Linguistically Routing Capsule Network for Out-of-distribution Visual Question Answering

Qingxing Cao[1], Wentao Wan[2], Keze Wang[2], Xiaodan Liang[1], Liang Lin[2*]
[1]Shenzhen Campus of Sun Yat-sen University, [2]Sun Yat-Sen University
caoqx8@mail.sysu.edu.cn, wentao.wan@qq.com,
kezewang@gmail.com, xdliang328@gmail.com, linliang@ieee.org

Abstract

Generalization on out-of-distribution (OOD) test data is an essential but underexplored topic in visual question answering. Current state-of-the-art VQA models often exploit the biased correlation between data and labels, which results in a large performance drop when the test and training data have different distributions. Inspired by the fact that humans can recognize novel concepts by composing existed concepts and capsule network's ability of representing part-whole hierarchies, we propose to use capsules to represent parts and introduce "Linguistically Routing" to merge parts with human-prior hierarchies. Specifically, we first fuse visual features with a single question word as atomic parts. Then we introduce the "Linguistically Routing" to reweight the capsule connections between two layers such that: 1) the lower layer capsules can transfer their outputs to the most compatible higher capsules, and 2) two capsules can be merged if their corresponding words are merged in the question parse tree. The routing process maximizes the above unary and binary potentials across multiple layers and finally carves a tree structure inside the capsule network. We evaluate our proposed routing method on the CLEVR compositional generation test, the VQA-CP2 dataset and the VQAv2 dataset. The experimental results show that our proposed method can improve current VQA models on OOD split without losing performance on the in-domain test data.

1. Introduction

The task of visual question answering (VQA) is to correctly answer a question about an image. It is regarded as a core task towards the complete AI [5] as it requires a vast range of knowledge across multiple domains. However, the complexity of this task also makes it impossible to annotate enough training data that can cover all background knowledge and reasoning routes. Thus, a VQA model must capable of generalizing on out-of-distribution (OOD) test data to handle the unconstrained VQA tasks for practical application.

Current state-of-the-art VQA models [47, 40, 41] focus on increasing models' capacity, but tend to catch the superficial correlation between questions and answers [17, 24]. As such correlation only holds on training distribution, their performance drops on test data that have a different distribution. Other works [42, 49, 21, 26, 23] have explored structured models to represent atomic elements(e.g. object size, color, or relationships) and then integrate elements to infer the final results. These methods have better interpretability and generalization ability but perform worse than state-of-the-art neural networks on general and unconstrained in-domain test data.

Humans can recognize novel concepts by incorporating learned concepts [30]. This compositional generalization ability allows people to solve a plethora of problems using a limited set of basic skills and is one of the major differences between human intelligence and the current deep neural networks. Meanwhile, the capsule network [38, 20, 19] has the potential to connect the end-to-end neural network with part-based models [14] that represent a sample with part-whole hierarchies. Each capsule can be used to represent a certain part and the routing process can be used to model the hierarchical structure. Although the capsule network demonstrates interesting grouping properties in some toy experiments, it still shows unsatisfactory results on the large-scale image datasets since the diverse visual compositions cannot be captured by learning grouping weights in a black-box manner without proper guidance.

Thus, we propose to inject the human-developed structure into the capsule network to improve neural networks' compositional generalization ability while maintaining their performance on in-domain settings, as shown in Figure 1.

*Corresponding author is Liang Lin. Qingxing Cao and Xiaodan Liang are with the School of Intelligent Systems Engineering, Shenzhen Campus of Sun Yat-sen University, China. Wentao Wan, Keze Wang and Liang Lin is with the School of Computer Science and Engineering, Sun Yat-sen University, China.

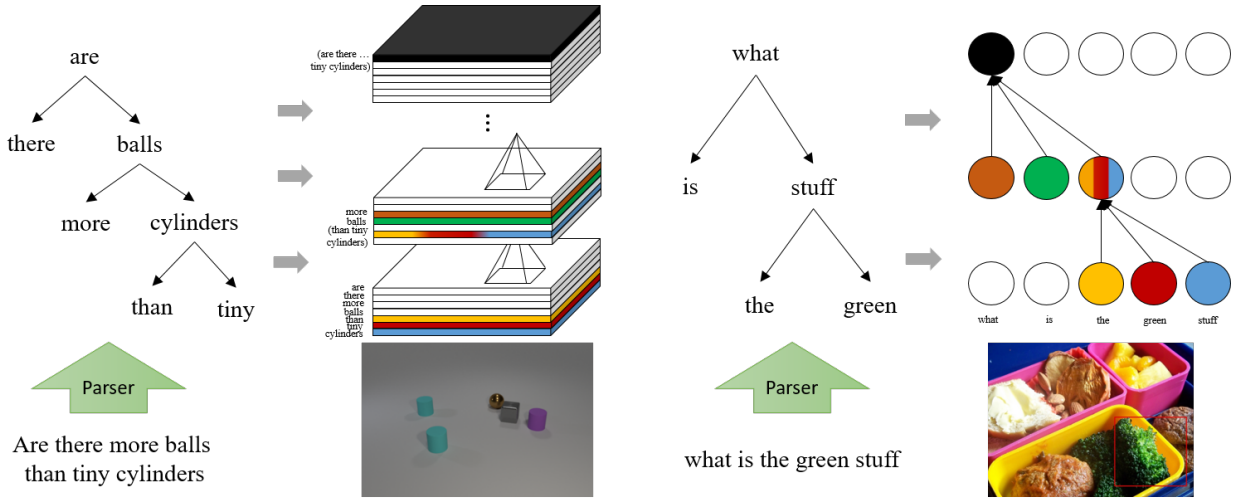

Figure 1: Our proposed Linguistically Routing aims to merge capsules from bottom to top following the linguistic parse tree guidance. Each row or circle represents a capsule and each color represents an encoded word. This compositional process is performed across multiple layers and results in the tree structure inside the capsule network.

