Question Guided Modular Routing Networks for Visual Question Answering

Yanze Wu\textsuperscript{1}, Qiang Sun\textsuperscript{1}, Jianqi Ma\textsuperscript{1}, Bin Li\textsuperscript{1}, Yanwei Fu\textsuperscript{1}, Yao Peng\textsuperscript{2}, Xiangyang Xue\textsuperscript{1}
\textsuperscript{1}Fudan University, \textsuperscript{2}Qiniu Inc
\{wuyanze123,sunqiang85\}@gmail.com, \{majq16,libin,yanweifu\}@fudan.edu.cn, pengyao@qiniu.com, xyxue@fudan.edu.cn

Abstract

Visual Question Answering (VQA) faces two major challenges: how to better fuse the visual and textual modalities and how to make the VQA model have the reasoning ability to answer more complex questions. In this paper, we address both challenges by proposing the novel Question Guided Modular Routing Networks (QGMRN). QGMRN can fuse the visual and textual modalities in multiple semantic levels which makes the fusion occur in a fine-grained way, it also can learn to reason by routing between the generic modules without additional supervision information or prior knowledge. The proposed QGMRN consists of three sub-networks: visual network, textual network and routing network. The routing network selectively executes each module in the visual network according to the pathway activated by the question features generated by the textual network. Experiments on the CLEVR and CLEVR-Humans datasets show that our model can achieve the state-of-the-art performance. Models and Code will be released.

1 Introduction

Visual Question Answering (VQA) (Antol et al. 2015) is a task that, given an image and question pair, the model can reason and answer the question by visual information. It is a popular topic in recent years that has connected the computer vision and natural language processing (NLP). VQA faces two major challenges: 1) How to better fuse the visual and textual modalities; 2) How to make the VQA model have the reasoning ability to answer more complex questions.

In order to fuse the visual and textual modalities, the most common form of VQA model is to extract the visual features from a modern CNN (e.g., ResNet (He et al. 2016) and textual features from a RNN (e.g., LSTM (Hochreiter and Schmidhuber 1997) or GRU (Cho et al. 2014)) separately, and then fuse them into a common latent space (Kazemi and Elqursh 2017; Anderson et al. 2018; Fukui et al. 2016). Feature fusion is explored from simple add or concatenation operation to advanced joint embedding techniques, such as MLB (Kim et al. 2016), MCB (Fukui et al. 2016) and MUTAN (Ben-Younes et al. 2017), and attention mechanisms such as one-hop attention (Kazemi and Elqursh 2017), multi-hop attention (Yang et al. 2016) and bottom-up attention (Anderson et al. 2018). However, these fusion methods are constrained since only the high-level visual features are involved in the fusion, the low-level visual features are not directly interacted with the textual features, which means the fusion occurs in a high semantic level or coarse-grained way.

To endow the VQA model with reasoning ability is difficult because although CNN like ResNet is powerful, it does not support reasoning natively. To make the VQA system support reasoning, relation network (Santoro et al. 2017) has been proposed to explicitly model the relational reasoning by pairwise comparisons over spatial locations. Although the relation network is simple and powerful, it still has main drawbacks like: the computational cost scales quadratically to the number of spatial locations, and the model has a suspicion that is tailored to the CLEVR dataset (Johnson et al. 2017a) which means it is not a general model. Another line of work (Andreas et al. 2016; Johnson et al. 2017b; Hu et al. 2017) believes that natural language questions are compositional, and can be answered by decomposing them into modular sub-problems. So some modules for solving specific sub-problems are artificially designed and a layout is learned from the question to assemble the question-relevant modules to predict the answer. However the design of the module requires prior knowledge and training process needs extra supervision information, which means this kind of methods is lack of generality.

