Efficient Visual Reasoning with Language Conditioning

Anonymous EMNLP submission

1 Introduction

Reasoning about everyday visual input is one of the most fundamental building blocks of human intelligence. The problem of visual reasoning has attracted wide attention from the research community. A number of approaches have been proposed.

Motivation In assessment of multiple possible solutions to VQA problems, it is hard to have an objective measurement of the improvements claimed by each work. Given that VQA is a complex problem, it is important yet difficult to have a breakdown analysis and understand why each work fails in some cases.

Another major concern for the VQA community is that the models are prone to learning biases in the datasets rather than the true reasoning abilities. Models might achieve seemingly good performance on certain datasets, according to certain metric numbers, however, it is still doubtful if the models really learn the reasoning ability which should be able to transfer to unseen tasks with ease.

VQA is a rapidly evolving field which witnesses many achievements and even more challenges. It is important to understand the trends and challenges in this field, in order to contribute to the next generation VQA models.

The above-mentioned problems and concerns provide the source of motivation for this research project:

First, develop a deeper understanding of the trends and challenges in this field.

Second, understand each method's improvements and drawbacks.

Third, assess and understand if VQA models really learn the reasoning ability.

Goals We make the hypothesis that recent VQA models still lack the reasoning abilities even if they might have remarkable performance on certain datasets. Hence, the ultimate goal of this research project is to find ideas for next generation VQA models, which might requires extensive literature review and experiments. However, due to constraints in time and computational resources, for this report we might need to narrow down the goals to the following:

Make a survey of recent methods for VQA and complete a trend analysis in the report;

Make a retrospective comparison and analysis of relevant VQA models;

Use both quantitative and qualitative analysis to find out evidence whether VQA models learn reasoning abilities.

Problem Statement In this project, I will make a survey, in order to find trends and emerging problems in the field of visual question answering.

For comparison and analysis of VQA algorithms, in this project I will consider 4 baseline models: Antol and Parikh (2015)'s LSTM model, LSTM+CNN framework, Yang and Smola (2016)'s LSTM+CNN+Stacked Attention framework, and LSTM+CNN+Stacked Attention+MLP framework.

I will also investigate 3 more advanced approaches: Johnson and Girshick (2017)'s Program Generator + Execution Engine (PG+EE) algorithm, Perez et al. (2018)'s FiLM algorithm, and the neuro-symbolic VQA algorithms such as Yi et al. (2018)'s NS-VQA and Mao et al. (2019)'s NSCL algorithms.

Input For literature review and trend analysis, the input are the relevant papers mentioned in the reference.

For experiments, the inputs are image and language features extracted from pictures and questions in Johnson et al. (2017)'s CLEVR dataset.

Output For literature review and trend analysis, the output is a report of the trends and current methods in the field of Visual Question Answering.

For experiments, the output will be the answers produced by the VQA algorithms. The output will be compared with the ground truth to generate the

accuracy of the answers as one of the major indicators for each VQA algorithm's performance.

Assumption In this paper, we assume that Johnson et al. (2017)'s CLEVR dataset has ruled out most of the potential biases, so that our comparison will be fair and reflect each model's reasoning abilities.

2 Related Work

Among all VQA models that achieve top performance on CLEVR, to the best of my knowledge, there are 3 main streams of research. Johnson and Girshick (2017) proposed the "Program Generator + Execution Engine" (PG + EE) approach, which decomposes visual question answering tasks into sequence of modular sub-problems and use reinforcement learning techniques to dynamically assemble the neural networks to produce answers. Perez et al. (2018) proposed FiLM, which has a conditional batch normalization layer that allows self modulation of image features according to encoding of the questions. Yi et al. (2018) Mao et al. (2019) Mao et al. (2021) proposed the neuro-symbolic approach, which seeks to marry deep representation learning for visual recognition and language understanding and symbolic execution for reasoning.

