TRRNet: Tiered Relation Reasoning for Compositional Visual Question Answering

Xiaofeng Yang[1], Guosheng Lin[1⋆], Fengmao Lv[2], and Fayao Liu[3]

[1] Nanyang Technological University
[2] Center of Statistical Research, Southwestern University of Finance and Economics
[3] Institute for Infocomm Research A*STAR
xiaofeng001@e.ntu.edu.sg
gslin@ntu.edu.sg

Abstract. Compositional visual question answering requires reasoning over both semantic and geometry object relations. We propose a novel tiered reasoning method that dynamically selects object level candidates based on language representations and generates robust pairwise relations within the selected candidate objects. The proposed tiered relation reasoning method can be compatible with the majority of the existing visual reasoning frameworks, leading to significant performance improvement with very little extra computational cost. Moreover, we propose a policy network that decides the appropriate reasoning steps based on question complexity and current reasoning status. In experiments, our model achieves state-of-the-art performance on two VQA datasets.

Keywords: Visual Question Answering, Visual Reasoning

1 Introduction

Visual Question Answering [3,10,15,17] is the task of answering a natural language question based on the content of an image. To precisely answer visual questions, the VQA models should be able to understand the language, the image and build a cross-modal mapping between the lingual and visual contents.

Current state-of-the-art VQA methods can be divided into two categories. The works from the first category [1,24,19] mainly focus on learning a multi-modal joint representation of language and vision. The visual features, which are usually extracted from pre-trained networks, are combined with the language features through multiple self-attention and co-attention [21,7] modules. Although these methods are proven to work well on VQA [3] tasks, they usually don't generalize well on the compositional reasoning tasks [15] due to their lack of relation reasoning abilities. For example, given a question of "*What is the color of the food on the plate to the right of the girl?* ", the VQA model needs to first understand the mapping between language representation and visual features, and then reason over the relations between each pair of objects to decide the

⋆ Corresponding author: G. Lin.

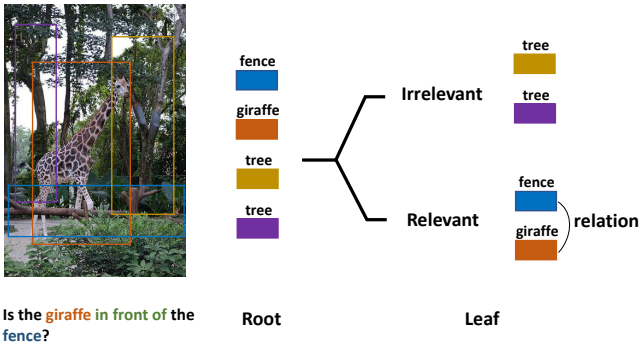

Fig. 1. The idea of tiered relation reasoning. In roots, there are many possible objects for answering questions. Based on the attention map of root, the objects can be classified as relevant or irrelevant objects. In leaf, pairwise relations are only generated between relevant objects.

final answer. Hence, methods based on simple self-attention and co-attention are insufficient in performing relation reasoning. On the other hand, the methods from the second category mainly focus on designing neural modules [2,23] that can perform more diverse reasoning tasks. These methods perform extremely well on simulated datasets like CLEVR [17]. However, the design of reasoning modules is tricky and heavily relies on human efforts. Hence, they are not widely adopted in real-world datasets [3,15] that contain far more object classes and possible reasoning actions.

The idea of reasoning over relations has been drawn in some of the previous works. In [26] the relations are generated for each pair of regions and then combined with language representations. In [4] dense pairwise relations are generated for multi-modal embeddings. Other works [34] also propose to generate relations based on geometry information e.g. generate relations for geometry close objects only. The dense object relations are usually noisy and computation consuming. Imagine the case that there are one hundred visual objects. There will be ten thousand possible pairwise relations. In fact, one sentence will not cover more than six objects. It brings in much difficulty to find relevant ones from all ten thousand relations. The geometry sparse relations are built on a strong assumption that relations are only valid for neighbor objects, which is not always true.