In this paper, we argue that even without dedicated-designed modules and extra supervision information, we can still make the model support compositional reasoning. Specifically, we propose a novel model called Question Guided Module Routing Networks (QGMRN) that consists of three sub-networks: visual network, textual network and routing network. The visual network is based on a generic modular network that each layer of the network is composed of some generic modules, as the module granularity changes, different modular networks can be spawned, in this paper, we will give two variants. When the input reaches a certain layer, the visual network dynamically selects a portion of modules from that layer to process the input according to the binary gates generated by the routing network, we combine the binary gates of each layer and call them the routing path. Therefore, the routing network is responsible for receiving the question features generated by the textual
network and mapping them to a discrete routing path. We claim that the routing path represents the process of compositional reasoning, which also increases the interpretability to some extent, we verify this by visualizing the routing paths in the experiments. Another benefit brought by the routing network is that we can fuse the visual and textual modalities in all semantic levels, since the routing network starts controlling the visual network at a very early stage.

Extensive experiments on the CLEVR and CLEVR-Humans datasets show that our model outperforms the standard VQA models by a large margin, and also achieves the state-of-the-art performance. Since CLEVR and CLEVR-Humans are challenging datasets that requires reasoning, which verifies our model has strong reasoning ability. The visualization of the routing paths also demonstrates the potential interpretability of our model.

To sum up, the contributions of this paper can be summarized as: (1) The proposed model can fuse the visual and textual modalities from low level to high level, that is, the fusion occurs in a fine-grained way. In addition, the model is equipped with strong reasoning ability; (2) To the best of our knowledge, this is the first time that the binary gated modular routing network conditioned on the question is applied in the VQA task. At the same time, this model can be regarded as a general framework. It only needs to change the granularity of the module to generate different interesting models; (3) We conduct extensive experiments on CLEVR and CLEVR-Humans datasets to show the superiority of our model, we also conduct various visualizations to analysis our model in-depth.

2 Related Work

2.1 Fusion Strategy of VQA

Usually, the first step of most VQA methods is to extract high-level visual features from a modern CNN and textual features from an RNN separately. In order to combine the visual and textual modalities to make the answer, many methods have been proposed to fuse the extracted visual and textual features. Multimodal Low-rank Bilinear pooling (MLB) (Kim et al. 2016) provides an efficient method to approximate the full bilinear pooling by forcing the weight matrix to be low-rank. Multimodal Compact Bilinear pooling (MCB) (Fukui et al. 2016) randomly projects the visual and textual features into a higher dimensional space, then convolves them in Fast Fourier Transform space. Multimodal Tucker Fusion (MUTAN) (Ben-Younes et al. 2017) proposes a general fusion method based on Tucker decomposition, which covers MLB and MCB. Yang et al. (Yang et al. 2016) propose a multi-hop spatial attention to fuse the visual and textual features so that the image regions related to the question will be focused. Anderson et al. (Anderson et al. 2018) propose a bottom-up model that combines the attention mechanism with object-level visual features, so that objects related to the question will be focused. However, these methods only fuse features at a high level, which will make the fusion occur in a coarse-grained way.

Another line of work proposes to fuse two modalities by using the question to predict the parameters of the visual network, due to the number of parameters in the visual network is usually too large, only a small portion of parameters can be learned from the question feature. For example, Gao et al. (Gao et al. 2018) propose the Question-guided Hybrid Convolution (QGHC) based on group convolution, which consists of question-dependent kernels and question-independent kernels: the parameters of batch-norm (Ioffe and Szegedy 2015) layers in MODERN (De Vries et al. 2017) are predicted by the question; FiLM (Perez et al. 2017) implements the condition through a general feature-wise transformation. Although this kind of methods can fuse the visual and textual modalities in a multi-level and fine-grained way, these methods still have limitations. If too many parameters are learned from the question, that will make the model difficult to train, but if only a few parameters are learned from the question, that will constrain the model’s learning capacity. Another concern is about the flexibility, e.g., QGHC can only be applied in the ResNext (Xie et al. 2017) architecture, MODERN can only be applied in CNN with batch-norm layers.

Our proposed model can fuse the visual and textual modalities in a multi-level and fine-grained way, more importantly, our model is very flexible. When we adjust the granularity of the module to filter level, MODERN and FiLM are related to our model, but with fundamental differences. First, they modulate the visual network at a feature level, but we modulate the visual network at a sub-network level. Second, we route the modules in a discrete way, which is a step towards discrete reasoning.