Specifically, we propose the "Linguistically Routing" to generate the adaptive reasoning routines inside the capsule network with the guidance of the question parse tree. We first fuse each visual capsule with a single question word to obtain the multimodal representation of the image and question fragment. At each layer, the linguistically routing generates the reweighting vector between $[0, 1]$ for each capsule, such that: 1) only the most compatible higher capsules are activated and receive the outputs from each lower capsule, and 2) two capsules should be merged if their corresponding question fragments are merged in the parse tree. To meet the above two requirements, the proposed linguistically routing learns to predict the unary potentials that select the most representative capsules for each specific sample; and generates the binary potentials that indicate whether two capsules should be merged or not. The linguistically routing maximizes the unary and binary potential with a conditional random field (CRF). After forwarding all layers, a composing structure isomorphic to the parse tree is carved inside the networks, and the capsules from the bottom to the top layer can encode the question words, phrases, clauses, and finally a sentence.

Our contribution can be summarized as follows. 1) We propose an end-to-end trainable routing method that can incorporate external structure information into the capsule network. 2) We propose to utilize the linguistic parse tree to guide the routing and tailor it to the visual question answering task. 3) We perform extensive experiments and show the proposed linguistically routing capsule network can obtain good generalization capability while maintaining performance on in-domain test data.

2. Related Works

Visual question answering. The VQA task requires co-reasoning over both images and text to infer the correct answer. Earlier works used the CNN-LSTM-based architecture and attention mechanism to train the neural networks in an end-to-end manner [44, 39, 50, 45, 32]. Later, lots of works [15, 28, 48, 6] focused on the joint embedding of image and question. Most recently, state-of-the-art methods [47, 40, 41] exploited transformer-based architecture to embed questions and image regions simultaneously. However, it has been argued that these black-box models might exploit the dataset bias instead of understanding the questions and images [17, 24]. This argument has led to the proposal of unbiased datasets [17, 25, 16, 27] and OOD dataset [1]. Recent OOD VQA methods [37, 8, 13] trained a question-only model to predict the answer and used the trained model as a regularizer to reduce the dataset biases and improve the performance on OOD test data.

Structured and interpretable model. Besides the end-to-end neural networks, other methods tried to incorporate additional structured information to improve the compositional reasoning and generalization ability. The neural modular networks [4, 3, 21] use neural modules to solve a particular subtask and assemble them following a structured layout to predict the final answer. [42, 35] used scene graph as an additional signal then applied a GRU or graph convolution network to obtain the question-specific graph representation. [34] also utilized a graph convolution network but embedded the extra retrieved knowledge. PTGRN [9] performed the interpretable reasoning process guided by the dependency parse tree. [46, 23] transformed images

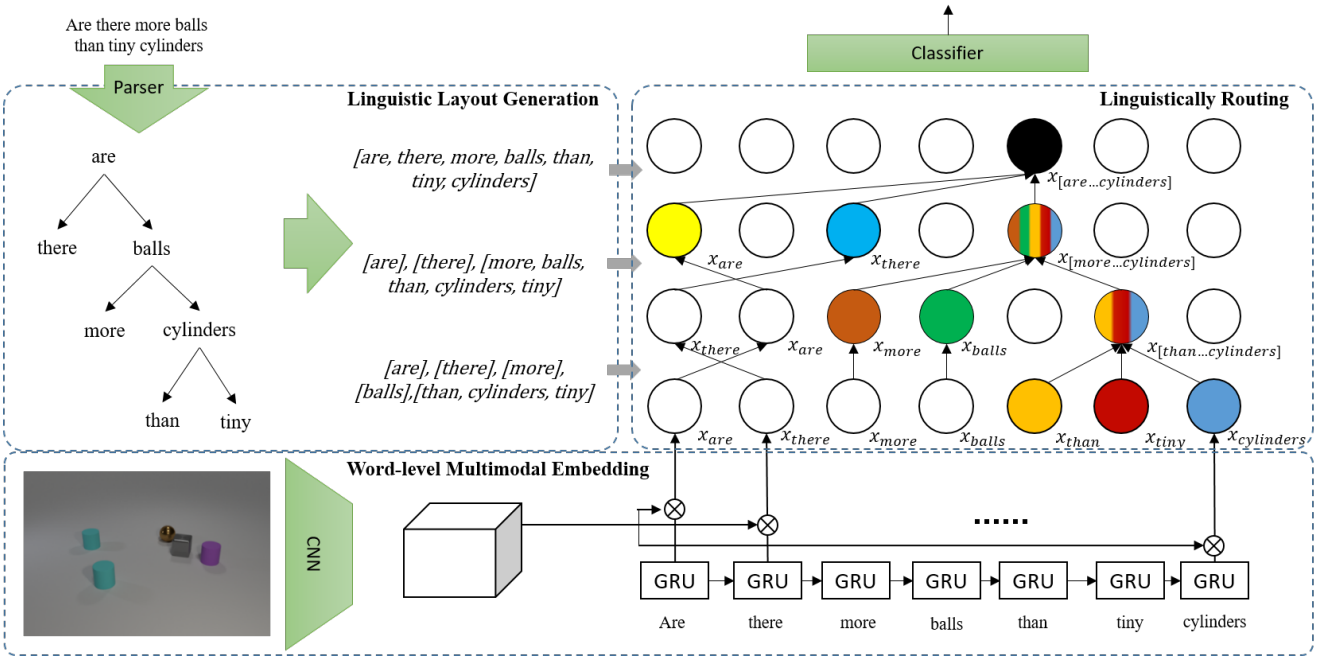

Figure 2: Overview of our linguistically routing within the capsule network. We first generate the linguistic layout and fuse the extracted image feature with individual words as the network inputs. In each layer, the capsules are forwarded to the next layer following the linguistic layout guidance. We use different colors to indicate different encoded words. Finally, the routing process carves the parse tree-like structure, which is represented by the colored circles.

to scene graphs and performed symbolic inference on the graph.

