LRRA:A Transparent Neural-Symbolic Reasoning Framework for Real-World Visual Question Answering

Zhang Wan[*], **Keming Chen**, **Yujie Zhang**[†], **Jinan Xu**, **Yufeng Chen**
School of Computer and Information Technology, Beijing Jiaotong University,
Beijing 100044, China
{19120413,20120341,yjzhang,jaxu,chenyf}@bjtu.edu.cn

Abstract

The predominant approach of visual question answering (VQA) relies on encoding the image and question with a "black box" neural encoder and decoding a single token into answers such as "yes" or "no". Despite this approach's strong quantitative results, it struggles to come up with human-readable forms of justification for the prediction process. To address this insufficiency, we propose LRRA[Look,Read,Reasoning,Answer],a transparent neural-symbolic framework for visual question answering that solves the complicated problem in the real world step-by-step like humans and provides human-readable form of justification at each step.Specifically, LRRA learns to first convert an image into a scene graph and parse a question into multiple reasoning instructions. It then executes the reasoning instructions one at a time by traversing the scene graph using a recurrent neural-symbolic execution module.Finally, it generates answers to the given questions and makes corresponding marks on the image. Furthermore,we believe that the relations between objects in the question is of great significance for obtaining the correct answer, so we create a perturbed GQA test set by removing linguistic cues (attributes and relations) in the questions to analyze which part of the question contributes more to the answer.Our experiments on the GQA dataset show that LRRA is significantly better than the existing representative model (57.12% vs. 56.39%). Our experiments on the perturbed GQA test set show that the relations between objects is more important for answering complicated questions than the attributes of objects.

Keywords: Visual Question Answering, Relations Between Objects, Neural-Symbolic Reasoning.

1 Introduction

Currently, the predominant approach to visual question answering (VQA)relies on encoding the image and question with a black-box transformer encoder (H.Tan et al., 2019; J.Lu et al., 2020). These works carry out complex calculations behind the scenes but only produce a single token as prediction output (for example, "yes", "no") and they can not provide an easy-to-understand form of justification consistent with their predictions. In addition, recent studies have shown that the end-to-end model can be easily optimized to learn the "shortcut bias" of the data set instead of reasoning (for example, the model uses the implicit fused question representations (S.Antol et al., 2015; Y. Goyal et al., 2017), the answer can be directly inferred according to certain language patterns), which tend to undesirably adhere to superficial or even potentially misleading statistical associations(A.Agrawal et al., 2016), so they do not really understand the question, and often perform poorly in the face of complex reasoning problems in the real world. In order to solve the above insuficiencys, we learn the correct problem solving process step-by-step mimicking humans and propose a neural-symbolic approach for visual question answering that fully disentangles vision and language understanding from reasoning. A human would first (1)

[*]First author
[†]Corresponding author

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1037–1045, Hohhot, China, August 13 – 15, 2021.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

look at the image, (2) read the question, (3) reason and think (4) answer questions. Following this intuition, our model deploys four neural modules, each mimicking one problem solving step that humans would take: A scene graph generation module first converts an image into a scene graph; A semantic parsing module parses each question into multiple reasoning instructions; A neural execution module interprets reason instructions one at a time by traversing the scene graph in a recurrent manner; Answer generation module predicts the answer with the highest probability. These four modules are connected through hidden states instead of explicit outputs. Therefore, the entire framework can be trained end-to-end from pixels to answers. In addition, since LRRA also produces human-readable output from individual modules during testing, we can easily locate the error by checking the modular output. Our experiments on the GQA dataset show that LRRA is significantly better than the existing representative model (57.12% vs. 56.39%).Furthermore,we believe that the relations between objects in the question is of great significance for obtaining the correct answer,so we create a perturbed GQA test set by removing linguistic cues (attributes and relations) in the questions to analyze which part of the question contributes more to the answer.Ablation experiment further show that the relations between objects is more important for answering complicated questions than the attributes of objects.To summarize, the main contributions of our paper are threefold:

- When we give the answer, we also make the corresponding mark on the image to improve explainability and discourage superficial guess for answering the questions.
- We propose an end-to-end trainable modular VQA framework LRRA. Compared with contemporary black-box methods, it has interpretability and enhanced error analysis capabilities.
- We create a perturbed GQA test set that provides an effificient way to validate our approach on the perturbed dataset.The dataset will be announced soon.