2.2 Neural Module Network.

In order to support compositional reasoning, Andreas et al. (Andreas et al. 2016) advocate a general purpose neural module network (NMN) which is dynamically instantiated from a collection of reusable modules based on the compositional structure of the question. Although the function of each module is learned from training, the question parser and the mapping rules from the parsing tree node to the module must be pre-defined. The performance of NMN model heavily relies on the quality of the question parser chosen. Further, Hu et al. (Hu et al. 2017) propose an End-to-End Module Network which predicts the module layout by an LSTM instead of an external question parser. Johnson et al. (Johnson et al. 2017b) propose a model combining with both program generator and execute engine based on neural module network. However, these models model still need prior knowledge about the specific purpose module or extra supervision information for training which is a limitation.

Different from the above methods, our model supports reasoning even without dedicated-designed modules and extra supervision information for training, no prior knowledge is needed.

2.3 Routing Models

The proposed modular routing network is related to conditional computation (CC) (Bengio, Léonard, and Courville 2013; Bengio et al. 2016) and mixture-of-experts (MoE) (Jordan and Jacobs 1994; Jacobs et al. 1991). Eigen,
Given the question \( Q = [w_1, w_2, \cdots, w_n] \) where \( w_i \) is the one-hot representation of \( i \)th word and \( n \) is the length of the question, we first use a lookup layer to embed \( Q \) into \( E_q = [e_1, e_2, \cdots, e_n] \) where \( e_i \in \mathbb{R}^d \), \( d \) is the embedding size. Then we feed \( E_q \) into Gated Recurrent Units (GRU) and use the final hidden state of GRU as the question features:

\[
q = \text{GRU}(E_q) \tag{1}
\]

where \( q \in \mathbb{R}^h \) and \( h \) is the hidden size of GRU units.

### 3.1 Textual Network

The whole model is differentiable with respect to the model parameters and can be trained end-to-end. The textual network takes a question and generates question features, the routing network takes the question features to generate a specific routing path, the visual network takes the raw image and dynamically chooses which modules are executed and performs the reasoning task. Finally, the extracted image features are sent to the classifier to predict the answer, the architecture of the model can be seen in Figure 1.

### 3.2 Visual Network

To better illustrate our approach, we will first introduce the generic Modular Architecture and Modular Routing Architecture. Modular Architecture defines granularity and the organization structure of the modules. Module Routing Architecture defines how the routing path controls over the Modular Architecture. Then we will introduce two forms of implementations of the module routing architecture: the first type, we call it FRN (Filter Routing Network) which is based on ResNet (He et al. 2016); the second type, we call it BRN (Branch Routing Network) which is based on ResNext (Xie et al. 2017).

#### Modular Architecture

We assume that a generic modular architecture is composed of \( L \) modular layers, each modular layer is composed of \( M \) modules\(^2\); each module \( f_{l,m}(x; \theta_{l,m}) \) for \( l \in \{1, \cdots, L\} \) and \( m \in \{1, \cdots, M\} \) is a function module that takes the input \( x \) and generates an output tensor, \( \theta_{l,m} \) is the learnable parameter of \( f_{l,m} \). The modules in the \( l \)th layer share the same input, and the output of the \( l \)th layer \( y_l \) is:

\[
y_l = \phi([\tilde{f}_{1,1}(y_{l-1}; \theta_{1,1}), \cdots, \tilde{f}_{1,M}(y_{l-1}; \theta_{1,M})]) + y_{l-1} \tag{2}
\]

where the composite function \( \phi(\cdot) \) can either be concatenation or summation.