In this work, we propose a tiered relation reasoning method that dynamically selects object level candidates based on language representation and generates tidy object relations within the selected candidates objects only. The basic idea of tiered relation reasoning is illustrated in Fig. 1. We denote our structure as a tiered structure, because the reasoning is from coarse objects to fine candidate objects. In root, we have many objects grouped, and in leaf, the objects are split into relevant and irrelevant objects. The irrelevant objects could be elimi-

nated from further processing. The tiered selection not only makes the network computational efficient but also improves relation reasoning performance.

Our proposed TRRNet(Tiered Relation Reasoning Network) consists of a series of TRR units. For each TRR unit, there are four basic components: root attention, root to leaf attention passing, leaf attention and a final message passing module to interact with the next TRR unit. The root attention is an object level attention attending to different visual features based on language representation. The functionality of this component could be achieved by many modern approaches [1,19,7]. After root attention, the network carries out root to leaf attention passing that selects the most significant visual components and builds pairwise relations. This is achieved by multi-head hard attentions. After that, the leaf attention module learns to reason on relations based on language features. Finally, the information from relation reasoning is passed to the next stage of reasoning through a message passing module. The mappings of language-objects, language-relations are purely supervised by final training loss, without seeking additional strong supervisions such as scene graph annotations [31] and functional programs.

Natural language questions are various and often require multiple reasoning steps [27]. Short and simple questions can be solved easily with one or two reasoning steps. For long and complex questions, it may take more steps to solve them. The proposed TRRNet is able to generate a series of reasoning outputs. On top of that, we design a policy network that decides the appropriate reasoning steps based on questions and current reasoning outputs. The policy network not only boosts final accuracy, but also improves overall processing speed.

In summary, our contributions are:

- We propose a novel tiered attention network for relation reasoning. The TRR network consists of a series of TRR units. Each TRR unit can be decomposed into four basic components: a root attention to model object level importance, a root to leaf attention passing module to select candidate objects based on root attention and generate pairwise relations, a leaf attention to model relation level importance and finally a message passing module for information communication between reasoning units.
- We propose a policy network that chooses the best reasoning steps based on natural language question and reasoning outputs.
- We achieve state-of-the-art performance on GQA dataset and competitive results on CLEVR datasets and VQAv2 dataset without functional program supervision.

2 Related Works

In this section, we categorize VQA tasks that do not require compositional reasoning skills as visual question answering [3] and VQA tasks that require multiple reasoning skills as visual reasoning [17,15].

2.1 Visual Question Answering

The improvement in visual question answering has mainly been done on two parts, namely better features and better attentions. Early VQA methods use pre-trained VGG or ResNet to extract visual features. The bottom-up and top-down network [1] proposed to extract visual features from an object detector [9]. The bottom-up features significantly improve VQA performance.

Better attentions also contribute a lot to improve VQA performance. Works in [20,37,32] further extended traditional attention methods for better visual groundings. Co-attention [29,21,36,24] performs attention based on a fused representation on image features and language features respectively. BAN [19] network further improved computation efficiency of co-attention through bilinear attention. Motivated by NLP works [28], self-attention is also widely adopted to improve VQA performance [7,35]. In these works, self-attention is used to generate intra-modality attention maps and a summation of the original features.

It is worth noticing that conventional VQA methods lack the ability of relation reasoning. Co-attention mechanism models correspondence between words and image regions. Relations between objects are largely missing. Self-attention models intra-modality importance, while it could not cover both semantic and spatial relations. Compared with transformer based methods which encode relations implicitly, our model encodes explicit relations directly, making it easier for the models to perform reasoning.

2.2 Visual Reasoning in VQA

Early visual reasoning works [26] generate dense relations for each pair of pixels. The dense relations are then combined with language features for language guided reasoning. State of the art reasoning methods are dominated by neural module networks [2]. In neural module networks, language representation is parsed into a series of logical steps. For each logical step, a different neural module for performance corresponding actions is designed. In [18,12,23], the questions are parsed into functional programs and specific executing engines are designed for the functional programs. The neural module approach [33] could achieve almost perfect performance on simulated datasets [17]. However, for real-world datasets [3,15], it's almost impossible to design specific modules for each type of reasoning due to the questions' high complexity.

In this work, we propose our TRRNet for relation reasoning. The model is purely trained with answer supervision. Strong functional program supervision is not used for training.