2 Related Work

Visual Reasoning. It is the process of analyzing visual information and solving problems based on it. The most representative benchmark of visual reasoning is GQA(Hudson et al., 2019) a diagnostic visual Q&A dataset for compositional language and elementary visual reasoning. The majority of existing methods on GQA can be categorized into two families: 1) holistic approaches (J.Johnson et al., 2017; A.Santoro et al., 2017; E. Perez et al., 2018; D.A.Hudson et al., 2018), which embed both the image and question into a feature space and infer the answer by feature fusion; 2) neural module approaches (J. Andrea et al., 2016; R.Hu et al., 2017; J. Johnson et al., 2017; D. Mascharka et al., 2018; R. Hu et al., 2018), which first parse the question into a program assembly of neural modules, and then execute the modules over the image features for visual reasoning. Our LRRA belongs to the second one but replaces the visual feature input with scene graphs.

Neural Module Networks. They dismantle a complex question into several subtasks, which are easier to answer and more transparent to follow the intermediate outputs. Modules are predefined neural networks that implement the corresponding functions of subtasks, and then are assembled into a layout dynamically, usually by a sequence-to-sequence program generator given the input question. The assembled program is finally executed for answer prediction(R.Hu et al., 2017; J. Johnson et al., 2017; D. Mascharka et al., 2018). In particular, the program generator is trained based on the human annotations of desired layout or with the help of reinforcement learning due to the nondifferentiability of layout selection. Recently, Hu et al. (R. Hu et al., 2018) proposed StackNMN, which replaces the hard-layout with soft and continuous module layout and performs well even without layout annotations at all. Our LRRA experiments on GQA follows their softprogram generator.

Recently, NS-VQA(K. Yi et al., 2018) firstly built the reasoning over the object-level structural scene repre-sentation, improving the accuracy on CLEVR from the previous state-of-the-art 99.1% (D. Mascharka et al., 2018) to an almost perfect 99.8%. Their scene structure consists of objects with detected labels, but lacked the relationships between objects, which limited its application on real-world datasets such as GQA(Hudson et al., 2019). In this paper, we propose a much more generic framework

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1037-1045, Hohhot, China, Augest 13 - 15, 2021.

for visual reasoning over scene graphs, including object nodes and relationship edges represented by either labels or visual features. Our scene graph is more flexible and more powerful than the table structure of NS-VQA.

Scene Graphs. This task is to produce graph representations of images in terms of objects and their relationships. Scene graphs have been shown effective in boosting several vision-language tasks (J. Johnson et al., 2015; D. Teney et al., 2017; X. Yin et al., 2018). However, scene graph detection is far from satisfactory compared to object detection (D. Xu et al., 2017; R. Zellers et al., 2018; Y. Li et al., 2018). To this end, our scene graph implementation also supports cluttered and open-vocabulary in real-world scene graph detection, where the nodes are merely RoI features and the edges are their relations.

3 Approach

We build our neural module network over scene graphs to tackle the visual reasoning challenge. As shown in Fig 1, given an input image andquestion, we first parse the image into a scene graph and parse the question into a module program, an then execute the program over the scene graph. Besides, our approach are totally attention-based,making all theintermediate reasoning steps transparent.The model framework as shown in Figure 1.

Scene Graph Generation Given an image I, its corresponding scene graph represents the objects in the image (e.g., girl, hamburger) as nodes and the objects' pairwise relationships (e.g., holding) as edges. The fifirst step of scene graph generation is object detection. We use DETR(Y. Li et al., 2018) as the object detection backbone since it removes the need for for hand-designed components like non-maximum suppression. DETR(Y. Li et al., 2018) feeds the image feature from ResNet50 (K. He et al., 2016) into a nonau-toregressive transformer model, yielding an orderless set of N object vectors $[o_1, o_2, \cdots, o_N]$,as in (1). Each object vector represents one detected object in the image. Then, for each object vector, DETR uses an object vector decoder (feed-forward network) to predict the corresponding object class (e.g., girl), and the bounding box in a multi-task manner. Since the set prediction of N object vectors is orderless, DETR calculates the set prediction loss by first computing an optimal matching between predicted and ground truth objects, and then sum the loss from each object vector. N is fifixed to 100 and DETR creates a special class label "no object", to represent that the object vector does not represent any object in the image.

$$[o_1, o_2, \cdots, o_N] = \text{DETR} \tag{1}$$

The object detection backbone learns object classes and bounding boxes, but does not learn object attributes, and the objects' pairwise relationships. We augment the object vector decoder with an additional object attributes predictor. For each attribute meta-concept (e.g., color), we create a classifier to predict the possi-ble attribute values (e.g., red, pink). To predict the relationships, we consider all $N(N-1)$ possible pairs of object vectors, $[e_1, e_2, \cdots, e_{N(N-1)}]$. The relation encoder transforms each object vector pair to an edge vector through feed-forward and normalization layers as in (2). We then feed each edge vector to the relation decoder to classify its relationship label. Both object attributes and inter-object relationships are supervised in a multitask manner. To handle the object vector pair that does not have any relationship, we use the "no relation" relationship label.