#### Modular Routing Architecture

First, we denote the routing path as

\[
P = \text{RNET}(q) \tag{3}
\]

where RNET is the routing network that generates a routing path \( P \in \{0, 1\}^{L \times M} \) conditioned on the question features \( q \in \mathbb{R}^h \). \( P_{l,m} \) for \( l \in \{1, \cdots, L\} \) and \( m \in \{1, \cdots, M\} \) is a binary gate that controls whether or not to execute the \( m \)th module of the \( l \)th layer. The output of the \( l \)th layer \( y_l \) of modular routing architecture is now changed to:

\[
y_l = \phi([\tilde{f}_{1,1}(y_{l-1}; \theta_{1,1}), \cdots, \tilde{f}_{1,M}(y_{l-1}; \theta_{1,M})]) + \tilde{y}_{l-1} \tag{4}
\]

where \( \tilde{f}_{l,m}(\tilde{y}_{l-1}; \theta_{l,m}) = P_{l,m} f_{l,m}(\tilde{y}_{l-1}; \theta_{l,m}) \)

Due to the discrete nature of \( P \), the gradient backpropagation algorithm cannot be applied here, we will use some tricks to make the whole model differentiable and the details will be discussed in section 3.3.

#### FRN and BRN

As we claimed before, with the introduced modular routing architecture, different models can be realized by changing the granularity of the module. Here we introduce two kinds of special cases that are easy to implement.

**FRN:** when the routed module is the filter in a convolutional layer, we call it Filter Routing Network (FRN). Note that the FRN can be plugged into any modern CNN, we applied it to the current popular ResNet for comparison. More specifically, we route the last convolutional layer of each residual block.

**BRN:** when the routed module is the branch of a multi-branch network, we call it Branch Routing Network (BRN).

---

\(^1\)We also tried using both image features and question features as condition, the details are provided in the supplemental material.

\(^2\)For convenience, we assume that the number of modules per modular layer is the same, but in fact they can be different.
The BRN can be plugged into any multi-branch convolutional network like Inception (Szegedy et al. 2015) or ResNext. In this paper, we test the BRN under the ResNext model.

3.3 Routing Network

As we claimed before, RNET aims to compute the binary gates conditioned on the question features. First, we temporarily ignore the fact that the routing path is discrete, then things become simple, all we need is a fully connected layer $f : \mathbb{R}^h \rightarrow \mathbb{R}^{L \times M}$ to map the question features to a real-valued routing path $\hat{P} = f(c(q))$. Now take into account the nature of the discrete, a naive attempt would be thresholding $\hat{P}$ into 1s and 0s, but unfortunately, it is not differentiable and the backpropagation algorithm cannot be applied here. Also note that the threshold function is deterministic, in order to find more possible paths, the generation of routing path will better to be stochastic. Based on the above considerations, we employ a reparameterization trick called Concrete Distribution (Maddison, Mnih, and Teh 2017) or Gumbel Softmax (Jang, Gu, and Poole 2017) in this paper.

In order to elicit this method, we first review the Gumbel-Max trick (Lu et al. 2016; Yellott Jr. 1977) which provides a way to sample $z$ from a categorical distribution with class probability of $\pi_1, \pi_2, \cdots, \pi_n$ as follows:

$$ z = \text{one-hot} \left( \arg \max_{i} \{ g_i + \log \pi_i \} \right) \quad (5) $$

where $P(z_k = 1) = \pi_k$

where $g_1, \cdots, g_n$ are i.i.d samples from Gumbel distribution $\mathcal{G}$.

But the argmax operation is still not differentiable, so the softmax function with temperature $\tau$ is introduced here to approximate the argmax function:

$$ \tilde{z}_i = \frac{\exp((\log(\pi_i) + g_i)/\tau)}{\sum_{j=1}^{n} \exp((\log(\pi_j) + g_j)/\tau)} \quad \text{for} \ i = 1, \cdots, n \quad (6) $$

as $\tau \to 0$, the softmax function is smoothly approaching the argmax function.