3 Our Approach

3.1 Overview

The task of VQA is described as follows: Given an image I and a question Q grounded on I, the purpose of VQA is to select the best answer from a set of all possible answers. The VQA task is usually defined as a classification problem.

Fig. 2 shows a detailed illustration of our method. Following the standard VQA practice, the image is first processed with an object detector to extract n region features $V \in \mathbb{R}^{n \times d_v}$ and n bounding boxes $B \in \mathbb{R}^{n \times d_b}$, where the ith regional feature and box feature is denoted as $v_i \in \mathbb{R}^{d_v}$ and $b_i \in \mathbb{R}^{d_b}$. For language processing, we adopt Bert [6] word embeddings and input the embedding vector to a GRU [5] to better adapt the embeddings to VQA task. This step gets us a set of language embedding features $E \in \mathbb{R}^{m \times d_e}$ where m represents the number of words in one sentence. The proposed approach TRRNet consists of a series of TRR units. One TRR unit can be decomposed into four basic components, namely root attention (object level attention), root to leaf attention passing, leaf attention (relation level attention) and message passing module to pass message to the next TRR unit. Root attention is an object level attention, which learns attention weights for objects and generates a merged feature. Root to leaf attention passing module processes the object level attention map and produce pairwise relations. Based on the pairwise relations and language, leaf attention attends to different relations. Message passing module summarizes relation level attention maps and combines them with object level features, which will then be propagated to the next stage.

3.2 Root Attention

Given a set of image features $V \in \mathbb{R}^{n \times d_v}$ and bounding box features $B \in \mathbb{R}^{n \times d_b}$ extracted from Faster R-CNN and a set of question features from GRU $E \in \mathbb{R}^{m \times d_e}$, the *root attention* has two roles.

The first role is to generate an attention map for object level visual features based on language representations:

$$\alpha^{object} = softmax(Net(V, B, E)), \tag{1}$$

where "Net" denotes any possible networks that can generate object level attentions.

The second role of root attention is to generate a fused visual feature:

$$O^{root} = \alpha^{object} V^T, \tag{2}$$

where O^{root} denotes the output on root stage.

There are multiple choices for root attentions. Actually, most modern VQA methods that contain object level attention can be used as our root attention, such as Bottom-Up [1], DCN [24], BAN [19], Inter-intra [7]. In this work, we try both a simple attention method as used in Bottom-Up Attention [1] and a more complicated and advanced method as proposed in BAN [19].

This setting guarantees the flexibility of our work. Besides using one network as root attention, it's even possible to use several networks as the root attention and average the final attention maps to form a more robust object level features.

3.3 Root to Leaf Attention Passing

The *root to leaf attention passing* further processes the object level attention map from the root attention to produce pairwise relations. Note that the object level attention map might have multi-heads. We use multi-head hard attention to select relevant object candidates.

The idea of hard attention was first proposed in [30], where only the most related areas are selected for image captioning. When the image features are selected in a "hard" way, the network becomes undifferentiable. In [30] REINFORCE is used to find network parameters through sampling. In [22], hard attention is used to select grid features to answer visual questions. In this paper, we can format the hard attention problem in an easier and fully differentiable way.

For multi-head hard attention passing, we only select "hard" objects that are related to questions. Specifically, based on the attention maps from root attention, we select only the top k related objects based on the attention weights. After the visual features V_{Hard} are selected, we build pairwise relations by concatenating the features one-by-one and process them through a MLP to map them to size d_r. In experiment, we use $d_r = 256$, a value smaller than original visual feature dimension, because the relations usually contain less information than the objects and using smaller feature size can help to reduce overfitting. This step is represented as *Relation* in Equation 4. Formally, for each attention head in root attention α^{object}, we perform:

$$V_{Hard} = Topk(\alpha, V, K), \tag{3}$$

$$R = Relation(V_{Hard}, B), \tag{4}$$

where K is a hyper-parameter for controlling how many objects for generating relations, B is the box features and R is the generated pairwise relations.

The refined object candidates also motivate us to generate triple-wise relations. For triple-wise relations, a set of relations $R_{triple} \in \mathbb{R}^{K^3 \times d_r}$ is generated for candidate objects. Ideally, the triple-wise relations can help the network to better understand long sentences with multiple objects. We show detailed ablation studies in Sec. 4.2.

