the block index number grows, which is consistent with the discovery of previous studies FiLM[12] and QGMRN[13]. Most of the gate values are near 1 in the first block, which indicates that most channels are required in the low-level neurons. The gating value mean of the last block is close to 0.65 and has a wide variance. It implies that the activation of high-level neurons is sparse and dynamic, and whether it will be activated is dependent on the input question.

4.5.2 *t-SNE for Gate Value*

We propose a pure cResNeXt model ("cccc") and compare it with the default McG ("ccct") on the CLEVR dataset by t-SNE to analyze the advantage of cMHSA in Fig.7. The t-SNE plot of the gate value in the last block has clearly shown that the gate value vector is most related to the high-level question category, such as query, compare, and count. For example, all the query-family questions, including "QueryColor", "QueryMaterial", "QueryShape", and "QuerySize", are clustered together for both the pure cResNeXts model and McG. Moreover, the t-SNE of McG has a more fine-grained cluster result. For example, the "Count" and "Exist" types are clustered closely but still separable. The excellent accuracy on the "Count" and "Exist" categories in Table 1 further demonstrate this.

4.5.3 *Heatmap Examples*

We adopt Grad-Cam[58] to visualize the most important area in the image for the predicted answer.

Heatmap Examples on CLEVR. The heatmaps on CLEVR in Fig.8 show that McG has the capability to focus on the key object by the question no matter whether the question is simple or complex. The first simple question involves size and color filtering and shape description. The second complex question involves shape and color filtering, equal-shape relation-