We construct the scene graph represented by N object vectors and $N(N-1)$ edge vectors instead of the symbolic outputs, and pass it to downstream modules.

$$e_{i,j} = \text{LayerNorm}(\text{FeedForward}(o_i \oplus o_j)) \tag{2}$$

Semantic Parsing The semantic parser works as a "compiler" that translates the question tokens $(q_1, q_2, \cdots, q_Q)$ into an neural executable program, which consists of multiple instruction vectors. We adopt a hierarchical sequence generation design: a transformer model (K. He et al., 2016) first parses the question into a sequence of Minstruction vectors, $[i_1, i_2, \cdots, i_M]$. The i^{th} instruction vector will correspond exactly to the i^{th} execution step in the neural execution.engine. To enable human to understand

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1037-1045, Hohhot, China, Augest 13 - 15, 2021.

the semantics of the instruction vectors, we further translate each instruction vector to human-readable text using a transformer-based instruction vector decoder. We pass the M instruction vectors rather than the human-readable text to the neural execution module.

$$[i_1, i_2, ..., i_M] = \text{Transformer}(q_1, ..., q_Q) \tag{3}$$

Visual Reasoning The neural execution engine works in a recurrent manner: At the m^{th} time step, the neural execution engine takes the m^{th} instruction vector (i_m) and outputs the scene graph tra-versal result. Similar to recurrent neural networks, a history vector that summarizes the graph traversal states of all nodes in the current time-step would be passed to the next time-step. The neural execution engine operates with graph neural network. Graph neural network generalizes the convolution operator to graphs using the neighborhood aggregation scheme(P. W. Battaglia et al., 2018; K. Xu et al., 2019). The key intuition is that each node aggregates feature vectors of its immediate neighbors to compute its new feature vector as the input for the following neural layers. Specifically, at m^{th} time step given a node as the central node, we first obtain the feature vector of each neighbor (f_k^m) through a feed-forward network with the following inputs: the object vector of the neighbor (o_k) in the scene graph, the edge vector between the neighbor node and the central node ($e_{k,central}$) in the scene graph, the $(m-1)^{th}$ history vector (h_{m-1}), and the m^{th} instruction vector (i_m).

$$f_k^m = \text{FeedForward}(o_k \oplus e_{k,central} \oplus h_{m-1} \oplus i_m) \tag{4}$$

We then average each neighbor's feature vector as the context vector of the central node

$$c_{central}^m = \frac{1}{K} \sum_{k=1}^{K} f_k^m \tag{5}$$

Next, we perform node classification for the central node, where an "1" means that the corre-sponding node should be traversed at the m^{th} time step and "0" otherwise. The inputs of the node classifier are: the object vector of the central node in the scene graph, the context vector of the central node, and the m^{th} instruction vector.

$$s_{central}^m = \text{Soft}\max(\text{FeedForward}(o_{central} \oplus c_{central}^m \oplus i_m)) \tag{6}$$

where $s_{central}^m$ is the classification confidence score of central node at m^{th} time step. The node classification results of all nodes constitute a bitmap as the scene graph traversal result. We calculate the weighted average of all object vectors as the history vector (h_m), where the weight is each node's classification confidence score.

$$h_m = \sum_{i}^{N} s_i^m \cdot o_i \tag{7}$$

Predict answer VQA is commonly formulated as a classification problem where the model learns to answers with one token (e.g., "yes" or "no"). To do this, the language output at the last step is passed to a feed-forward network with softmax activation to obtain the distribution for the predicted answers.

$$w = atgmax(softmax(wh_t)) \tag{8}$$

End-to-End Training: From Pixels to Answers We connect four modules through hidden states rather than symbolic outputs(W. Liang et al., 2020). Therefore, the whole framework could be trained in an end-to-end manner, from pixels to answers. The training loss is simply the sum of losses from all four mod-ules. Each neural module receives supervision not only from the module's own loss, but also from the gradient signals backpropagated by downstream modules. We start from the pretrained weights of DETR for the object detection backbone and all other neural modules are randomly initialized.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1037-1045, Hohhot, China, Augest 13 - 15, 2021.