3.4 Leaf Attention

The *leaf attention* performs reasoning over object relations. Same as root attention, the leaf attention also produces both an attention map and a feature merged by relation features and language features. Given the relation features $R \in \mathbb{R}^{K^2 \times d_r}$ and the final question embedding $e \in \mathbb{R}^{d_e}$, we first obtain a hidden feature by fusing language and visual features and then the fused features are further propagated through non-linear functions:

$$h = f(g(e) \cdot k(R)), \tag{5}$$

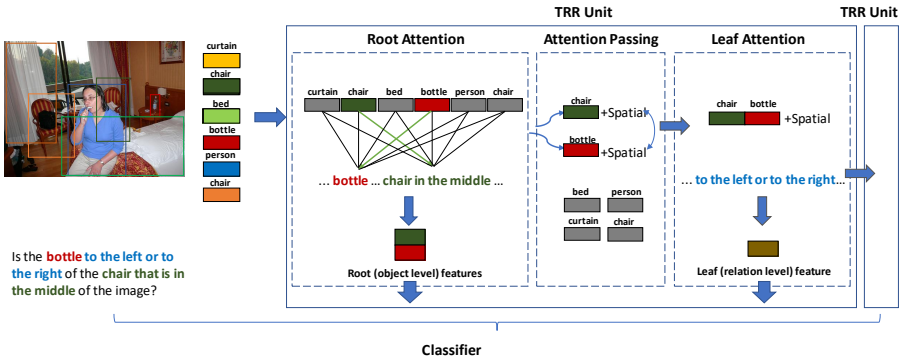

Fig. 2. A detailed illustration of a one layer TRR network with hard attention. Given an image and a natural question, the image is first processed with Faster R-CNN for feature extraction. In the root attention module, the object features are merged with language and an attention map of objects is generated. The root attention also produces a merged object level feature. The attention passing module processes the root attention map to select the most significant objects based on the attention map. Then pairwise relations are constructed and combined with spatial features (bounding boxes). The leaf attention performs attention on relations and generates relation level features. Root features and leaf features are finally combined with language features for predicting the final answer.

where g, k, and f are fully connected layers with activation function ReLU and "." operation represents element-wise multiplication. The h value is now regarded as a mapping correspondence. Based on the correspondence, an attention map can be calculated from h for each relation:

$$\alpha^{relation} = softmax(h). \tag{6}$$

Finally, a merged relation feature is generated by matrix multiplication:

$$O^{leaf} = \alpha^{relation} R^T. \tag{7}$$

If R is a multi-head feature, the final leaf feature O^{leaf} is a concatenation of all heads.

3.5 Message passing module for units interaction

To enable multi-stage reasoning, we propose a *message passing module*. In this module, we fuse the attended relation features O^{leaf} and object level features V through concatenation and a fully connected layer. The new visual feature is generated by:

$$V_{new} = f([O^{leaf}, V]), \tag{8}$$

where f is a linear function with activation.

3.6 Multi-stage Reasoning and Policy Network

Compositional natural language questions require multi-stage reasoning. Easy questions can be solved within two reasoning steps, while long and complex questions may require more reasoning steps. The proposed TRRNet is able to generate a series of reasoning outputs. For the tth TRR unit, the network takes the bounding box feature B, the tth visual features V_t and the question embedding E as input and output the root feature O_t^{root}, the leaf feature O_t^{leaf}, and the aggregated visual feature V_{t+1}:

$$O_t^{root}, O_t^{leaf}, V_{t+1} = TRR_t(B, V_t, E). \tag{9}$$

At the tth TRR unit, the policy network decides whether to proceed to the next stage of reasoning. The design of policy network follows two assumptions. First, complicated questions should be reasoned for more steps and second if the attended features from the previous steps are already stable, the network should stop.

We format the reasoning process as a sequential decision making process. At time step t, the "State" s_t is the question embedding E, question length l, current time step t and root feature O_{t-1}^{root} and O_t^{root}. The reason why we choose the root level features is because leaf features are calculated based on root, so root features itself can represent the reasoning status. The "Action" a_t is a binary action [1,0] where "1" denotes to proceed and "0" denotes not to proceed to the next reasoning step. The "Reward" r_t is the number of correct VQA predictions of current batch. In order to minimize the total number of reasoning steps, we also introduce a small scalar penalty term $p_t = 0.1$ for each time the policy network chooses to proceed. The "Policy" is a function $\pi(a_t|s_t, \theta)$. The policy function decides the next action based on State s_t. The decision process is regarded as a Markov Decision Process and can be trained with reinforcement learning settings.