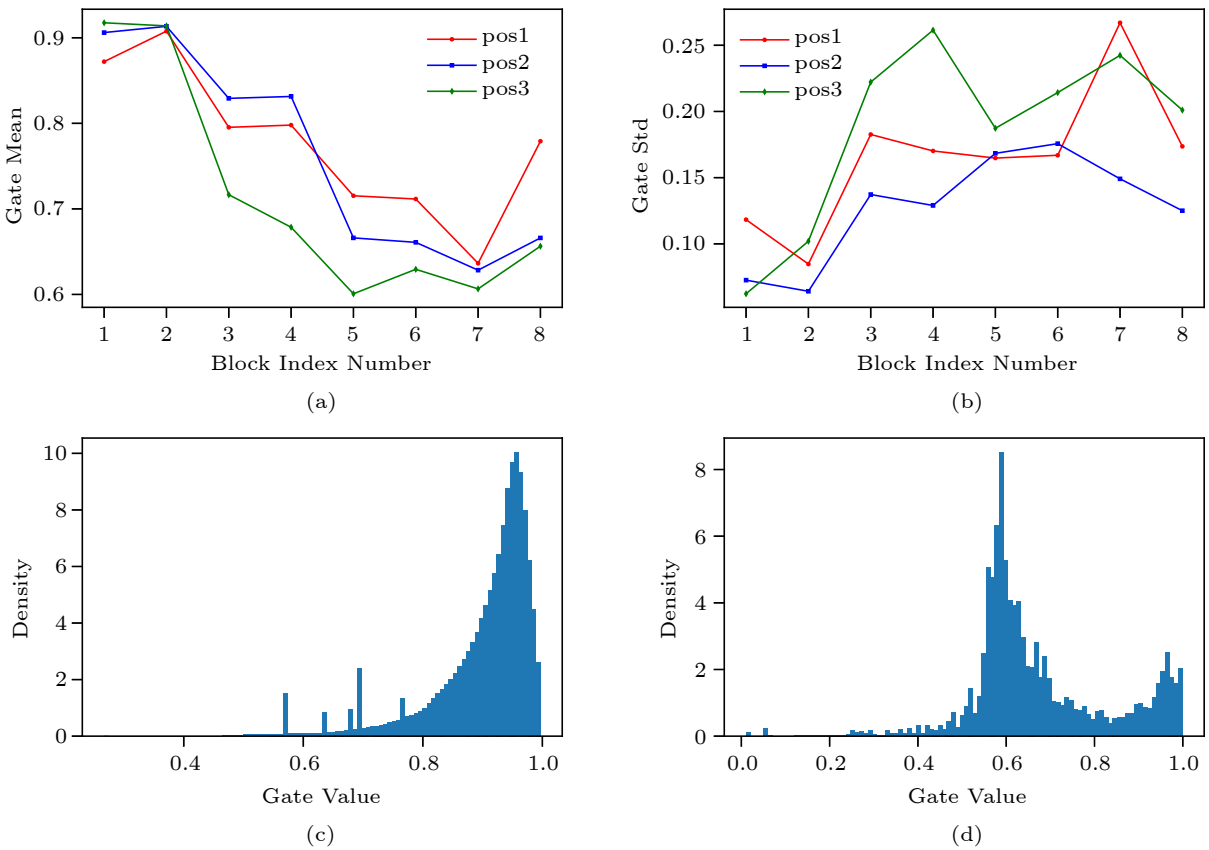


Fig.6. Gate value distribution. (a) The gate mean of each position in all blocks. (b) The gate standard deviation (std) of each position in all blocks. (c) The gate density distribution in the first block. (d) The gate density distribution in the last block.

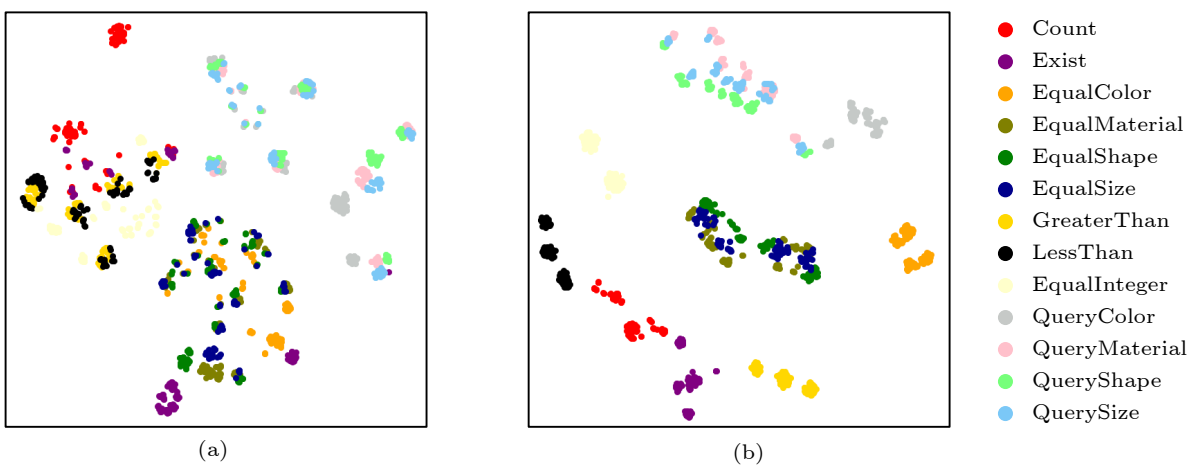
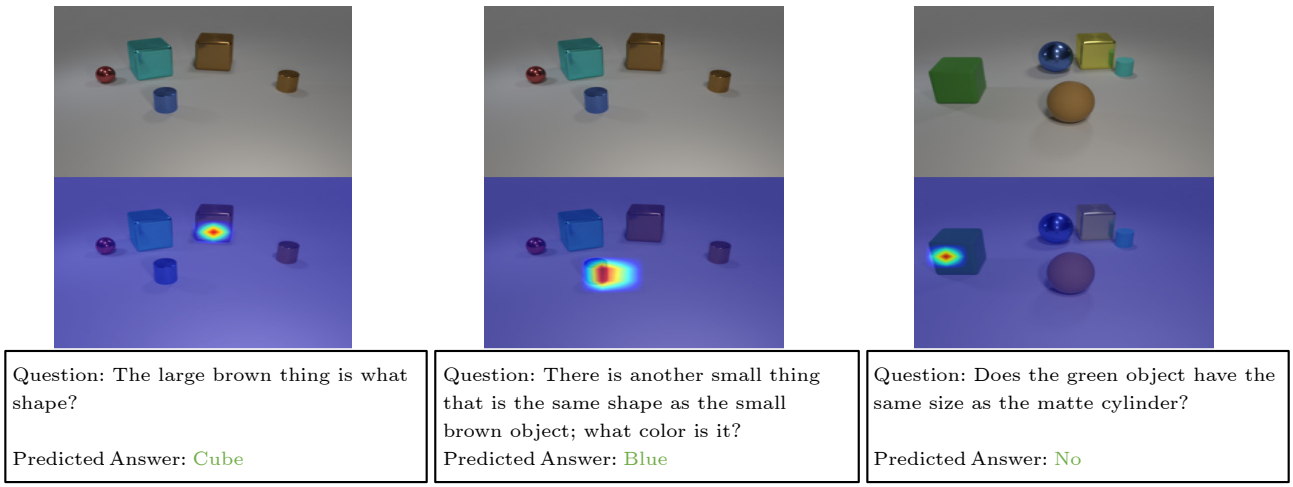