Capsule network. Sabour and Hinton *et al.* [38, 20] proposed to divide each layer in a neural network into many small groups of neurons called "capsules". The capsules can represent various properties of an object, and the capsule routing can activate certain higher-level capsules if their represented property appears in a certain sample. However, the existing capsule network studies learn the grouping weights based on the discriminating loss only. They do not incorporate human priors and have not been evaluated on large-scale datasets.

Compared with the above methods, our proposed linguistically routing aims at incorporating the humor-prior structure inside the network without hard restrictions on the information flow. Further, since the linguistic structure is applied on the visual feature level, our method can combine with other state-of-the-art to improve their generalization ability while maintaining their expressive power on in-domain data.

3. Linguistically Routing Capsule Network

Given the questions Q, images I, our proposed Linguistically Routing is to align the capsule network's routing weights $\mathbf{R}$ with the question parse tree for predicting the answer y. As shown in Figure 2, we first parse a question and transform it into a linguistic layout G. Then, we fuse the image feature and each word in the questions. We denote each resulting feature capsule as $\mathbf{x}_i^0$, and use a vector $\mathbf{c}_i^0 \in \mathbb{R}^{n_q}$ to represent which word is encoded by the i-th capsule, where n_q is the maximum length of the questions. All capsules are concatenated as the network's input $\mathbf{X}^0$.

In each layer l, we have n_c capsules $\mathbf{X}^l = \{\mathbf{x}_i^l\}_{i=1:n_c}$, their encoded words $\mathbf{C}^l = \{\mathbf{c}_i^l\}_{i=1:n_c}$, and the linguistic layout g^{l+1}. The linguistically routing process aims at generating the reweighting vector $\mathbf{R}^l = \{r_{ij}^l\}_{i,j=1:n_c}$ for each pair of capsule $\mathbf{x}_i^l$ and $\mathbf{x}_j^{l+1}$, such that each lower-level capsule $\mathbf{x}_i^l$ can activate a proper high-level capsule $\mathbf{x}_j^{l+1}$, while two lower-level capsule $\mathbf{x}_i^l$ and $\mathbf{x}_{i'}^l$ will be merged if their encoded words $\mathbf{c}_i^l$ and $\mathbf{c}_{i'}^l$ are merged in the layout g^{l+1}.

After forwarding all layers, a tree-structured routing path is generated inside the capsule network and the last layer encodes the entire question-image embedding. We perform global average pooling and linear transformation on the last layer to predict the final answer.

3.1. Linguistic layout generation

We first generate the linguistic layout given the input question Q. We obtain the dependency parse tree by parsing the question with the off-the-shelf universal Stanford Parser [10]. Then, we group the words according to whether they are merged in the parse tree.

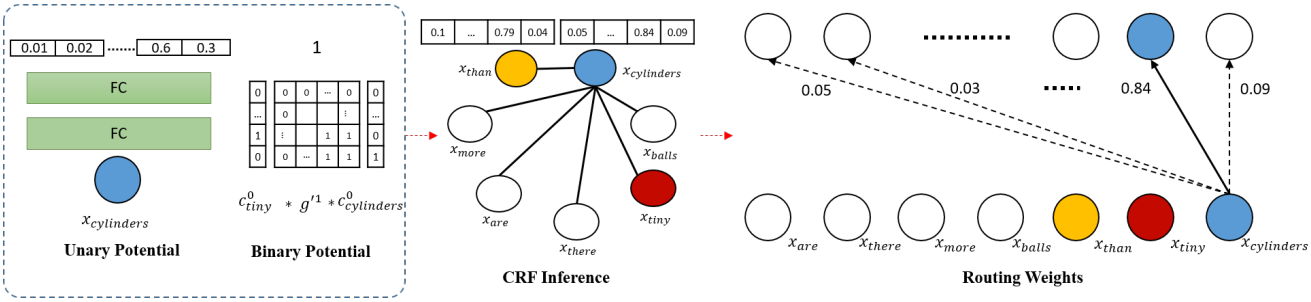

Figure 3: Routing process inside the capsule layer. We first generate the unary potential for each capsule and assign the binary potential for each pair of capsules based on the linguistic layout. Then, we build a fully connected graph and perform the CRF inference to maximize these two potentials across all capsules. The inference results are the routing weights of this layer. As illustrated, the words "tiny" and "cylinders" should be merged together; thus, the binary potential of their edge is 1 if they select the same high-level capsule. The edges that do not connect to "cylinders" are omitted for clarity.

Specifically, we denote a node's level l as the distance between that node and the furthest leaf node. Consider a subtree rooted at node i with level l, we group the words in this subtree into a set and denote it by g_i^l. All groups that are at the same level form a list g^l. For example, the groups at levels 0 and 1 are $g^0 = \{\{are\}, \{there\}, \{more\}, ..., \{tiny\}, \{cylinders\}\}$ and $g^1 = \{\{are\}, \{there\}, \{more\}, ..., \{than, tiny, cylinders\}\}$, as shown in Figure 2. The generated layout $G = \{g^0, g^1, ..., g^H\}$ is used to guide the routing process at different levels of layers, where H is the maximum height of the parse tree.