To be more specific about the policy network structure, we first calculate the L2 distance of root features from the previous two time steps: $d = L2(O_{t-1}^{root}, O_t^{root})$. The distance and question embedding are then processed by two separate mlps for final prediction:

$$d = L2(O_{t-1}^{root}, O_t^{root}), \tag{10}$$

$$z = MLP[MLP(d), MLP(E, t, l)], \tag{11}$$

$$\pi(a_t|s_t, \theta) = softmax(z). \tag{12}$$

The policy network is trained with policy gradient method REINFORCE. During training, we alternatively train the main network and the policy network. We first train the main network and then fix the parameters of main network and train the policy network. After that, we go back to train the main network again. We set the maximum number of reasoning steps to be $N = 3$. Since at

time step $t = 1$, the O_0^{root} value is $None$, the policy network is only used after $t = 2$. The loss of policy network is defined as: $L = -E_{s\sim\pi}[r - p]$.

3.7 The Readout Layer

In readout layer, the final stage root features O^{root} and leaf features O^{leaf} are further combined and processed with a linear function to represent the final visual features:

$$O^{all} = f([O_t^{root}, O_t^{leaf}]). \tag{13}$$

Question embedding features are processed through a fully connected layer with an activation function ReLU to represent the final language features:

$$E^{final} = g(E). \tag{14}$$

Finally, the question feature and visual features are combined via an element-wise product for classification:

$$Answer = softmax(h(O^{all} \cdot E^{final})), \tag{15}$$

where h is a fully connected layer with an activation function.

4 Experiments

4.1 Experimental Setup

We use GQA [15] dataset for our experiments. The GQA dataset contains 22M compositional questions and 140K images. Compared to VQA [3], GQA dataset contains questions that require multiple reasoning skills. In our experiment, we use pre-trained Faster-RCNN in [1] to extract region features of size 36×2048. We use pre-trained Bert word embedding and GRU to extract language features of size 20×1024, where 20 is the length of questions. There are two training and testing splits in GQA, one split "all-split" contains all images and the other split "balanced-split" contains balanced-split with re-sampled answer distribution. For ablation studies, all models are trained on the "balanced-split" and tested on the "testdev" set. The final model is trained on the "all-split" and finetuned on the "balanced-split".

4.2 Ablation Study

We perform extensive ablation studies on the GQA "balanced-set". The overall results are shown in Table 1.

TRRNet VS. basic attention models We first investigate the improvement of our proposed TRRNet over basic attention models. For each basic attention model, we perform two experiments. First, we train the network alone on "balanced-split" and second we use the model as the root attention component in a one layer TRRNet with hyper-meter $K = 6$. All the models are trained

with the same hyper-parameter settings. Evaluation is done on the "testdev" dataset. The experiments are carried out on two attention methods, a simple attention model used in Bottom-Up attention [1], a more advanced model used in BAN [19]. Results show that when using weaker attention models, our TRRNet significantly outperforms the baseline by 6%. Even when the baseline model is a strong and complicated attention model, our one layer TRRNet could improve performance by 0.6%. For simplicity, all remaining ablation experiments use Bottom-Up attention as root attention.

Component analysis We then study the influence of root to leaf attention passing. The experiments are done in three folds: first, we build a simple baseline relation module where the relations are generated for each pair of objects. Then we compare the baseline model and root to leaf attention passing with different K choices. Finally, we demonstrate the result of triple-wise relation attention. By simply adding a relation module to Bottom-Up attention, the accuracy is improved by 5%. Surprisingly, by choosing top 6 most significant objects, hard attention with more than 36 times less computation could achieve 0.5% better performance than the relation baseline. This observation further proves the fundamental idea of our networks. Dense relations are usually noisy and using only important objects could generate even better results. By increasing the K value to 12, the model could achieve 0.2% improvement. We continue to generate triple-wise relations for hard attentions. The attention features of triple-wise relations are combined with pairwise features and object level features to generate the final results. The triple-wise relations also help to improve accuracy by 0.2% compared with pairwise relations. Due to the complexity of triple-wise relations, it's not used in our final models.