These three approaches are based on different insights and have different advantages. Johnson and Girshick (2017)'s Program Generator + Execution Engine (PG + EE) architecture is heuristic and matches people's intuition the best. Perez et al. (2018)'s FiLM architecture has a very large model capacity, and similar architecture designs have been used in GANs and proved successful Zhang et al. (2021). Yi et al. (2018)s approach involves syntactic parser and requires less training data.

In the domain of artificial intelligence, we often see the competitions among different architecture designs. Sometimes the results are not conclusive, and the "winner" might change over time as we improve the implementations and alter multiple factors. So, it is not clear which of the above-mentioned 3 approaches have the greatest potential to capture the interaction among images, questions, and reasoning. In this project, I proposed to compare the performance of the 3 approaches in a retrospective way.

All the above-mentioned works realized that previous models are prone to learning merely the biases in the datasets, and they all set the aim of equipping VQA models with real reasoning abilities. On one hand, these works did provide convincing evidence that their works made contributions to learning the real reasoning abilities; on the other hand, we had to admit that reasoning is still far from a fully-solved problem. While these works focus more on the positive side how they contribute to modelling real reasoning abilities, in this work, I will change the perspective and focus more on the downside: does these work exhibit real reasoning abilities in VQA tasks? How far away are these VQA models from the ultimate triumph of reason? Since I realize that metric numbers cannot fully reflect the models' capacities and performances, in this work I will make qualitative analysis with the focus on the failure cases of these models to find evidence whether these models have learned certain degree of reasoning abilities.

All the papers mentioned above put emphasis on their uniqueness and their contributions. However, in order to have a better understanding of the "big picture" in the field of VQA, in this work I propose to make a trend analysis to re-evaluate the 3 approaches and to find challenges and common trends in VQA research works.

3 Approach

This work includes two parts: one is the literature review whose purpose is to find the trends and challenges in the field of VQA, another is the experiment whose purpose is to establish a retrospective evaluation of each VQA model's performance and reasoning abilities.

3.1 Trends Analysis

In this part, I shall make a comprehensive review of the relevant literature and find out trends and emerging problems in the field of visual question answering and reasoning. The relevant papers include: Johnson and Girshick (2017), Perez et al. (2018), and Yi et al. (2018). I have summarized my analysis and findings in this report.

3.2 VQA models

In this part, to evaluate VQA models' performances and abilities, I include 4 baseline models: the LSTM model (Johnson and Girshick (2017)), the LSTM + CNN model (Johnson and Girshick (2017)), the LSTM + CNN + Stacked Attention model (Johnson and Girshick (2017)), and the LSTM + CNN + Stacked Attention + MLP model

(Johnson and Girshick (2017)). I also include 2 advanced models: PG + EE (Johnson and Girshick (2017)), and NS-VQA (Yi et al. (2018)) .

LSTM In LSTM's framework, questions are first processed with learned word embeddings, then they are fed into a word level LSTM Hochreiter and Schmidhuber (1997). Following that, the LSTM's hidden states will be passed down to a multi-layer perceptron (MLP) which predicts distributions over candidate answers. In this framework, only features from the questions are extracted, and the model has no access to the images features.

LSTM + CNN In LSTM + CNN's framework, LSTM is used to extract features from the questions, and CNN is used to extract features from the images. Then, the multi-modal features will be concatenated together and fed into a MLP to generate the distribution over the possible answers.

LSTM + CNN + Stacked Attention Similar to the LSTM + CNN's framework, the image and language features are extracted by CNN and LSTM, respectively. Then, the features are fed into the stacked attention network, which combines the features by one or more rounds of soft spatial attention. At last, the final answer distribution is made by a linear transformation of the attention output.

LSTM + CNN + Stacked Attention + MLP LSTM + CNN + Stacked Attention + MLP's framework is similar to LSTM + CNN + Stacked Attention's framework, except that the attention output is now processed by a MLP.

PG + EE PG + EE consists of a program generator and an execution engine.

The program generator predicts programs z from questions q:

$$z = \pi(q)$$

The question q is a sequence of words, and the output program's abstract syntax tree is serialized by prefix traversal, which allows the program generator to be implemented by a LSTM sequence-to-sequence model.