In the above we have showed how to sample a categorical distribution $\text{Cat}(\pi_1, \pi_2, \cdots, \pi_n)$ using the Gumbel-Softmax trick, now we discuss the “binary” case of our problem, that means we need to sample from $\text{Cat}(\pi_1, \pi_2)$ where $\pi_2 = 1 - \pi_1$. Note that this distribution is equivalent to Bernoulli distribution $\text{Bern}(\pi_1)$, we use $\rho$ to substitute $\pi_1$ to distinguish from the previous symbol definition, then we apply the Gumbel-Max trick again to sample Bernoulli variable $b$ from $\text{Bern}(\rho)$ as

$$ b = \begin{cases} 1, & \text{if} \ g_1 + \log(\rho) > g_2 + \log(1 - \rho) \\ 0, & \text{else} \end{cases} \quad (7) $$

where

$$ P(b = 1) = P(g_1 + \log(\rho) > g_2 + \log(1 - \rho)) \quad (8) $$

$$ = P(g_1 - g_2 + \log(\frac{\rho}{1 - \rho}) > 0) \quad (9) $$

$$ = P(t + \log(\frac{\rho}{1 - \rho}) > 0) \quad (10) $$

the derivation of Eq.(10) uses the fact that the difference of two Gumbels is a Logistic distribution $\mathcal{L}(g_1 - g_2) \sim \text{Logistic}(0, 1)$ and $t \sim \text{Logistic}(0, 1)$. Just as we use the softmax function with temperature to approximate the argmax function, here we use the sigmoid function with temperature $t$ to sample from $\text{Bern}(\rho)$.

\footnote{To sample from Logistic distribution, computing random variable $t$ as $t = \log(u) - \log(1 - u)$ where $u \sim \text{Uniform}(0, 1)$, then $t \sim \text{Logistic}(0, 1)$}
to approximate the unit step function:
\[
\tilde{b} = \frac{1}{1 + \exp(-(t + \log(\frac{\tilde{B}_{l,m}}{\tau}))/\tau)}
\]
as \(\tau \to 0\), the sigmoid function is smoothly approaching the unit step function. Readers who are interested in Gumbel-Softmax or Concrete Distribution are referred to (Maddison, Macih, and Teh 2017) [Jang, Gu, and Poole 2017] for more details.

Based on previous discussion, we can convert the real-valued \(\tilde{P}\) to binary value in a simple way. First, for each entry \(\tilde{P}_{l,m}\), we sample \(\tilde{B}_{l,m}\) as:
\[
\tilde{B}_{l,m} = \frac{1}{1 + \exp(-(t_{l,m} + \tilde{P}_{l,m}/\tau)}
\]
where \(t_{1,...,L,1,...,M}\) are i.i.d samples from Logistic distribution. That means each entry of \(\tilde{P}\) learns an independent logit \(\log(\frac{\tilde{B}_{l,m}}{1-\tilde{B}_{l,m}})\). However \(\tilde{B}_{l,m}\) is continuous value as \(\tau > 0\), here we use a Straight-Through (ST) method introduced in (Bengio, Layonard, and Courville 2013) to convert the continuous \(\tilde{B}_{l,m}\) to discrete value, that is, during forward process, we use a threshold of 0.5 to thresholding \(\tilde{B}_{l,m}\) to 0 and 1, but during the backward process, the gradient is normally passed to \(\tilde{B}_{l,m}\) just as the thresholding function is an identity function.

### 3.4 Training Loss

In order to avoid model collapse and prevent certain modules from being always executed or always not executed, some sparsity and variance regularizations (Bengio et al. 2016) are introduced. In this paper, we do not require our model to be sparse. But we add a regularization called total balanced loss \(L_{\text{balance}}\), which was first introduced in (Shazeer et al. 2017) to prevent model collapse. We define the number of times the \(m\)th module in the \(l\)th layer is executed in a batch as module importance \(O_{l,m}\). Further, we define the square of the coefficient of variation (CV) for the module importance in the \(l\)th layer as the layer balanced loss \(L_{\text{layer}}\) to encourage the modules in the \(l\)th layer being involved equally. The sum of layer balanced loss is defined as total balanced loss of \(L_{\text{balance}}\):
\[
O_{l,m} = \sum_{i=1}^{N} P_{l,m}^{(i)}
\]
\[
L_{\text{layer}} = \text{CV}(O_{l,:})^2
\]
\[
L_{\text{balance}} = \sum_{l=1}^{L} L_{\text{layer}}
\]
where \(P_{l,m}^{(i)}\) is the gate of the \(m\)th module in the \(l\)th layer for the \(i\)th instance in the mini-batch, \(N\) is the batch size. With the standard VQA loss \(L_{\text{vqa}}\), the full loss is:
\[
L = L_{\text{vqa}} + \lambda L_{\text{balance}}
\]

where \(\lambda\) is a hyper-parameter.