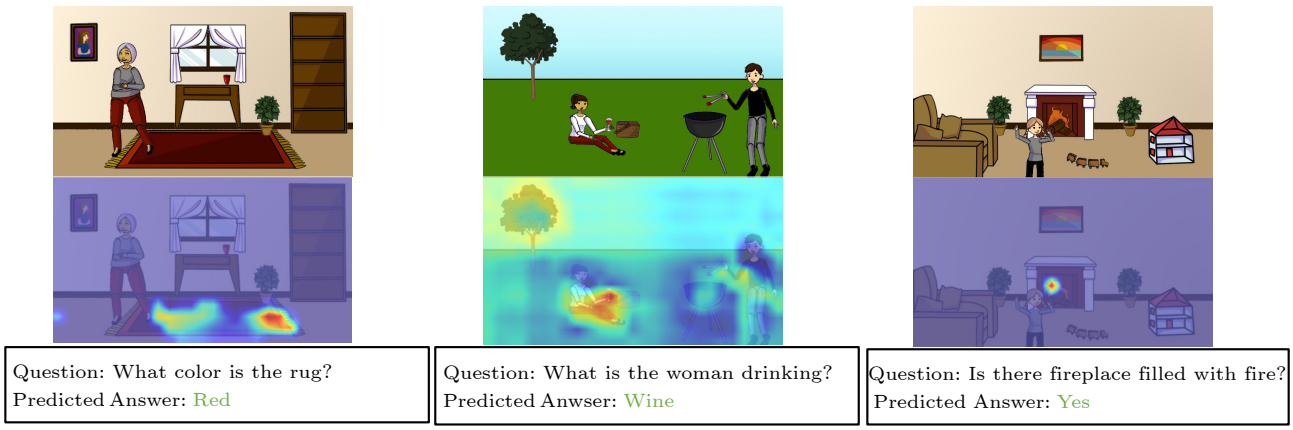
Fig.7. t-SNE visualization. (a) Gate value of the last cResNeXt in a pure cResNXt model. (b) Gate value of the last cMHSA in the default McG model.

ship reasoning, and color description. The third question is an attribute comparison type, and we can find the heatmap highlights the subject in the question.

Heatmap Examples on VQA-Abstract. The heatmaps on VQA-Abstract in Fig.8 show that our McG model is able to query an attribute, identify an object, and judge a fact. In attribute querying, McG focuses on the noun word mentioned by the question, such as "rug" in the first column. In object identification, it links the image region to the related action word, such as the wine goblet is highlighted in the second column because "drinking" is mentioned by the question in the second column. In fact judging, the McG model is also able to locate the image region that expresses the fact, such as "fireplace filled with fire" in the third column.