The execution engine uses a predicted program z and an input image x as input and outputs a predicted answer a:

$$a = \phi(x, z)$$

The execution engine is implemented with neural module network Andreas and Klein (2016). For each function $f \in \mathcal{F}$ generated by the program

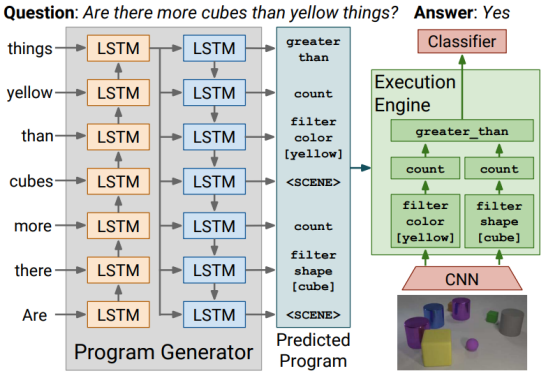

Figure 1: Overview of PG + EE's architecture

generator, the execution engine maintains a module network m_f. Following the program z, the execution engine assemble module networks m_f together in the order defined by the program, and create the network $m(z)$. Then $m(z)$ is executed on the image to generate the predicted answers. Here image features are provided by ResNet-101 pretrained on ImageNet.

NS-VQA NS-VQA consists of a scene parser, a question parser, and a program executor.

The scene parser uses Mask R-CNN to generate segment proposals of objects first. Then ResNet-34 is used to extract attributes of the objects.

The question parser is an attention-based sequence-to-sequence model with an encoder-decoder structure. Taking the questions as inputs, the encoder generates encoded vectors e_i, and the decoder outputs a vector q_t which is used to generate the context vector c_t by attention mechanisms:

$$\alpha_{ti} \propto exp(q_t^T W_A e_i), c_t = \Sigma_i \alpha_{ti} e_i$$

.

Concatenating q_t and c_t together, a fully connected layer finally generates the predicted program token y_t as the output:

$$y_t \sim softmax(W_O[q_t, c_t])$$

The program executor is a collection of deterministic, generic functional modules. Similar to PG + EE, there is a one-to-one correspondence between the program tokens and the functional modules, and the functional modules will be executes sequentially according to the order of the program tokens. The last module's output is the final answer to the question.

Source Code For Johnson and Girshick (2017)'s PG + EE approach and the baseline models,