## 4 EXPERIMENTS

### 4.1 Datasets

The proposed method is evaluated on two datasets: (1) CLEVR dataset (Johnson et al. 2017a) is proposed to study the ability of VQA systems to perform reasoning. Answering question about a CLEVR image requires various kinds of reasoning, which makes the standard VQA methods perform poorly on the dataset. The dataset contains 100K 3D-rendered images and about one million automatically-generated questions. Specifically, the question in the dataset can be divided into the following five types: Count: ask the number of certain objects; Exist: ask whether a certain object is present; Compare Numbers: ask which of two particular objects is larger; Query Attribute: query a attribute of particular object; Compare Attribute: ask whether two particular objects have same value on some attribute. Figure 2 give examples of (image, question, answer) tuple sampled from CLEVR dataset. (2) CLEVR-Humans dataset (Johnson et al. 2017b) contains human-posed questions on CLEVR images, which makes the dataset more complex and realistic. We train all of our models with the official training set and test the models on the official validation set, and we train our models from raw pixels, the image shape is resized to 480 × 360.

### 4.2 Implementation details

**Configuration of BRN and FRN:** BRN has 8 modular routing layers, and each layer has 32 modules. In the terminology of the ResNext paper, BRN uses bottleneck block with group convolution, the cardinality is 32 and the depth is \(26 = 3 \times 8 + 2\). FRN is based on ResNet34, that is, each layer is a basicblock and has 16 modular layers in total, the number of modules for each layer depends on the second convolutional layer’s filter number.

**Visual Network:** all images are resized to \(480 \times 360\) and we do not use any data augmentation. In order to achieve the best performance, we concatenate the two coordinate feature maps indicating relative x and y spatial position on the feature map before sending to the classifier.

**Textual Network:** the word embedding size is set to 200, the GRU hidden size is set to 512, we observe that the hidden size set to 512 or 1024 has little effect on the final accu-
| Model                                      | Overall | Count | Exist | Compare Numbers | Query Attribute | Compare Attribute |
|-------------------------------------------|---------|-------|-------|-----------------|-----------------|-------------------|
| Human (Johnson et al. 2017b)              | 92.6    | 86.7  | 96.6  | 86.5            | 95.0            | 96.0              |
| LSTM (Johnson et al. 2017b)               | 46.8    | 41.7  | 61.1  | 69.8            | 36.8            | 51.8              |
| CNN+LSTM (Johnson et al. 2017b)           | 52.3    | 43.7  | 65.2  | 67.1            | 49.3            | 53.0              |
| CNN+LSTM+MCB (Fukui et al. 2016)          | 51.4    | 42.1  | 63.4  | 66.4            | 49.0            | 51.0              |
| CNN+LSTM+SA (Santoro et al. 2017)         | 76.6    | 64.4  | 82.7  | 77.4            | 82.6            | 75.4              |
| N2NMN (Hu et al. 2017)                    | 83.7    | 68.5  | 85.7  | 84.9            | 90.0            | 88.7              |
| PG+EE (9K prog.) (Johnson et al. 2017b)   | 88.6    | 79.7  | 89.7  | 79.1            | 92.6            | 96.0              |
| PG+EE (700K prog.) (Johnson et al. 2017b) | 96.9    | 92.7  | 97.1  | **98.7**        | 98.1            | 98.9              |
| FiLM (Perez et al. 2018)                  | 97.6    | 94.3  | 99.3  | 94.9            | 99.3            | 99.3              |
| FiLM with 12 ResBlocks (Perez et al. 2018)| 96.9    | –     | –     | –               | –               | –                 |
| QGHC (Gao et al. 2018)                    | 86.3    | 78.1  | 91.7  | 80.7            | 89.4            | 86.8              |
| Relation Network (Santoro et al. 2017)    | 95.5    | 90.1  | 97.8  | 93.6            | 97.9            | 97.1              |
| BRN (ours)                                | 88.5    | 83.9  | 94.8  | 80.0            | 90.1            | 91.9              |
| FRN (ours)                                | **97.8**| **95.7**| **99.6**| **93.6**       | **98.5**        | **99.7**         |