(a)



(b)

Fig.8. Heatmap examples on the CLEVER and VQA-Abstract datasets, with the original images, Grad-Cam visualizations, and corresponding questions arranged from top to bottom. (a) Heatmap examples on CLEVER. (b) Heatmap examples on VQA-Abstract.

5 Conclusions

Our proposed mixture of conditional gating blocks (McG) represents a significant advancement in addressing the visual question answering (VQA) task. The integration of conditional multi-head self-attention block (cMHSA) and conditional ResNeXt block (cResNeXt) within McG allows us to harness the strengths of dynamic parameters, Transformer, and convolutional neural network (CNN) simultaneously. Dynamic parameters improve the adaptability, model capacity, and interpretability. Transformer captures the global feature dependency. CNN backbone models local patterns and translation invariance. Specifically, McG is trained on the raw images with the channel gate layer generated by the question in an end-to-end manner. Thus McG is able to influence the visual network in both the low-level and high-level stages through the question features. Our findings suggest that the mixture of conditional gating blocks is a promising approach for advancing the field of VQA and holds potential for broader applications in the vision and language tasks.

Acknowledgements We would like to thank Yan-Ze Wu (a postgraduate student at the School of Computer Science, Fudan University) for the insightful discussions.

Conflict of Interest The authors declare that they have no conflict of interest.

References

[1] Antol S, Agrawal A, Lu J, Mitchell M, Batra D, Zitnick C L, Parikh D. VQA: Visual question answering. In *Proc. the IEEE International Conference on Computer Vision*, Dec. 2015, pp.2425–2433. DOI: 10.1109/iccv.2015.279.

[2] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł, Polosukhin I. Attention is all you

need. In *Proc. the 31st In. Conf. Neural Information Processing Systems*, Dec. 2017, pp.6000–6010.

[3] Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y Q, Li W, Liu P J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 2020, 21(1): Article No. 140.

[4] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X H, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. the 9th International Conference on Learning Representations*, May 2021.

[5] Nicolas C, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers. In *Proc. the 16th European Conference on Computer Vision*, Aug. 2020, pp.213–229. DOI: 10.1007/978-3-030-58452-8_13.

[6] Zheng S X, Lu J C, Zhao H S, Zhu X T, Luo Z K, Wang Y B, Fu Y W, Feng J F, Xiang T, Torr P H S, Zhang L. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proc. the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2021, pp.6877–6886. DOI: 10.1109/CVPR46437.2021.00681.

[7] Lu J S, Batra D, Parikh D, Lee S. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Proc. the 33rd International Conference on Neural Information Processing Systems*, Dec. 2019, Article No. 2.

[8] Tan H, Bansal M. LXMERT: Learning cross-modality encoder representations from transformers. In *Proc. the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, Nov. 2019, pp.5100–5111. DOI: 10.18653/v1/d19-1514.

[9] Huang Z C, Zeng Z Y, Liu B, Fu D M, Fu J L. Pixel-BERT: Aligning image pixels with text by deep multi-modal transformers. arXiv: 2004.00849, 2020. https://arxiv.org/abs/2004.00849, Jun. 2024.

[10] Kim W, Son B, Kim I. ViLT: Vision-and-language transformer without convolution or region supervision. In *Proc. the 38th International Conference on Machine Learning*, Jul. 2021, pp.5583–5594.

[11] Johnson J, Hariharan B, Van Der Maaten L, Hoffman J, Fei-Fei L, Zitnick C L, Girshick R. Inferring and executing programs for visual reasoning. In *Proc. the 2017 IEEE International Conference on Computer Vision*, Oct. 2017, pp.3008–3017. DOI: 10.1109/iccv.2017.325.

[12] Perez E, Strub F, De Vries H, Dumoulin V, Courville A. FiLM: Visual reasoning with a general conditioning layer. In *Proc. the 32nd AAAI Conference on Artificial Intelligence*, Apr. 2018. DOI: 10.1609/aaai.v32i1.11671.

[13] Wu Y Z, Sun Q, Ma J Q, Li B, Fu Y W, Peng Y, Xue X Y. Question guided modular routing networks for visual question answering. arXiv: 1904.08324, 2019. https://arxiv.org/abs/1904.08324, Jun. 2024.