3.2. Word-level multimodal embedding

We fuse the extracted image feature v and encoded words $\{w_i\}_{i=1:n_q}$ with low-rank bilinear pooling [28] to obtain the multimodal representations $\{\mathbf{x}_i^0\}_{i=1:n_q}$ at layer 0. Specifically, we project the words and the image feature to d-dimensional space, then perform element-wise multiplication to obtain the $\mathbf{x}_i^0$:

$$\mathbf{x}_i^0 = ReLU(FC(w_i) \circ FC(v)). \tag{1}$$

Each multimodal representation contains the information of the image and a single word w_i, thus the $\mathbf{c}_i^0$ is a one-hot vector where $\mathbf{c}_i^0[i] = 1$. The concatenated representation $\mathbf{X}^0$ is the input of the network.

3.3. Linguistically Routing

Given the capsules $\mathbf{X}^l$, their encoded words $\mathbf{C}^l$ and the guided layout g^{l+1}, the routing process generates the routing weights $\{r_{ij}^l\}_{i,j=1:n_c}$ in order to activate several higher-level capsules and compose lower-level capsules with the linguistic guidance. We use the unary potential ψ_i to indicate the probability of activating each higher-level capsule and use the binary potential $\phi_{i,i'}$ that encourage capsule i and capsule i' to select the same capsule if they are merged in the parsed tree. We maximize both potentials with a fully connected CRF. The inference result of the CRF is the routing weights $\mathbf{R}^l$, as shown in Figure 3.

Unary potential The unary potential ψ_i indicates which higher-level capsule should be activated to represent the capsule i. We project the $\mathbf{x}_i^l$ or its global max pooling onto an n_c-dimensional vector with two fully connected layers, where n_c is the number of capsules. We apply softmax to normalize the resulting vector such that each element is between $[0, 1]$.

Binary potential The binary potential $\phi_{i,i'}(j, j)$ is used to encourage capsule i and i' to select the same high-level capsule j if their corresponding words $\mathbf{c}_i^l$ and $\mathbf{c}_{i'}^l$ are merged in the linguistic layout $g^{(l+1)}$. Suppose c_i^l is a set of words merged at capsule i, we consider they are merged if $\exists k$, such that $c_i^l \cup c_{i'}^l \supseteq g_k^{l+1}$.

However, the set operation is not differentiable and thus prevents the entire model from end-to-end training. We hence make $\mathbf{c}_i^l$ a n_q-dimensional vector whose entries represent how much the i-th capsule encodes each question word. For example, at the input layer 0, the capsule i is the fusion between the image and i-th word in the question, and $\mathbf{c}_i^0$ is a one-hot vector where $\mathbf{c}_i^0[i] = 1$. Then, if the softmax-normalized routing weights have $r_{ij}^0 = 0.9$ and $r_{ik}^0 = 0.05$, the encoded degree of the i-th word for capsules j and k are $\mathbf{c}_j^1[i] = 0.9$ and $\mathbf{c}_k^1[i] = 0.05$ respectively. Given the routing weight $\mathbf{R}^l$, we can update the $\mathbf{c}_i^l$ described above and have:

$$\mathbf{C}^{l+1} = \mathbf{R}^{l\top}\mathbf{C}^l. \tag{2}$$

To obtain the binary potential, we firstly transform the guided layout g to an $n_q * n_q$ correlation matrix $\mathbf{g}'$ to represent whether two words i and i' are merged:

$$\mathbf{g}'^l(i, i') = \begin{cases} 1 & \exists a \quad i, i' \in g_a^l \\ -1 & otherwise, \end{cases} \tag{3}$$

where g is the linguistic layout described in Section 3.1.

$\mathbf{g'}^l(i,i') = 1$ indicates that words i and i' are in the same group, e.g., $\{than, tiny, cylinders\}$. Otherwise We set $\mathbf{g'}^l(i,j) = -1$ to prevent two words from early merging. Then, We obtain the binary potential $\phi_{i,i'}$ given the words compatibility matrix $\mathbf{g'}$. Intuitively, the binary potential $\phi_{i,i'}$ should be higher if capsule i and i' contain more compatible words. Thus we have: $\phi_{i,i'} = {\mathbf{c}_{i'}^l}^\top \mathbf{g'}^{l+1} \mathbf{c}_i^l$. For all capsule pairs at layer l, we re-write the above equation in matrix form, and have:

$$\phi = {\mathbf{C}^l}^\top \mathbf{g'}^{l+1} \mathbf{C}^l. \tag{4}$$

Then, we expand each binary potential $\phi_{i,i'}$ into a diagonal $n_c * n_c$ matrix to set the potential as 0 when they select different high-level capsules. Lastly, we construct the CRF with this binary potential and the unary potential to build the CRF, and obtain the routing weights $\mathbf{R}^l$ for all capsules in layer l.

CRF inference The routing weights $\mathbf{R}^l$ should maximize both the unary and binary potentials globally. We construct a conditional random field(CRF) and use the Loopy Belief Propagation to find the optimised routing weight. Specifically, we construct a CRF where each node represents a capsule, and node i's n_c-dimensional random variables $\{z_i\}$ correspond to the routing weights. Given the unary ψ_i and binary potentials $\phi_{i,i'}$ described above, we initialize the message $m_{ii'}^0(z_i)[i'] = 1/n_c$ as uniform distribution, and update the message with the following formula:

$$m_{i\to i'}^t(z_{i'}) = \sum_{z_i} \phi_{i,i'}(z_i, z_{i'}) \psi_i(z_i) \prod_{k \supseteq N_i \backslash i'} m_{k\to i}^{t-1}(z_i). \tag{5}$$

where $N_i \backslash i'$ is the neighbors of node i except the node i'. After T iteration, we gather the message for all node and variable and obtain the marginal probability:

$$b_i(z_i) = \frac{1}{Z_b} \psi_i(z_i) \prod_{k \supseteq N_i} m_{k\to i}^T(z_i), \tag{6}$$

where Z_b is the normalizing factor. The resulting marginal probability is the corresponding routing weights $r_{ij}^l = b_i(z_i)[j]$. We implement the above Loopy Belief Propagation process as a non-parametric layer such that it can back-propagate the gradient.