Table 1: Overall accuracy and accuracy of five question types on the CLEVR validation set.

racy. The parameters of the GRU and word embedding layer are initialized with orthogonal initialization and uniform initialization respectively.

**Routing Network:** although the temperature $\tau$ can be annealed to a small value during training, we find that just keep it a constant value 1.0 can get decent accuracy. Note that we need the generation of routing path to be stochastic to exploit more possible routing paths during training, but during test phrase, we want the generation of routing path to be deterministic, so just use the sigmoid function with a threshold 0.5 to convert the $P$ to $P$. We also initialize the parameters of the routing network so that the probability of each module being executed at the beginning is 0.7.

**Training:** all the models are trained with ADAM (Kingma and Ba 2015) optimizer, betas are set to (0.9, 0.999). We observe that applying a warmup scheme (Goyal et al. 2017) can help the model to achieve better performance, i.e., we start training our model with a small learning rate 3e-6, and slowly increase the learning rate until it reaches 3e-4, then use 3e-4 to train the model until the end.

### 4.3 Results on CLEVR

The results of all the compared methods on CLEVR are shown in Table 1, as we claimed before, the standard VQA methods like CNN+LSTM, CNN+LSTM+MCB, CNN+LSTM+SA perform poorly on this challenging dataset. Noted that the N2NMN and PG+EE can also be regarded as a modular network, although the sub-modules are designed elaborately and the extra supervision programs are used, our generic model that is not specifically designed for CLEVR dataset still performs better than the two modular networks without extra supervision information. Compared with QGHC which is also a “question guided” network, our model still outperforms it by a large margin. Compared with strong opponents FiLM and Relation Network, our best-performed model still outperforms them. In order to make a fair comparison with FiLM, we also report the FiLM with 12 ResBlocks whose parameter amounts are nearly same as our FRN. Our model also surpasses human in all the question categories.

In conclusion, our best-performed model outperforms standard VQA methods by a large margin, compared with state-of-the-art methods, our model can still achieve the same level or better results than them. Specially, on Count question type, our model outperforms all other methods by a large margin, on Exist and Compare Attribute question types, our model has reached nearly 100% accuracy! Note that although higher accuracy is reported in MAC (Hudson and Manning 2018), but we believe that our model is simpler and more interpretable, also, nor spatial attention and bidirectional rnn are used in our model.

**BRN vs FRN**  Notice that the performance of BRN is not as good as FRN, we suspect that it may be because the number of modules per layer is too small, resulting in a representation capability that is not as strong as FRN. However BRN still outperforms QGHC which is also based on ResNext, this also reflects that routing is really useful.

### 4.4 Results on CLEVR-Humans

| Model                                      | Val Accuracy |
|-------------------------------------------|--------------|
| LSTM (Johnson et al. 2017b)               | 36.5         |
| CNN+LSTM (Johnson et al. 2017b)           | 43.2         |
| CNN+LSTM+SA (Johnson et al. 2017b)        | 57.6         |
| PG+EE (18K prog.) (Johnson et al. 2017b)  | 66.6         |
| FiLM (Perez et al. 2018)                  | 75.9         |
| FRN (ours)                                | **79.9**     |

Table 2: Accuracy of compared methods on val set of CLEVR-Humans dataset.

To further validate the reasoning ability of our model, we next provide the results on CLEVR-Humans. To make a fair comparison with other methods, the reported result is fine-tuned from previous best performed model on CLEVR.
A dataset, i.e., FRN. Also, pretrained word embeddings are not used.