[14] Zhong H S, Chen J Y, Shen C, Zhang H W, Huang J Q, Hua X S. Self-adaptive neural module transformer for visual question answering. *IEEE Trans. Multimedia*, 2021, 23: 1264–1273. DOI: 10.1109/tmm.2020.2995278.

[15] Hu R H, Andreas J, Rohrbach M, Darrell T, Saenko K. Learning to reason: End-to-end module networks for visual question answering. In *Proc. the 2017 IEEE International Conference on Computer Vision*, Oct. 2017, pp.804–813. DOI: 10.1109/iccv.2017.93.

[16] Mascharka D, Tran P, Soklaski R, Majumdar A. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *Proc. the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp.4942–4950. DOI: 10.1109/cvpr.2018.00519.

[17] Noh H, Seo P H, Han B. Image question answering using convolutional neural network with dynamic parameter prediction. In *Proc. the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp.30–38. DOI: 10.1109/cvpr.2016.11.

[18] Gao P, Li H S, Li S, Lu P, Li Y K, Hoi S C H, Wang X G. Question-guided hybrid convolution for visual question answering. In *Proc. the 15th European Conference on Computer Vision*, Sept. 2018, pp.485–501. DOI: 10.1007/978-3-030-01246-5_29.

[19] Xie S N, Girshick R, Dollár P, Tu Z W, He K M. Aggregated residual transformations for deep neural networks. In *Proc. the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017, pp.5987–5995. DOI: 10.1109/cvpr.2017.634.

[20] Johnson J, Hariharan B, Van Der maaten L, Fei-Fei L, Zitnick C L, Girshick R. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proc. the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017, pp.1988–1997. DOI: 10.1109/cvpr.2017.215.

[21] Chen Y P, Dai X Y, Liu M C, Chen D D, Yuan L, Liu Z C. Dynamic ReLU. In *Proc. the 16th European Conference on Computer Vision*, Aug. 2020, pp.351–367. DOI: 10.1007/978-3-030-58529-7_21.

[22] Goyal Y, Khot T, Summers-Stay D, Batra D, Parikh D. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Proc. the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017, pp.6325–6334. DOI: 10.1109/cvpr.2017.670.

[23] Hudson D A, Manning C D. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proc. the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2019, pp.6693–6702. DOI: 10.1109/cvpr.2019.00686.

[24] Anderson P, He X D, Buehler C, Teney D, Johnson M, Gould S, Zhang L. Bottom-up and top-down attention for image captioning and visual question answering. In *Proc. the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp.6077–6086. DOI: 10.1109/cvpr.2018.00636.

[25] Liu F, Liu J, Fang Z W, Hong R C, Lu H Q. Visual question answering with dense inter- and intra-modality interactions. *IEEE Trans. Multimedia*, 2021, 23:3518–3529. DOI: 10.1109/tmm.2020.3026892.

[26] Yu J, Zhang W F, Lu Y H, Qin Z C, Hu Y, Tan J L, Wu Q. Reasoning on the relation: Enhancing visual representation for visual question answering and cross-modal retrieval. *IEEE Trans. Multimedia*, 2020, 22(12): 3196–3209. DOI: 10.1109/tmm.2020.2972830.

[27] He K M, Zhang X Y, Ren S Q, Sun J. Deep residual learning for image recognition. In *Proc. the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp.770–778. DOI: 10.1109/cvpr.2016.90.

[28] Fukui A, Park D H, Yang D, Rohrbach A, Darrell T, Rohrbach M. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proc. the 2016 Conference on Empirical Methods in Natural Language Processing*, Nov. 2016, pp.457–468. DOI: 10.18653/v1/d16-1044.

[29] Ben-Younes H, Cadene R, Cord M, Thome N. MUTAN: Multimodal tucker fusion for visual question answering. In *Proc. the 2017 IEEE International Conference on Computer Vision*, Oct. 2017, pp.2631–2639. DOI: 10.1109/iccv.2017.285.