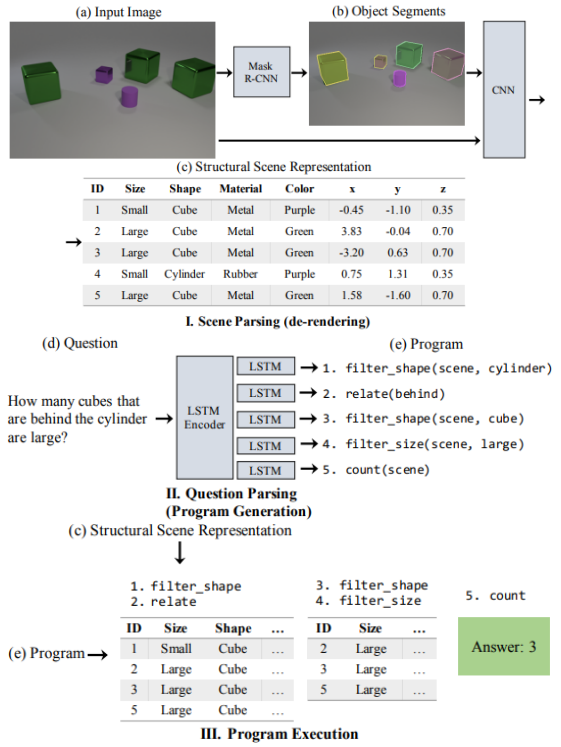

Figure 2: Overview of NS-VQA's architecture

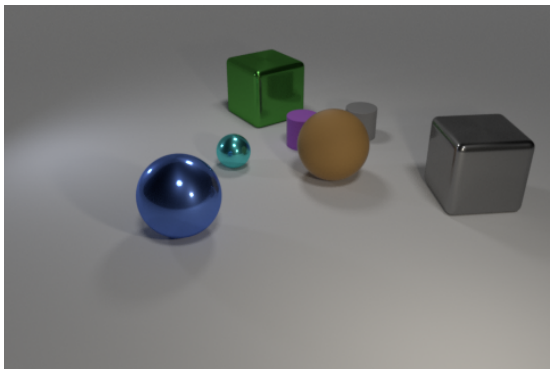

Figure 3: A Sample Image from CLEVR

I adapted the code from https://github.com/facebookresearch/clevr-iep. For Yi et al. (2018)'s approach, I borrowed the code from https://github.com/kexinyi/ns-vqa.

4 Experiments and Results

4.1 Datasets and Metrics

CLEVR Dataset In this paper, evaluations for the VQA models are carried out on the CLEVR dataset Johnson et al. (2017).

CLEVR is a synthetic dataset whose images are generated by randomly sampling a scene graph and rendering it using Blender.

CLEVR has 100,000 images associated with nearly one million questions among which more than 853,000 are unique. The questions are generated by the associated functional programs. The question types are diverse in CLEVR, which allows testing of different types of reasoning.

Split	Images	Questions	Unique questions	Overlap with train
Total	100,000	999,968	853,554	-
Train	70,000	699,989	608,607	-
Val	15,000	149,991	140,448	17,338
Test	15,000	149,988	140,352	17,335

Table 1: Statistics for CLEVR

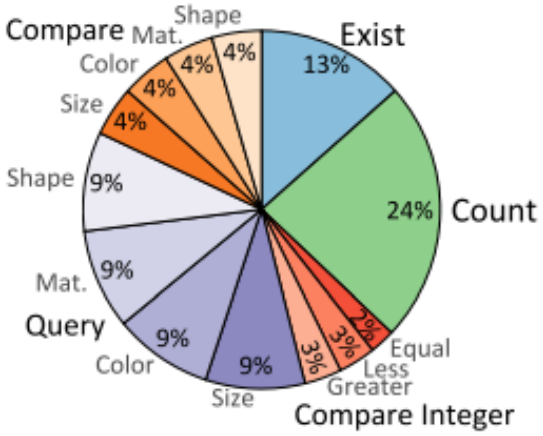

Figure 4: Distribution of Question Types in CLEVR

Figure 4 shows the distribution of question types in CLEVR. In CLEVR, 40% of the questions require the answers "yes" or "no" (*compare attribute*, *compare integer*, and *exist*), and the rest 60% require the facts as the answers (*count* and *query*).

Metrics For quantitative analysis, the accuracy of answers is used as the metric of performance. The accuracy of the answers is defined as the ratio of the number of answers correctly predicted by VQA models to the number of total questions:

$$accuracy = \frac{num_{correctanswers}}{num_{questions}} \times 100\%$$

4.2 Results

Overall Performance Comparison

I compared baseline models' performance with Johnson and Girshick (2017) and Yi et al. (2018)'s approaches on CLEVR's validation dataset. For quantitative analysis, I use the accuracy of answers as the metric.

From the results we can see that Johnson and Girshick (2017)'s model (95.19%)

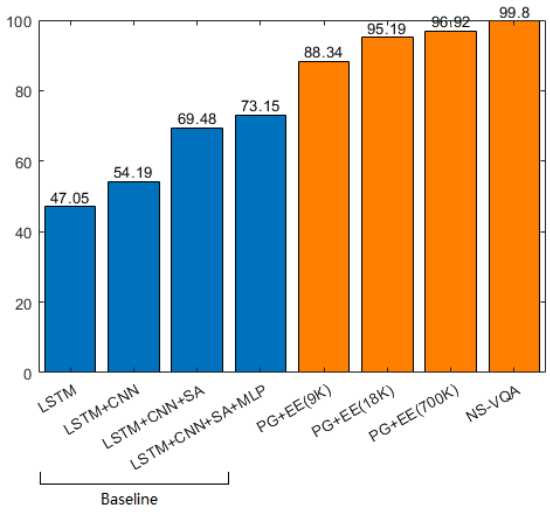

Figure 5: Accuracy of VQA models

	PG+EE(9K)	PG+EE(18K)	PG+EE(700K)
Total	61	26	15
Exist	12	6	3
Count	25	13	9
Compare Integer	8	0	1
Query Attribute	13	6	1
Compare Attribute	3	1	1