The Number of TRR units Also we investigate the effect of reasoning steps i.e. the number of TRR units. We start from a 1 layer TRR network and increase the length to 3 layers. We observe that a 2 layer TRRNet significantly improves model performance by almost 1% compared with 1 layer reasoning model. While adding more TRR units to above 3 layers does not help to further improve the performance. With our proposed policy network, the accuracy could be increased further by 0.3% compared with a simple 2 layers TRRNet.

Length of GRU embeddings Finally, we study the length of GRU embeddings. The feature dimension is increased from 512 to 2048. The network achieves the highest accuracy at feature dimension 1024. Longer or shorter feature dimensions all reduce overall performance.

4.3 Experimental Results on GQA

Training Details. In final experiments, we use dual root attentions, one Bottom-Up attention and one BAN attention to better capture object importance. Both of the root attentions are appended with their own leaf attentions. For both leaf attentions, we adopt attention passing with $K = 6$. The final root features and leaf features are concatenated before processed by the classifier. For training, we first use the "all-split" to train our model for 4 epochs with learning rate of 1×10^{-4} and further fine-tune the trained model on the "balanced-split" with a

Method	Accuracy(%)
Bottom-Up [1]	50.20
TRR (Bottom-Up)	**56.13**
BAN [19]	55.82
TRR (BAN)	**56.43**
Bottom-Up + Relation	55.60
TRR Hard Attention (K=6)	56.13
TRR Hard Attention (K=12)	56.28
TRR Triple Attention (K=6)	56.29
TRR 1 layer	56.13
TRR 2 layers	57.00
TRR 3 layers	56.88
TRR Policy	**57.32**
Embedding 512	55.85
Embedding 1024	**56.13**
Embedding 2048	55.90

Table 1. Ablation experiments of TRRNet on the GQA balanced dataset. K stands for the number of objects chosen for hard attention.

Method	Binary	Open	Consistency	Plausibility	Validity	Accuracy
Bottom-Up [1]	66.64	34.83	78.71	84.57	96.18	49.74
MAC [14]	71.23	38.91	81.59	84.48	96.16	54.06
BAN [19]	76.00	40.41	91.70	85.58	96.16	57.10
GRN [11]	74.93	41.24	87.41	84.68	96.14	57.04
LCGN [13]	73.77	42.33	84.68	84.81	96.48	57.07
TRRNet (Ours)	77.83	45.65	90.95	85.15	96.40	**60.74**

Table 2. Comparisons with state-of-the-art methods of GQA on the blind test2019 set. Our model achieves a state-of-the-art performance of 60.74% for a single model without using functional programs and new image features.

learning rate of 5×10^{-5}. A step by step result on the "testdev" split is shown in Table 3.

Comparison with state-of-the-art of GQA. As illustrated in Table 2, our single model achieves state-of-the-art performance on the GQA "test2019" split without new image features and functional program. We notice that [16] proposes to generate symbolic representation and new visual features with scene graph annotations. To further improve performance, we also train Faster-CNN models for feature extraction using the scene graph annotation provided. With new visual features, our model could be further improved by 3% to **63.20%**. Moreover, we use model ensemble and the tiny evaluation trick mentioned in [8] to further improve model performance. An ensemble version of model achieves an accuracy of **74.03%** , ranked the 2nd place on GQA 2019 leaderboard.

Method	Accuracy(%)
Dual Root trained on "balanced-split"	56.10
TRR Dual Root trained on "balanced-split"	57.86
TRR Dual Root trained on "all-split"	57.89
TRR Dual Root fine-tuned on " balanced-split" (ours)	60.32

Table 3. Step-by-step results of the TRRNet Dual Root Attention on GQA "testdev" split.

4.4 Experimental Results on VQAv2 and CLEVR

We also evaluate our proposed TRRNet on VQAv2 dataset and simulated visual reasoning dataset CLEVR.