[30] Kazemi V, Elqursh A. Show, ask, attend, and answer: A strong baseline for visual question answering. arXiv: 1704.03162, 2017. https://arxiv.org/abs/1704.03162, Jun. 2024.

[31] Niu Y L, Tang K H, Zhang H W, Lu Z W, Hua X S, Wen J R. Counterfactual VQA: A cause-effect look at language bias. In *Proc. the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2021, pp.12695–12705. DOI: 10.1109/cvpr46437.2021.01251.

[32] Pan Y H, Li Z C, Zhang L Y, Tang J H. Causal inference with knowledge distilling and curriculum learning for unbiased VQA. *ACM Trans. Multimedia Computing, Communications, and Applications*, 2022, 18(3): 67. DOI: 10.1145/3487042.

[33] Chen L, Yan X, Xiao J, Zhang H W, Pu S L, Zhuang Y T. Counterfactual samples synthesizing for robust visual question answering. In *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2020, pp.10797–10806. DOI: 10.1109/cvpr42600.2020.01081.

[34] Yu Z, Yu J, Cui Y H, Tao D C, Tian Q. Deep modular co-attention networks for visual question answering. In *Proc. the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2019, pp.6274–6283. DOI: 10.1109/cvpr.2019.00644.

[35] Yu J, Li J, Yu Z, Huang Q M. Multimodal transformer with multi-view visual representation for image captioning. *IEEE Trans. Circuits and Systems for Video Technology*, 2019, 30(12): 4467–4480. DOI: 10.1109/TCSVT.2019.2947482.

[36] Wang J H, Jin L, Li Z C, Tang J H. Crossmodal knowledge distillation hashing. *Scientia Sinica Technologica*, 2022, 52(5): 713–726. DOI: 10.1360/sst-2021-0214.

[37] Andreas J, Rohrbach M, Darrell T, Klein D. Learning to compose neural networks for question answering. In *Proc. the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Jun. 2016, pp.1545–1554. DOI: 10.18653/v1/n16-1181.

[38] Devlin J, Chang M W, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Jun. 2019, pp.4171–4186. DOI: 10.18653/v1/N19-1423.

[39] Radford A, Narasimhan K, Salimans T, Sutskever I, Improving language understanding by generative pre-training, https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf, May 2024.

[40] Srinivas A, Lin T Y, Parmar N, Shlens J, Abbeel P, Vaswani A. Bottleneck transformers for visual recognition. In *Proc. the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2021, pp.16514–16524. DOI: 10.1109/cvpr46437.2021.01625.

[41] Huang G, Chen D L, Li T H, Wu F, Van Der Maaten L, Weinberger K Q. Multi-scale dense networks for resource efficient image classification. In *Proc. the 6th International Conference on Learning Representations*, May 2018.

[42] Teerapittayanon S, McDanel B, Kung T H. BranchyNet: Fast inference via early exiting from deep neural networks. In *Proc. the 23rd International Conference on Pattern Recognition*, Dec. 2016, pp.2464–2469. DOI: 10.1109/icpr.2016.7900006.

[43] Wang X, Yu F, Dou Z Y, Darrell T, Gonzalez J E. SkipNet: Learning dynamic routing in convolutional networks. In *Proc. the 15th European Conference on Computer Vision*, Sept. 2018, pp.420–436. DOI: 10.1007/978-3-030-01261-8_25.

[44] Wu Z X, Nagarajan T, Kumar A, Rennie S, Davis L S, Grauman K, Feris R. BlockDrop: Dynamic inference paths in residual networks. In *Proc. the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp.8817–8826. DOI: 10.1109/cvpr.2018.00919.

[45] Shazeer N, Mirhoseini A, Maziarz K, Davis A, Le Q V, Hinton G E, Dean J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *Proc. the 5th International Conference on Learning Representations*, Apr. 2017.