Table 2: Breakdown analysis for the failing cases: statistics of the failing cases for 500 samples

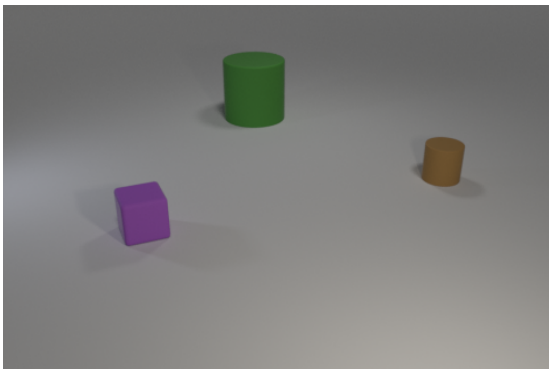

Figure 6: Example of *irrelevant* type of answers
Question: What shape is the brown thing?
Correct Answer: Cylinder
Answer Predicted by PG+EE(18K): No

and Yi et al. (2018)'s model (99.8%) outperform the baseline models (LSTM: 47.05%, LSTM+CNN: 54.19%, LSTM+CNN+SA: 69.48%, LSTM+CNN+SA+MLP: 73.15%) by a very large margin. Meanwhile, the performance difference between Johnson and Girshick (2017) and Yi et al. (2018) is not significant.

Failing Cases by Question Types

In CLEVR, each question has a question type which is defined by the outmost function in the question's program. There are 5 major types of questions: *exist*, *count*, *compare integer*, *query attribute*, and *compare attribute*.

For the 3 PG + EE models (PG+EE(9K), PG+EE(18K), PG+EE(700K)), we sampled 500 question-answer pairs and analyzed the failing cases.

From Table 2 we can see that the performance of the models increase as the number of ground truth programs increase from 9K to 700K. Among all 3 models, *count* type of questions consistently have the highest error-rates, whereas *compare attribute* type of questions tend to have low error-rates. We also notice that for PG+EE models, counting the numbers is much more error-prone than comparing the numbers. It is also true that for PG+EE models, reporting the attributes is somehow harder than comparing the attributes.

Failing Cases by Answers

In this part, I looked into the answers produced by the models in the failing cases with the aim of finding evidence if the models learned reasoning abilities.

One obstacle for my aim is that the answers are quite simple. For *exist*, *compare integer*, and *compare attribute* questions, the answers should be "yes" or "no"; for *count* questions, the answers should be numbers; for *query attribute* questions, the answers should be attributes, such as colors, materials, etc. In most of the cases, we cannot decide if the wrong answer is due to compromised *perception or reasoning* abilities.

However, we still can identify 2 kinds of answers which clearly indicate that the models don't understand the question or don't learn reasoning abilities.

Irrelevant This happens when the answer is supposed to be "yes" or "no", but a number or attribute is predicted as the answer; or the answer is supposed to be a number, but "yes", "no" or an attribute is predicted as the answer; or the answer is supposed to be within a certain category of the attributes (for example, colors), however, "yes", "no", numbers or attributes in other categories (for example, materials) are predicated as the answer.

Unseen This happens for the *query attribute* questions when an attribute that is not presented in the image is predicted as the answer.