3.4. Capsule layer

In a general neural network, the forward propagation has $x_j = \sigma(\sum_i \mathbf{W}_{ij} x_i)$, where x_i and x_j are the neurons in consecutive layers and σ is a activation function. After grouping a set of neurons into n_c capsules, the linguistically routing weights $\{r_{ij}^l\}_{i,j=1:n_c}$ are applied to reweight the linear transformation from capsule i to capsule j. Formally,

$$\mathbf{x}_j^{l+1} = \sigma(\sum_i r_{ij}^l \mathbf{W}_{ij} \mathbf{x}_i^l), \tag{7}$$

where $\mathbf{x}_i^l$ is the i-th capsule in layer l and $\mathbf{x}_j^{l+1}$ is the j-th capsule in the next layer $l+1$. Routing weight r_{ij}^l is a number between $[0,1]$ and have $\sum_j r_{ij}^l = 1$.

For the convolution layer, the same convolution operation on spatial dimension, and apply the routing weights on the feature channel.

$$\mathbf{x}_{w,h,j}^{l+1} = \sigma(\sum_i r_{ij}^l \sum_a \sum_b \mathbf{W}_{ij} \mathbf{x}_{w+a,h+b,i}^l), \tag{8}$$

where w and h are the spatial location in the feature map. We apply the convolution operation on each capsule i to obtain n_c feature maps $\hat{\mathbf{x}}_{ij}^l$. We also apply global max pooling and two fully-connected layers on each capsule i to predict the unary potential and obtain the routing weights r_{ij}^l. Lastly, we obtain capsule j in the next layer by summing the weighted feature maps $\sum_i r_{ij}^l \hat{\mathbf{x}}_{ij}^l$.

All the above operations are differentiable. Thus, the proposed linguistically routing can be end-to-end trained along with other network parameters. During training, We only use the answer label as supervision signal and train the whole capsule network in an end-to-end manner.

4. Experiment

In this section, we validate the effectiveness and generalization capability of our method on the CLEVR composition generalization test, and the VQA-CP v2 dataset. We also evaluate our proposed method on the VQAv2 dataset to verify its performance on in-domain test data.

4.1. Datasets

The **CLEVR composition generalization test (CLEVR-CoGenT)** [25] is proposed to investigate the composition generalization ability of a VQA model. This dataset contains $130,000$ images and $1,299,923$ questions. The images are rendered with objects of random shapes, colors, materials, and sizes. And the questions are synthesized based on functional program layouts. Its validation split has two conditions: in condition A, all cubes are gray, blue, brown, or yellow, and all cylinders are red, green, purple, or cyan. In condition B, cubes and cylinders swap color palettes. Thus, the test samples are out of training distribution. A model cannot achieve good performance on condition B by simply memorizing and overfitting the samples in condition A.

Visual Question Answering under Changing Priors (VQA-CP) v2 dataset [1] is constructed by re-organizing the train and validation splits of the VQA v2 dataset, such that the training and testing answer have different distributions. The VQA-CP v2 has been one of the most popular benchmarks for the out-of-distribution VQA task.

VQAv2 [17] is the most popular VQA benchmark. Its training split contains $82,783$ images and $443,757$

Model	A	B
IEP [26]	96.6	73.7
NS-VQA [46]	99.8	63.9
NS-VQA+Ori [46]	99.8	**99.7**
SA [26]	80.3	68.7
MAC [22]	97.66	74.75
PTGRN [9]	97.35	83.50
FiLM [36]	98.3	75.6
FiLM 0-Shot [36]	98.3	78.8
TbD+reg [33]	98.8	75.4
LR-Capsule(ours)	98.1	<u>85.6</u>

Table 1: Answering accuracy on the CLEVR-CoGenT validation set. Each method is trained on condition A only and is evaluated on both conditions A and B.

	VQA-CP v2 Test			
Method	All	Yes/no	Number	Other
AReg [37]	41.17	65.49	15.48	35.48
MuRel [7]	39.43	42.85	13.17	45.04
ReGAT [31]	40.42	-	-	-
NSM [23]	45.80	-	-	-
RUBi [8]	47.11	68.65	20.28	43.18
RUBi+UpDn [8]	44.23	67.05	17.48	39.64
SCR [43]	48.47	70.41	10.42	47.29
LMH [13]	**52.45**	69.81	44.46	45.54
LR-Capsule(ours)	<u>52.19</u>	76.44	28.37	46.02

Table 2: Question answering accuracy on the VQA-CP v2 test split.

questions; the validation split contains 40504 images and $214,354$ questions; and its test split contains $81,434$ images and $447,793$ questions. Each question has 10 human-annotated answers.

4.2. Implementation details

To verify the effectiveness of our proposed linguistically routing, we use two state-of-the-art methods, FiLM [36] and MCAN [47], as backbone architecture and replaced their convolution or fully connected layer with capsule layer respectively.