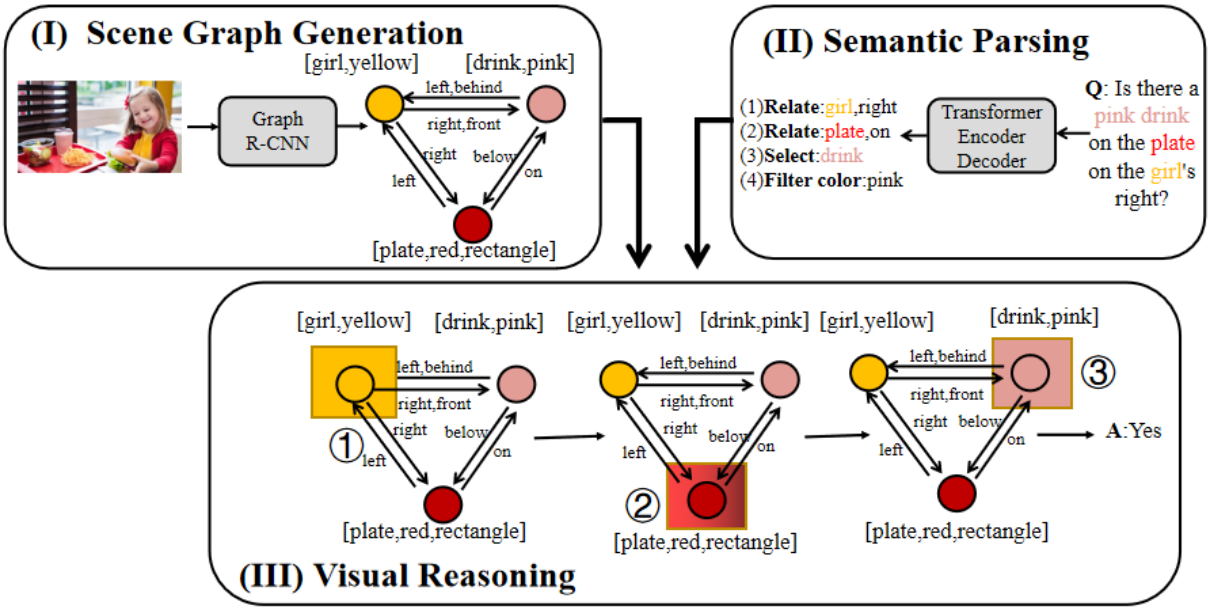

Figure 1: The framework of LRRA model

4 Experiments

4.1 Dataset

We demonstrate the value and performance of our model on the "balancedsplit" of GQA v1.1, which contains 1M questions over 140K images with a more balanced answer distribution. Compared with the VQA v2.0 dataset (Ren, S et al., 2015), the questions in GQA are designed to require multi-hop reasoning to test the reasoning skills of developed models. Compared with the CLEVR dataset (W. Chen et al., 2019), GQA greatly increases the complexity of the semantic structure of questions, leading to a more diverse function set. The real-world images in GQA also bring in a bigger challenge in visu-al understanding. Following (W. Chen et al., 2019), the main evaluation metrics used in our experiments are accuracy, validity and distribution.

4.2 Implementation Details

We first pre-trained DETR for object detection, and then fix the parameters in the backbone to train the scene graph generation model. SGD is used as the optimizer, with initial learning rate 1e-2 for both training stages. For question parser, we train with learning rate 7×10-4 for 20,000 iterations. The batch size is fixed to be 64.

4.3 Results

We evaluated our method on the GQA dataset (Hudson et al., 2019), which contains 1.5 million questions out of 1.1 million images. We use standard data set splitting. In the training process, we use the basic facts of the scene graph, reasoning explanation, and the traversal result of the scene graph for each step. During the test, we only used images and questions. We will present the state-of-the-art model LXMERT (H.Tan et al., 2019) as a baseline. We report the accuracy of the answers of LXMERT and LRRA.