[46] Cai S F, Shu Y, Wang W. Dynamic routing networks. In *Proc. the 2021 IEEE Winter Conference on Applications of Computer Vision*, Jan. 2021, pp.3587–3596. DOI: 10.1109/wacv48630.2021.00363.

[47] Fedus W, Zoph B, Shazeer N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 2022, 23(1): 120.

[48] Ha D, Dai A M, Le Q V. HyperNetworks. In *Proc. the 5th International Conference on Learning Representations*, Apr. 2017.

[49] Jiang A W, Liu B, Wang M W. Deep multimodal reinforcement network with contextually guided recurrent at-

tention for image question answering. *Journal of Computer Science and Technology*, 2017, 32(4): 738–748. DOI: 10.1007/s11390-017-1755-6.

[50] Cho K, Van Merrienboer B, Bahdanau D, Bengio Y. On the properties of neural machine translation: Encoder-decoder approaches. In *Proc. 8th Workshop on Syntax, Semantics and Structure in Statistical Translation*, Oct. 2014, pp.103–111. DOI: 10.3115/v1/w14-4012.

[51] Shaw P, Uszkoreit J, Vaswani A. Self-attention with relative position representations. In *Proc. the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Jun. 2018, pp.464–468. DOI: 10.18653/v1/n18-2074.

[52] Santoro A, Raposo D, Barrett D G T, Malinowski M, Pascanu R, Battaglia P, Lillicrap T. A simple neural network module for relational reasoning. In *Proc. the 31st International Conference on Neural Information Processing Systems*, Dec. 2017, pp.4974–4983.

[53] Hudson D A, Manning C D. Compositional attention networks for machine reasoning. In *Proc. the 6th Int. Conf. Learning Representations*, Apr. 2018.

[54] Mao J Y, Gan C, Kohli P, Tenenbaum J B, Wu J J. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *Proc. the 7th International Conference on Learning Representations*, May 2019.

[55] Kim J H, Lee S W, Kwak D, Heo M O, Kim J, Ha J W, Zhang B T. Multimodal residual learning for visual QA. In *Proc. the 30th International Conference on Neural Information Processing Systems*, Dec. 2016, pp.361–369.

[56] Saito K, Shin A, Ushiku Y, Harada T. DualNet: Domain-invariant network for visual question answering. In *Proc. the 2017 IEEE International Conference on Multimedia and Expo*, Jul. 2017, pp.829–834. DOI: 10.1109/icme.2017.8019436.

[57] Teney D, Liu L Q, Van Den Hengel A. Graph-structured representations for visual question answering. In *Proc. the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017, pp.3233–3241. DOI: 10.1109/cvpr.2017.344.

[58] Selvaraju R R, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proc. the 2017 IEEE International Conference on Computer Vision*, Oct. 2017, pp.618–626. DOI: 10.1109/iccv.2017.74.



Qiang Sun received his Ph.D. degree from Academy of Engineering & Technology, Fudan University, Shanghai, in 2023, his M.S. degree from the Department of Computer Science and Technology, Nanjing University, Nanjing, in 2011, and his B.S. degree from Software Engineering Institute, Nanjing University, Nanjing, in 2008. He is currently a lecturer at the School of Statistics and Information, Shanghai University of International Business and Economics, Shanghai. His research interests include visual language answering and vision language navigation.



Yan-Wei Fu received his Ph.D. degree from Queen Mary University of London, London, in 2014, and his M.Eng. degree from the Department of Computer Science and Technology, Nanjing University, Nanjing, in 2011. He held a post-doctoral position at Disney Research, Pittsburgh, from 2015 to 2016. He is currently a tenure-track professor at the School of Data Science, Fudan University, Shanghai. His research interests are image and video understanding, and life-long learning.



Xiang-Yang Xue received his B.S., M.S., and Ph.D. degrees in communication engineering from Xidian University, Xi'an, in 1989, 1992, and 1995, respectively. He is currently a professor at the School of Computer Science, Fudan University, Shanghai. His research interests include computer vision, multimedia information processing, and machine learning.