Table 3 shows that both PG+EE(9K) and PG+EE(18K) make such mistakes. For PG+EE(700K), we don't find such case in

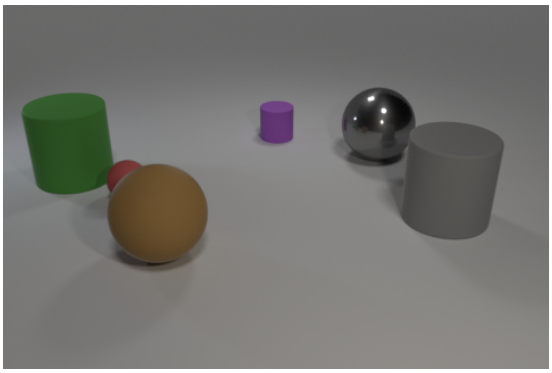

Figure 7: Example of *unseen* type of answers
Question: What color is the small ball?
Correct Answer: Red
Answer Predicted by PG+EE(18K): Cyan
Note that there is no cyan object in the image

	PG+EE(9K)	PG+EE(18K)	PG+EE(700K)
Total	61	26	15
Irrelevant	4	1	0
Unseen	0	1	0

Table 3: Statistics for the wrong answers

our small sample of 500 question-answer pairs. However, we might need large-scale studies to reveal PG+EE(700K)'s performance in this aspect. Overall, we might have to cast doubt on whether PG+EE really learns the reasoning abilities.

4.3 Analysis

Overall Performance

From Figure 5 we can see that Johnson and Girshick (2017) and Yi et al. (2018) outperform the baseline models by a large margin.

It is natural to see that most of the models use the LSTM + CNN architecture to process the images and questions. However, from the experiment results we can see that perception alone is not enough to solve VQA problems, which is why the research community is working towards reasoning abilities.

With the introduction of stacked attention, the framework performs significantly better than previous models, which shows that attention is a step towards reasoning abilities, though it is not adequate for a full solution of VQA problems.

From more recent works' performance in experiments, we can see that the top-performance frameworks have advanced designs over the LSTM + CNN architecture. The advanced models studied in our experiments use program generator + executor frameworks, which proves to be an efficient design for modelling of the reasoning processes.

Do VQA models learn reasoning abilities?

Recent models can undoubtedly capture complex relations and produce better outcomes for VQA tasks, but it is still not clear if the model really learns reasoning abilities.

Based on our experiments, there are evidences that VQA models might fail to reason with the images and question.

Reporting the facts is harder than comparing the facts

In fact, some researchers suggest that human brains have the built-in ability to roughly estimate which is "more" and which is "less" without actually counting them. So, it is not a surprise if neural networks can exhibit similar behaviors (comparing numbers is easier than counting the numbers).

However, if a VQA model really solves the problems by reasoning, it is still supposed to count the numbers first and then compare. So, the question comes: how can the framework correctly compare the attributes/numbers without correctly identify them first, if the framework really complete the tasks by reasoning?

Irrelevant Answers

According to Figure 6, VQA models can make totally irrelevant answers. This seems to suggest that the model totally fails to understand the questions, which makes reasoning impossible.

It will be interesting to examine if the model tend to produce "yes" or "no" as the irrelevant answers. If so, the model might still learns the biases (in CLEVR, 40% of the answers are "yes" or "no").

Mentioning Unseen Attributes

According to Figure 7, VQA models sometimes mention unseen colors in the answers, which leads to the question: where does that unseen attribute come from? Does that come from previous knowledge (that is, bias)?

Perception Problem or Reasoning Problem?

We have mentioned evidences which seem to be not in favor of the hypothesis that VQA models have learned reasoning abilities. However, we should be careful about the interpretation, because we can't rule out the possibility that the errors are caused by perception rather than reasoning. Maybe an irrelevant answer is produced because the LSTM fails to parse the question? Maybe unseen attributes are mentioned because CNN makes wrong classification for its observation? At this time, I don't collect enough evidence to rule out this possibility to reach a decisive conclusion.