VQA We compare our performance both with baselines, as appear in (Hudson et al., 2019), as well as with other prior arts of VQA model. Apart from the standard accuracy metric and the more detailed type-based diagnosis (i.e. Binary, Open), we get further insight into reasoning capabilities by reporting three more metrics (Hudson et al., 2019): Validity, and Distribution. The validity metric checks whether a given answer is in the question scope, e.g. responding some color to a color question. The distribution score measures the overall match between the true answer distribution and the model predicted distribution (for this metric, lower is better). As Table 1 shows, our model achieves competitive accuracy

among.

Model	Accuracy	Distribution ↓	Binary	Validity
Language (Hudson et al., 2019)	41.07	16.63	60.39	95.70
BottomUp (Anderson.P et al.,2018)	49.74	5.60	65.64	94.13
MAC (D.A.Hudson et al., 2018)	54.05	5.14	70.49	96.16
LCGN (Hu, R et al.,2019)	56.28	4.30	74.87	96.48
Vision (Hudson et al., 2019)	18.93	19.27	36.05	–
LXMERT (H.Tan et al., 2019)	56.39	4.80	75.16	96.35
Ours	57.12	3.75	74.87	96.87

Table 1: VQA results on GQA data sets.

Table 2. The accuracy (%) of our question parser and symbolic executor. Program Acc. repre-sents the accuracy of generated program, which is evaluated by the accuracy of operation token, arguments token and the function (It is positive when both operation and arguments in a function are correct). Executor Acc. represents the accuracy of the answers obtained by our deterministic part of program executor executed on the ground-truth scene graph, by using ground-truth (G.T.) and generated (Gen.) program.

Data Split	Program Acc			Executor Acc	
	Operation	Arguments	Function	G.T	Gen
Testdev	96.65	80.49	81.34	-	-
Val	97.49	82.50	81.75	96.84	90.46

Table 2: The accuracy of LRRA model in question parser and symbolic executor

Re-posuton GQA Dataset and Additional Analysis Finally, we use a comprehensive list of attributes obtained by (W. Chen et al., 2019) and mask them using a prede-fined mask token. For effectively mask-ing relationships, we use Spacy POS-Tagger (M. Honnibal et al., 2017) and mask verbs (VB) and prepo-sitions (PRPN) from the question. The results are reported in Table 3. For attributes, we see that LRRA performance drops by 21.35% as compared to 8.07% drop in LXMERT, while for relations, the margin is more signifificant at 27.73% and 18.33% respectively, thus providing us a strong convergent evidence for our hypothesis that the relations between objects is more important for answering complicated questions than the attributes of objects.

Model	Acc Drop(from→to)
Relations masked	
LXMERT	18.33% (55.49%→37.16%)
LRRA	27.73% (57.15%→29.39%)
Attributes masked	
LXMERT	8.07% (55.49%→47.42%)
LRRA	21.35% (57.15%→35.80%)

Table 3: Perturbation analysis on testdev set.

4.4 Example analysis

Illustrative execution trace generated by our Neuro-Symbolic Concept Learner on the GQA dataset. Execution traces A and B shown in the figure leads to the correct answer to the question.

Our model effectively learns visual concepts from data. The symbolic reasoning process brings trans-parent execution trace and can easily handle quantities (e.g., object counting in Example A).

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1037-1045, Hohhot, China, Augest 13 - 15, 2021.