VQAv2 dataset. For VQAv2 dataset, we use root attention similar to the network structure mentioned in [35] to encode visual and language features and generate root level attentions. Due to the nature of VQAv2 dataset that it does not contain questions that require compositional reasoning skills, we remove the policy network and simply adopt a two layer TRRNet. Testing results are shown in table 4. Our model achieves better results compared with models trained with the same training data and same visual language features. It does extremely well on yes or no questions.

CLEVR dataset. For CLEVR dataset, we use grid features generated from pretrained Resnet, instead of region proposal features from Faster-RCNN. We use [35] as root attention. Since the visual feature is grid feature, we increase the K value to 36. Testing results are shown on table 5. Functional programs work as a very strong supervision for visual reasoning. For a fair comparison, our model is only compared with those methods using natural languages only.

Method	Test-dev				Test-std
	All	Y/N	Num	Other	All
Bottom-up [1]	65.32	81.82	44.21	56.05	65.67
DCN [24]	66.87	83.51	46.61	57.26	66.97
MFH [10]	68.76	84.27	49.56	59.89	-
BAN [19]	70.04	85.42	54.04	60.52	70.35
DFAF [7]	70.22	86.09	53.32	60.49	70.34
MCAN [35]	70.63	85.82	53.26	60.72	70.90
TRRNet (Ours)	**70.80**	**87.27**	51.89	61.02	**71.20**

Table 4. Single model performance on *test-dev* and *test-standard* splits of VQAv2 dataset.

4.5 Visualization

In this section, we show visualizations of the attention maps in our network. Similar to ablation studies, we use Bottom-Up attention as the root attention.

Fig. 3. Visualization of attention maps. For root attention, we plot all object regions with attention score larger than 0.1. For leaf attention, we only display the most confident relation attended. The first two examples show the model's ability to identifying the right objects. Example 3 shows the network's power of learning common sense: the correlation of weather and sky. Example 4 demonstrates an example of three stage reasoning.

Method	All	Count	Exist	Compare Numbers	Query Attribute	Compare Attribute
Human [17]	92.6	86.7	96.6	86.5	95.0	96.0
CNN+LSTM+SA [17]	73.2	59.7	77.9	75.1	80.9	70.8
RN [26]	95.5	90.1	97.8	93.6	97.9	97.1
FiLM [25]	97.7	94.3	99.1	96.8	99.1	99.1
MAC [14]	**98.9**	97.1	99.5	99.1	99.5	99.5
TRRNet (Ours)	**98.8**	96.8	99.5	98.9	99.6	99.3

Table 5. Single model performance on CLEVR dataset.

Both root attention and leaf attention visualizations are shown in Fig. 3. The images are chosen from the "testdev" set. Interestingly, the majority of the test images are reasoned for two stages. For root attentions, we plot objects with attention scores more than 0.1. For leaf attentions, we only show the relation with the highest attention score.

The first example shows an easy question that contains only one relation. In stage1, the root attention could already identify the most important area for answering questions. The second example shows a difficult question that contains more than one relations. Although the name "cake" is not directly displayed in the question, the root attention could successfully focus on the object. The third example is asking for abstraction. There is no clue from the words to guide where the attention should look and no relations can be found from the sentence. Visualizations show that the model learns to locate meaningful areas: the sky. The final example shows an example of three stage reasoning.

5 Conclusion

In this work, we propose a tiered relation reasoning method that dynamically selects object level candidates based on language representation and generates tidy object relations within the selected candidates objects only. The tiered selection not only makes the network computational efficient but also improves relation reasoning performance. Moreover, we propose a policy network that decides the appropriate reasoning steps based on question complexity and current reasoning status. In experiments, our model achieves state-of-the-art performance on GQA dataset and competitive results on CLEVR datasets and VQAv2 dataset without functional program supervision.