For the CLEVR-CoGenT datasets, we follow FiLM [36] to extract the image feature and word embedding. We resize the images to 224×224, and extract $14 \times 14 \times 1024$ feature v from conv4 of the ResNet-101 [18] that was pretrained on ImageNet. The 1024-dimensional feature maps are concatenated with a 2-channel coordinate map and are projected onto a 128-dimensional space using a single 3×3 convolutional layer. The word embedding vector w_i for the question is obtained via the gated recurrent network (GRU) [12]. We first embed the word into a 200-dimensional vector and

Method	All	Yes/no	Number	Other
AReg [37]	62.75	79.84	42.35	55.16
ReGAT [31]	67.18	-	-	-
RUBi [8]	61.16	-	-	-
RUBi+UpDn [8]	50.56	49.45	41.02	53.95
SCR [43]	62.30	77.40	40.90	56.50
LMH [13]	61.64	77.85	40.03	55.04
LMH-CSS [11]	59.91	73.25	39.77	55.11
MCAN [47] baseline	**67.2**	84.8	49.3	58.6
LR-Capsule(ours)	<u>67.04</u>	84.57	48.66	58.57

Table 3: Question answering accuracy on the VQA v2 validation split.

then feed the entire question into a 512-dimensional bi-GRU. The word embedding $\{w_i\}_{i=1:n_q}$ are the hidden vectors of the GRU at their corresponding position. Then, we perform the word-level multimodal embedding on the image feature v and the word embedding $\{w_i\}_{i=1:n_q}$. The resulting multimodal representations $\mathbf{X}^0 = \{\mathbf{x}_i^0\}_{i=1:n_q}$ is the lowest feature map of the neural network, where the $n_q = 46$ is the maximum length of questions in CLEVR-CoGenT datasets. Each capsule $\mathbf{x}_i^0$ is a $14 \times 14 \times 128$ feature map, and its encoded words $\mathbf{c}_i^0$ is a one-hot vector where $\mathbf{c}_i^0[i] = 1$. Since the maximum number of the level-1 nodes in the parsed tree is 9, we set the capsule number as 9; each capsule has 16 feature channels. The heights of parse trees in CLEVR-CoGenT are mostly less than 4. Thus, we keep the top-4 levels of the parse tree and set the number of convolutional capsule layers to 4. During linguistically routing, each $14 \times 14 \times 16$ capsule is fed into a global max pooling layer, two fully connected layers with output sizes of 512 and 9, where 9 is the number of capsules in the next layer. Given the binary potential and the 9-dimensional unary potential, we perform the loopy belief propagation 2 iterations to obtain the routing weights r. Each capsule layer has $3 \times 3 \times 144$ convolution kernel, followed by a batch normalization, a multiplicative fusion with the transformed question embedding, a ReLU activation, and a residual connection. Lastly, the classifier convolves the 144-dimensional feature maps to 512 dimensions and feeds the result into two fully connected layers with output sizes of 1024 and 29, where 29 is the number of candidate answers.

For the VQA-CP v2 and VQAv2 dataset, we modified the Modular Co-Attention Networks (MCAN) [47] and introduce the linguistically routing in the guided-attention blocks. Similar to MCAN, the words are embedded by an LSTM and 6 self-attention blocks, resulting in 512-dimensional word embedding vectors. The image feature is extracted by the bottom-up top-down model [2]. Each image has 36 objects with 2048-dimensional feature vectors. We split each object feature into 16 capsules with 32-

Binary Potential	CLEVR-CoGenT A	CLEVR-CoGenT B	VQA-CP v2 Test
Baseline	97.59	78.19	51.15
0	98.00	82.17	51.62
0.5	97.71	82.25	51.68
1	**98.10**	**85.58**	**52.18**
2	96.27	79.16	51.73

Table 4: The performance of different binary potentials on CLEVR-CoGenT and VQA-CP v2.

	CLEVR-CoGenT A	CLEVR-CoGenT B	VQA-CP v2 Test
Baseline	97.59	78.19	51.15
+scale	97.89	81.73	51.74
+mul	97.32	78.01	51.15
+unary	98.00	82.17	51.62
+unary+binary	93.61	78.29	51.63
+unary+binary+parser	**98.10**	**85.58**	**52.18**

Table 5: The performance of different reweighting schemes on CLEVR-CoGenT and VQA-CP v2.

dimension. Then we project the words' embedding to 32-dimensional vector and fuse them with the 32-dimensional visual feature. Thus, the capsule number is 16 and each capsule contains 32 neurons. The image is first passed through 3 guided-attention blocks. Then we only replace the feed-forward layer in the last 3 guided-attention blocks with the linguistically routing capsule layer. We perform linguistically routing for each object individually. For each object, their 32-dimensional capsules are fed into two fully connected layers with output sizes of 32 and 16 to predict the unary potential. We also perform 2 iterations of loopy belief propagation to obtain the routing weights r. The classifier is the same as the MCAN [47]. It performs attention on question words and 36 image objects, then obtains a 1024-dimensional vector. The classifier project the 1024-dimensional vector to 3129-dimension, where the number of the answer candidates is 3129.

To reduce the computational complexity for CLEVR-CoGenT, we prune the leaf nodes that are neither nouns nor words denoting colors. The model is trained with Adam optimizer [29]. The base learning rate is $3e^{-4}$ for the CLEVR-CoGenT and is $1e^{-4}$ for the VQA-CP v2 and VQAv2, respectively. The batch size are 64 and 256 respectively. The weight decay, β_1 and β_2 are $1e^{-5}$, 0.9, and 0.999.