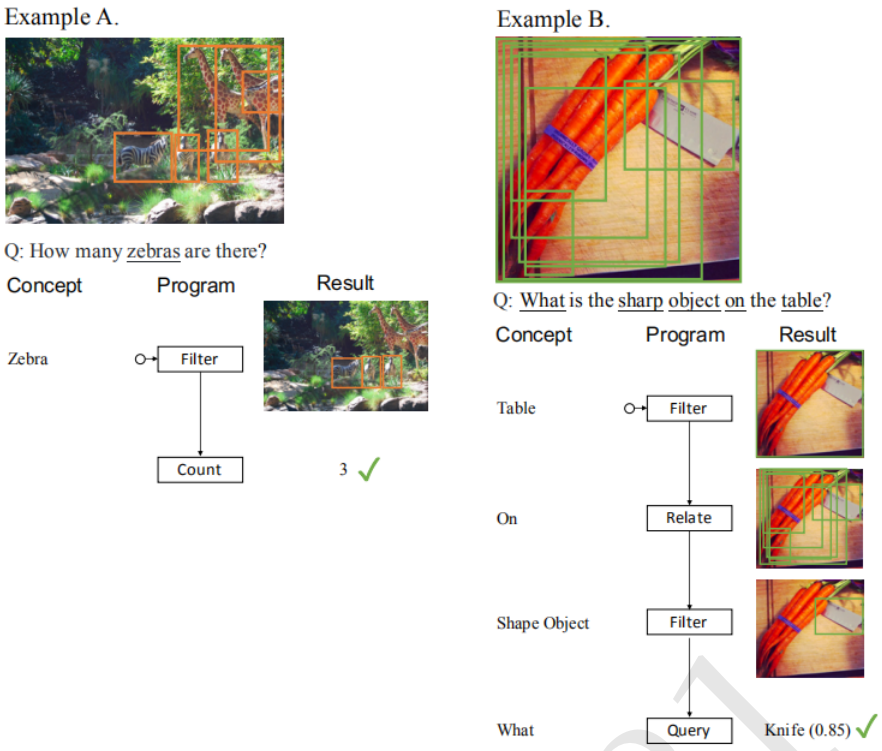

Figure 2: Example of GQA data set generation

5 Conclusion

We present a transparent neural-symbolic reasoning framework for visual question answering, providing a human-readable form of justification at each step. The modular design of our methodology enables the whole framework to be trainable end-to-end. Our experiments on GQA dataset show that LRRA achieves high accuracy on answer generation task, outperforming the state-of-the-art LXMERT results. In addition, Our experiments on the perturbed GQA test set show that the relations between objects is more important for answering complicated questions than the attributes of objects. Furthermore LRRA performance drops significantly more than LXMERT, when object attributes and relationships are masked, hence indicating that LRRA makes a step forward,towards truly understanding the question.

Acknowledgements

The research work descried in this paper has been supported by the National Nature Science Foundation of China(Contract 61876198, 61976015, 61976016). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China, November 2019. Association for Computational Linguistics.

J. Lu, V. Goswami, M. Rohrbach, D. Parikh, and S. Lee. 12-in-1: Multi-task vision and language representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C.Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2425–2433, 2015.

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 *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6325–6334, 2017.

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. Analyzing the behavior of visual question answering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960, Austin, Texas, November 2016. Association for Computational Linguistics.

D.A. Hudson and C. D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

J. Johnson, B. Hariharan, Lvd Maaten, F. F. Li, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

E. Perez, F. Strub, H De Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer. *AAAI*, 2018. 2

Drew.A Hudson and Christopher D Manning. Compositional attention networks for machine reasoning. In *International Conference on Learning Representations (ICLR)*, 2018.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 39–48, 2016.

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 804–813, 2017.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Judy Hoffman, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3008–3017, 2017.

David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4942–4950, 2018.

R. Hu, J. Andreas, T. Darrell, and K. Saenko. Explainable neural computation via stack neural module networks. *European Conference on Computer Vision*, 2018.

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678, 2015.

D. Teney, L. Liu, and Avd Hengel. Graph-structured representations for visual question answering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

X. Yin and V. Ordonez. Obj2text: Generating visually descriptive language from object layouts. 2017.

Long Chen, Hanwang Zhang, Jun Xiao, Xiangnan He, Shiliang Pu, and Shih-Fu Chang. Counterfactual critic multi-agent training for scene graph generation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4612–4622, 2019.

Danfei Xu, Yuke Zhu, Christopher Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5831–5840, 2018.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1037-1045, Hohhot, China, Augest 13 - 15, 2021.

Y. Li, W. Ouyang, B. Zhou, Y. Cui, and X. Wang. Factorizable net: An efficient subgraph-based framework for scene graph generation. *Springer, Cham*, 2018.

N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. *End-to-End Object Detection with Transformers*. 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

Peter W Battaglia, Jessica B Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, and R. Faulkner. Relational inductive biases, deep learning, and graph networks. 2018.

K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? 2018.

W. Liang, Tian Y, C. Chen, and Z. Yu. Moss: End-to-end dialog system framework with modular supervision. 2019.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

W. Chen, Z. Gan, L. Li, Y. Cheng, W. Wang, and J. Liu. Meta module network for compositional visual reasoning. 2019.

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 *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018.

Ronghang Hu, Anna Rohrbach, Trevor Darrell, and Kate Saenko. Language-conditioned graph networks for relational reasoning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10293–10302, 2019.

M. Honnibal and I. Montani. spacy 2: Natural language understanding with bloom embeddings, con-volutional neural networks and incremental parsing. *To appear*, 2017.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1037-1045, Hohhot, China, Augest 13 - 15, 2021.