Trend Analysis

With the advent of the neural network era, a number of neural-network-based designs were proposed to tackle the visual question answering problems. In the early years, the community witnessed the marvelous success of neural networks which are capable of dealing with previous challenges with unprecedented performance and accuracy. Because of this success, the basic building blocks, such as CNN and RNN, were being applied in a variety of domains, opening up many fast-growing fields of research. In this phenomenal growth of neural-network-related research, we can also see works which proposed to use CNN and RNN for VQA models. It is quite straight-forward that pre-trained CNNs are used to extract features from the images, and word embeddings or RNNs are used to represent questions or answers. Many works also incorporated attention mechanisms over images or questions.

Antol and Parikh (2015) first proposed Visual Question Answering as a free-form and open-ended task. In their work, the model that won the best performance was the so-called "LSTM Q+I", which used the last hidden layer of VGGNet as the 4096 dimensions of image feature representations followed by a linear transformation to match the image features with the LSTM encoding of the questions. To the best of my knowledge, this work mentioned the idea of fusing image and question encodings via element-wise multiplication for the first time.

Different from Antol and Parikh (2015) whose LSTM used one-hot encoding for each of the words in the questions, Zhou and Fergus (2015) proposed to use averaging word vectors to encode the questions. Similar to Antol and Parikh (2015), this "CNN+Bag of Words" approach then uses CNN to extract image features. Then, in Zhou and Fergus (2015) image features and word features are concatenated together and passed to a MLP to generate the distribution over answers.

While most of the works inherited the "CNN + LSTM" framework or its variants, some work took a step further to explore additional mechanisms built upon that. To explore how to combine visual and language features efficiently, Fukui and Rohrbach (2016) and Gao and Darrell (2016) proposed to use Multimodal Compact Bilinear pooling (MCB) to expressively combine multi-modal features. Yang and Smola (2016), on the other hand, proposed to incorporate attention mechanisms, so that the language clues can be redirected to the most relevant part in the image. It was after this first wave of exploration that researchers started to reflect: What abilities are required for VQA tasks? What high-level design is desirable? How to understand each method's limitations? How to interpret poor performance, and make a break-down analysis? How to measure improvements among the methods, given the complex nature of VQA tasks?

One major concern is that some of the methods only present marginal improvements over strong baselines, and it is unclear if the "achievement" is obtained by implicitly utilizing the biases in the datasets. To address this problem, in Johnson et al. (2017) researchers built a synthetic dataset, CLEVR, to control the biases found in prior VQA datasets. This is to make sure that the algorithm doesn't "cheat" by using biases and providing good answers without really solving the reasoning problem. Because of its synthetic nature, CLEVR is able to provide detailed annotations which facilitates in-depth analyses of reasoning abilities. This would be impossible if the dataset is not generated in a controlled manner.

Since the proposal of Johnson et al. (2017), researchers soon came up with new approaches. Some of the methods' performance is so good, that visual reasoning on CLEVR has been considered a nearly-solved problem within the following next few years.

The new approaches contribute to many different high level ideas. While acknowledging early work's contributions to build benchmarks and test performance of many possible combinations of basic architectures, Johnson and Girshick (2017) made the critique that the early works merely used the black-box architectures. Without modeling the underlying reasoning processes explicitly and properly, Johnson and Girshick (2017) argues, the model would be prone to learning biases but not reasoning. To actively offset the tendency to resort to biases in the datasets, Johnson and Girshick (2017) implements a program generator and an execution engine, which to some degree mimics people's cognitive ability to make plans to perform composite tasks by dynamically combining basic cognitive functions. The execution engine implements basic functionalities using small module networks, whereas the program generator reads the questions and learns how to arrange the module

networks by reinforcement learning. Johnson and Girshick (2017) did produce strong performance and it achieved the state-of-the-art when it was published.