4.3. Comparison with state-of-the-art methods

CLEVR compositional generalization test We report the answering accuracy of different models on CLEVR-CoGenT in Table 1. The accuracy is obtained by training the models in Condition A, and evaluating on both Condition A and Condition B without fine-tuning. As shown in Table 1, while achieving a comparable accuracy in Condition A, our proposed linguistically routing significantly outperforms all the compared methods except NS-VQA+Ori [46] in Condition B. Note that NS-VQA+Ori requires both scene graph and a question's functional layout as additional supervised signals. Without additional training signals, its accuracy downgrades to 63.9%. This verifies the effectiveness of our model in terms of the composition generalization ability.

VQA-CP v2 dataset We report the standard VQA evaluation metric [5] in Table 2. We combine our method with RUBi [8] and obtain 52.19%. This result improves the original RUBi [8] by 5.08% and is the best performance among single-model-based methods. This result is also close to 52.45%, which is obtained by the ensemble-based method LMH [13]. The experiments have shown the effectiveness of our method and its potential of combining other works for better performance.

VQAv2 dataset Table 3 gives the results on VQAv2 validation set. The MCAN baseline surpasses all compared methods in terms of answering accuracy. Compared with this strong baseline, our method can achieve similar results on in-domain test data while achieving superior performance on the VQA-CP v2 dataset. Our method also surpasses the best VQA-CP v2 method LMH [13] by a large margin on the VQAv2 dataset. The experimental results demonstrate that our model can improve generalization ability while not losing the performance on in-domain test data.

4.4. Ablation Studies

We evaluate the effectiveness of our proposed linguistically routing on the CLEVR-CoGenT and the VQA-CP v2 by changing the binary potential $\phi_{i,i'}$. The results are shown in Table 4. The "Baseline" model is a regular network that has the same architecture and word-level multimodal embedding as our main model but without routing. The next row is a baseline result that only includes multiplicative unary potential by setting the binary potential $\phi_{i,i'}$ to 0. In the following rows, the linguistic constraint becomes stricter as we increase the binary potential. The performance increases as the binary potential increases from 0 to 1, but drops when the potential becomes larger than 1. Since we normalize the unary potential between $[0, 1]$, we assume smaller binary potential can't introduce the linguistic constraint strictly enough. But if the potential becomes too large, the constraint will prevent the routing process select the proper capsules and lead to the decrease of model capacity.

We show the performance of different model variants on Table 5tTo further inspect the influence of multiplicative interaction. We remove the softmax normaliza-

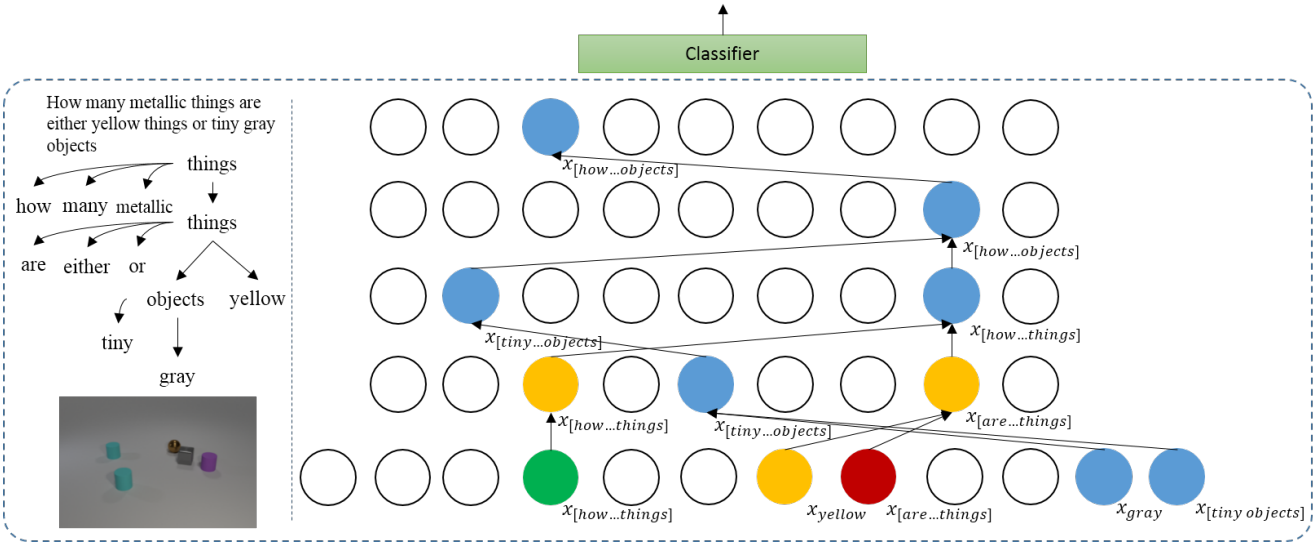

Figure 4: Visualization example of our routing result on the CLEVR-CoGenT. We display all capsules and all layers but omit the padding words. Thus, the number of capsules displayed at layer 0 is equal to the question length. The curved edge in the parse tree indicates the pruned leaf nodes, and the blue circles indicate the capsule that should be merged at the next layer.

tion("+scale") applied on the unary potential and generate the elementwise multiplicative vector instead of a single unary value("+mul"). The "+unary+binary" indicates a model that learns the binary potential based on the answer classification loss only. We concatenate the features of every capsule pairs and use two fully connected layers to predict the $n_c * n_c$ matrix described in Section 3.3. Table 5 shows that different model variants achieve similar accuracy on Condition A. However, the accuracy on out-of-distribution samples varies considerably. The unary routing improves the accuracy by 3.98% and 0.47% compared with the baseline models on the CLEVR-CoGenT Condition B. Our full model "+unary+binary+parser" also achieves better results than "+scale" and "+mul", which demonstrates the effectiveness of the linguistic guidance.