Acknowledgements

This research was supported by the National Research Foundation Singapore under its AI Singapore Programme (Award Number: AISG-RP-2018-003) and the MOE Tier-1 research grants: RG28/18 (S) and RG22/19 (S). F. Lv's participation is supported by National Natural Science Foundation of China (No.11829101 and 11931014)

References

1. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6077–6086 (2018)
2. Andreas, J., Rohrbach, M., Darrell, T., Klein, D.: Neural module networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 39–48 (2016)
3. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: Vqa: Visual question answering. In: Proceedings of the IEEE international conference on computer vision. pp. 2425–2433 (2015)
4. Cadene, R., Ben-Younes, H., Cord, M., Thome, N.: Murel: Multimodal relational reasoning for visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1989–1998 (2019)
5. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
7. Gao, P., Jiang, Z., You, H., Lu, P., Hoi, S.C., Wang, X., Li, H.: Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6639–6648 (2019)
8. Geng, S., Zhang, J., Zhang, H., Elgammal, A., Metaxas, D.N.: 2nd place solution to the gqa challenge 2019. arXiv preprint arXiv:1907.06794 (2019)
9. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
10. 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: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6904–6913 (2017)
11. Guo, D., Xu, C., Tao, D.: Graph reasoning networks for visual question answering. arXiv preprint arXiv:1907.09815 (2019)
12. Hu, R., Andreas, J., Rohrbach, M., Darrell, T., Saenko, K.: Learning to reason: End-to-end module networks for visual question answering. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
13. Hu, R., Rohrbach, A., Darrell, T., Saenko, K.: Language-conditioned graph networks for relational reasoning. arXiv preprint arXiv:1905.04405 (2019)
14. Hudson, D.A., Manning, C.D.: Compositional attention networks for machine reasoning. arXiv preprint arXiv:1803.03067 (2018)
15. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6700–6709 (2019)
16. Hudson, D.A., Manning, C.D.: Learning by abstraction: The neural state machine. arXiv preprint arXiv:1907.03950 (2019)
17. Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., Girshick, R.: Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2901–2910 (2017)

18. Johnson, J., Hariharan, B., van der Maaten, L., Hoffman, J., Fei-Fei, L., Lawrence Zitnick, C., Girshick, R.: Inferring and executing programs for visual reasoning. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2989–2998 (2017)
19. Kim, J.H., Jun, J., Zhang, B.T.: Bilinear attention networks. In: Advances in Neural Information Processing Systems. pp. 1564–1574 (2018)
20. 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: Advances in neural information processing systems. pp. 361–369 (2016)
21. Lu, J., Yang, J., Batra, D., Parikh, D.: Hierarchical question-image co-attention for visual question answering. In: Advances In Neural Information Processing Systems. pp. 289–297 (2016)
22. Malinowski, M., Doersch, C., Santoro, A., Battaglia, P.: Learning visual question answering by bootstrapping hard attention. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–20 (2018)
23. Mascharka, D., Tran, P., Soklaski, R., Majumdar, A.: Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4942–4950 (2018)
24. Nguyen, D.K., Okatani, T.: Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6087–6096 (2018)
25. Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A.: Film: Visual reasoning with a general conditioning layer. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
26. Santoro, A., Raposo, D., Barrett, D.G., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T.: A simple neural network module for relational reasoning. In: Advances in neural information processing systems. pp. 4967–4976 (2017)
27. Sukhbaatar, S., Weston, J., Fergus, R., et al.: End-to-end memory networks. In: Advances in neural information processing systems. pp. 2440–2448 (2015)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
29. Xiong, C., Zhong, V., Socher, R.: Dynamic coattention networks for question answering. arXiv preprint arXiv:1611.01604 (2016)
30. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning. pp. 2048–2057 (2015)
31. Yang, X., Tang, K., Zhang, H., Cai, J.: Auto-encoding scene graphs for image captioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
32. Yang, Z., He, X., Gao, J., Deng, L., Smola, A.: Stacked attention networks for image question answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 21–29 (2016)
33. Yi, K., Wu, J., Gan, C., Torralba, A., Kohli, P., Tenenbaum, J.: Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In: Advances in Neural Information Processing Systems. pp. 1031–1042 (2018)
34. Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., Berg, T.L.: Mattnet: Modular attention network for referring expression comprehension. In: Proceedings of

the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1307–1315 (2018)
35. Yu, Z., Yu, J., Cui, Y., Tao, D., Tian, Q.: Deep modular co-attention networks for visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6281–6290 (2019)
36. Yu, Z., Yu, J., Xiang, C., Fan, J., Tao, D.: Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering. IEEE transactions on neural networks and learning systems **29**(12), 5947–5959 (2018)
37. Zhu, C., Zhao, Y., Huang, S., Tu, K., Ma, Y.: Structured attentions for visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1291–1300 (2017)