As shown in Table 2, our model achieve state-of-the-art performance on CLEVR-Humans again, and outperforms compared methods by a large margin. This shows that our model has better robustness and strong reasoning ability.

Figure 3: The t-SNE visualization of routing paths of stage1, stage2, stage3 and stage4 are represented at top-left, top-right, bottom-left and bottom-right sub-figures respectively. Better viewed in color.

4.5 Visualization on CLEVR dataset

T-SNE visualization of routing paths

To investigate what the routing network learns, we use t-SNE (Maaten and Hinton 2008) to visualize the routing paths in 2D embedding. Specifically, we divide the routing path into 4 stages according to the downsampling in the visual network, then flatten the routing path of each stage into a vector, and finally project it to the 2D space through t-SNE. For efficiency, we did not visualize the entire CLEVR dataset, but randomly select 500 instances (i.e., image question pair) on the validation set for each officially provided question subtype (i.e., subtypes subdivided from the previous 5 types). The visualization and detailed 13 question subtypes can be seen in Figure 3. Each point represents an instance, and points of the same question subtype are labeled with the same color. From the figure, we can discover many interesting phenomenons, we list some of them as follows: a) as the stage increases, data points with different question types can be discriminated better; b) The visualization of stage1 may be confusing at first glance, but note that data points with question subtype of query_color and equal_color are clustered together and data points with question subtype of query_material and equal_material are clustered together, this makes sense because the first few layers of CNN are responsible for detecting features about colors, textures, and edges (Olah, Mordvintsev, and Schubert 2017); c) data points with the question subtype belonging to the same question type (e.g., greater_than, less_than and equal_integer belong to the Compare Numbers type) are tend to cluster together. d) data points with the question type of count and exist are clustered together.

Figure 4: The module executed ratio in each layer.

How many modules are executed for each layer

To investigate how many modules are executed in each layer, we plot the bar graph of the module executed ratio (i.e., the number of executed modules divided by the total number of the modules) for each layer in the validation set, it can be seen in Figure 4. From the figure, we can discover that the first few layers have a high executed ratio, the latter layers have a lower executed ratio, and the executed ratio will not be lower than 0.7. This may tell us that most of the features extracted at low levels are generic, but the features extracted at higher layers are more relevant to the specific question.

5 Conclusion

In this paper, we propose the Question Guided Modular Routing Networks for VQA which can fuse the visual and textual modalities in multiple semantic levels and learn to reason by routing between the generic modules, different interesting variant models can be generated by changing the granularity of the module. In the experiments, we show that our model achieves the state-of-the-art performance. In particular, we find a suitable application for routing models where static models struggle, and we also successfully applied routing models to large model and large-scale dataset. We believe that routing models will play an important role in future multimodal fusion and embedding.

References

[Anderson et al. 2018] Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In CVPR.

[Andreas et al. 2016] Andreas, J.; Rohrbach, M.; Darrell, T.; and Klein, D. 2016. Neural module networks. In CVPR.

[Antol et al. 2015] Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Lawrence Zitnick, C.; and Parikh, D. 2015. Vqa: Visual question answering. In ICCV.
[Veit and Belongie 2018] Veit, A., and Belongie, S. J. 2018. Convolutional networks with adaptive inference graphs. In *ECCV*.

[Wu et al. 2018] Wu, Z.; Nagarajan, T.; Kumar, A.; Rennie, S.; Davis, L. S.; Grauman, K.; and Feris, R. S. 2018. Block-drop: Dynamic inference paths in residual networks. *CVPR*.

[Xie et al. 2017] Xie, S.; Girshick, R. B.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. *CVPR* 5987–5995.

[Yang et al. 2016] Yang, Z.; He, X.; Gao, J.; Deng, L.; and Smola, A. 2016. Stacked attention networks for image question answering. In *CVPR*, 21–29.

[Yellott Jr 1977] Yellott Jr, J. I. 1977. The relationship between luce’s choice axiom, thurstone’s theory of comparative judgment, and the double exponential distribution. *Journal of Mathematical Psychology* 15(2):109–144.