4.5. Visualization of routing results

We visualize our routing result in Figure 4. The input questions, image, and linguistic guidance are shown on the left, while the routing results are shown on the right. The example firstly combines the terms "gray" and "objects", same with the parse tree. However, it combines the "yellow objects" with "how many" in the third layer and encodes the "yellow objects" and the "gray objects" separately. It then combines them to predict the answer at last. The example follows the linguistic guidance at first but demonstrates a more reasonable routing process than the parse tree to answer the question. Due to the limited page space, more examples are provided in the supplementary file.

5. Conclusion

We propose the Linguistically Routing that can incorporate the linguistic information in an end-to-end manner to improve the capsule network's generalization capability on OOD data. We use the unary potential for each capsule to activate a proper high-level capsule, and use the binary potential for capsule pairs to incorporate the linguistic structures. A CRF is applied to maximize two types of potential. As we bind the lowest visual feature with a single word, the bottom-up linguistic-guided merging process can combine the words into phrases, clauses, and finally a sentence. After forwarding all layers, the parse tree is carved inside the network and entangled with visual patterns. In the future, we will progressively refine our model to further improve its generalization ability and broaden its application domain.

6. Acknowledgement

This work was supported in part by National Key R&D Program of China under Grant No. 2020AAA0109700, National Natural Science Foundation of China (NSFC) under Grant No. U19A2073, 61976233, 62006255 and 61876045, Guangdong Province Basic and Applied Basic Research (Regional Joint Fund-Key) Grant No.2019B1515120039, Guangdong Outstanding Youth Fund (Grant No. 2021B1515020061), Shenzhen Fundamental Research Program (Project No. RCYX20200714114642083, No. JCYJ20190807154211365), Zhejiang Lab's Open Fund (No. 2020AA3AB14) and CSIG Young Fellow Support Fund.

References

[1] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *CVPR*, 2018.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

[3] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. In *NAACL*, 2016.

[4] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *CVPR*, 2016.

[5] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *ICCV*, 2015.

[6] Hedi Ben-younes, Remi Cadene, Matthieu Cord, and Nicolas Thome. Mutan: Multimodal tucker fusion for visual question answering. In *ICCV*, 2017.

[7] Remi Cadene, Hedi Ben-Younes, Matthieu Cord, and Nicolas Thome. Murel: Multimodal relational reasoning for visual question answering. In *CVPR*, 2019.

[8] Remi Cadene, Corentin Dancette, Hedi Ben younes, Matthieu Cord, and Devi Parikh. Rubi: Reducing unimodal biases for visual question answering. In *NIPS*, 2019.

[9] Q. Cao, X. Liang, B. Li, and L. Lin. Interpretable visual question answering by reasoning on dependency trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019.

[10] Danqi Chen and Christopher D Manning. A fast and accurate dependency parser using neural networks. In *EMNLP*, 2014.

[11] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering, 2020.

[12] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[13] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *EMNLP*, 2019.

[14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, Sept 2010.

[15] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, 2016.

[16] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. Iqa: Visual question answering in interactive environments. In *CVPR*, 2018.

[17] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *CVPR*, 2017.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016.

[19] Geoffrey Hinton. How to represent part-whole hierarchies in a neural network, 2021.

[20] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with EM routing. In *ICLR*, 2018.

[21] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *ICCV*, 2017.

[22] Drew A Hudson and Christopher D Manning. Compositional attention networks for machine reasoning. 2018.

[23] Drew A. Hudson and Christopher D. Manning. Learning by abstraction: The neural state machine. In *NIPS*, 2019.

[24] Allan Jabri, Armand Joulin, and Laurens van der Maaten. Revisiting visual question answering baselines. In *ECCV*, 2016.

[25] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

[26] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Fei-Fei Li, C. Lawrence Zitnick, and Ross B. Girshick. Inferring and executing programs for visual reasoning. In *ICCV*, 2017.

[27] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. Dvqa: Understanding data visualizations via question answering. In *CVPR*, 2018.

[28] Jin-Hwa Kim, Kyoung Woon On, Jeonghee Kim, JungWoo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling. In *ICLR*, 2017.

[29] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[30] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[31] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering. *ICCV*, 2019.

[32] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *NIPS*, 2016.

[33] David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *CVPR*, 2018.

[34] Medhini Narasimhan, Svetlana Lazebnik, and Alexander G Schwing. Out of the box: Reasoning with graph convolution nets for factual visual question answering. In *NIPS*, 2018.

[35] Will Norcliffe-Brown, Efstathios Vafeias, and Sarah Parisot. Learning conditioned graph structures for interpretable visual question answering. In *NIPS*, 2018.

[36] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

[37] Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. Overcoming language priors in visual question answering with adversarial regularization. In *NIPS*, 2018.

[38] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *NIPS*, 2017.

[39] Kevin J. Shih, Saurabh Singh, and Derek Hoiem. Where to look: Focus regions for visual question answering. In *CVPR*, 2016.

[40] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *ICLR*, 2020.

[41] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019.

[42] Damien Teney, Lingqiao Liu, and Anton van den Hengel. Graph-structured representations for visual question answering. In *CVPR*, 2017.

[43] Jialin Wu and Raymond Mooney. Self-critical reasoning for robust visual question answering. In *NIPS*, 2019.

[44] Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*, 2016.

[45] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016.

[46] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B. Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *NIPS*, 2018.

[47] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *CVPR*, 2019.

[48] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *ICCV*, 2017.

[49] Chen Zhu, Yanpeng Zhao, Shuaiyi Huang, Kewei Tu, and Yi Ma. Structured attentions for visual question answering. In *ICCV*, 2017.

[50] Yuke Zhu, Oliver Groth, Michael S. Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *CVPR*, 2016.