However, Johnson and Girshick (2017) is not the only one design philosophy for top-performance VQA systems. Other researchers find out that explicit modeling of the reasoning processes is actually not a key component for good performance on CLEVR. For example, Perez et al. (2018) didn't assume any explicit models for reasoning. It instead embraces a more long standing idea in the community. Antol and Parikh (2015) mentioned the idea of fusing multimodal features by element-wise multiplication. Perez et al. (2018) proposed feature-wise linear modulation, which, to some degree, uses language features to modulate the attention upon image features. Contrary to Johnson and Girshick (2017)'s opinion, Perez et al. (2018) pointed out that we can actually achieve top performance by the CNN + LSTM architecture, if we can find a good design which allows the features to interact efficiently and expressively. Perez et al. (2018) claimed to outperform Johnson and Girshick (2017). In experiments, Perez et al. (2018) proved that FiLM really gains reasoning ability which can generalize well to novel tasks. Perez et al. (2018) also showed that FiLM has great expressive capacity. As a result, Perez et al. (2018) competitively established an alternative approach which can produce equally good performance on CLEVR.

Yet, Yi et al. (2018) and Mao et al. (2019) suggested that certain sort of prior structures or explicit modelling are after all beneficial. Similar to Johnson and Girshick (2017), they proposed to dynamically combine module networks to complete composite tasks, however, different from Johnson and Girshick (2017), they suggest that semantic structures in the questions should guide this dynamic combination process. This proposal makes sense and can reduce the data required for training. Mao et al. (2019) achieves top performance on CLEVR dataset.

So far, we have multiple frameworks with different design philosophies yet achieve comparable performance on CLEVR dataset. Roughly speaking, these frameworks can be divided into 2 categories: Johnson and Girshick (2017) and Yi et al. (2018) use dynamic combination of module networks, which form category I, while Perez et al. (2018) stick to the traditional LSTM+CNN structure, which forms category II.

However, there is no clear-cut boundary between the two categories. Loosely speaking, FiLM's modulation layer can also be seen as a "soft" program. So, to some degree, Johnson and Girshick (2017), Perez et al. (2018), Yi et al. (2018), and Mao et al. (2019)'s work can all be unified in the program generator + executor framework. The difference is that Perez et al. (2018)'s FiLM uses "soft" programs, and it uses ResNet for execution which provides great capacity for the model, wheras in Johnson and Girshick (2017), Yi et al. (2018), and Mao et al. (2019)'s work, the program is hard-coded, and module networks are used for execution, which put more emphasis on interpret-ability and data efficiency. Maybe in the future we will see that different approaches would "merge" with each other to get the best of both worlds.

5 Limitations

In this paper, we compare the overall accuracy of the answers among the VQA models, which only gives us a rough estimation of each model's performance. In order to obtain in-depth understanding of the VQA models, we might need to include breakdown analysis, and ablation studies.

In this paper, comparisons are made only on the synthetic CLEVR datasets. So, we don't know if these VQA models can transfer to other datasets and maintain their performance. It is also unclear if these VQA models can perform well on real-world datasets.

The experiments show that top performance on CLEVR dataset does not necessarily mean that the reasoning problem is solved. So, to evaluate the VQA models on the CLEVR dataset is not adequate to reveal each VQA model's abilities. We might want to focus on solving the reasoning problem, not the performance on certain datasets.

Future Directions

To make a thorough analysis on the CLEVR dataset, we need to include breakdown analysis and ablation studies in future works.

To evaluate VQA models' performance, we need to include tests on multiple datasets in future works. We should include both synthetic and real-world datasets in future studies.

Based on current evidences, we can't draw a decisive conclusion if the VQA models really learn reasoning abilities. To assess VQA models' reasoning abilities, we need to gather more information

about the parse trees and the programs generated by the models. Then we can find out if the errors are due to perception problems or reasoning problems.

One of the main focuses for the VQA community, is to equip the algorithms with reasoning abilities. So, in future works, we might want to find novel ways to assess VQA models' reasoning abilities.

6 Conclusion

Based on the experiment results shown in this paper, we can draw the following conclusions:

First, it requires advanced designs to achieve high performance on the CLEVR dataset. Although most of the models rely on LSTM+CNN for feature extraction and representations, the naive implementation of LSTM+CNN framework is not adequate to cope with the complexity of VQA problems.

Second, for CLEVR dataset, there are multiple VQA models that are nearly of equal competence. Johnson and Girshick (2017)'s PG + EE suggests that explicitly modelling of the reasoning processes is the key component for VQA models to overcome the bias issues. Perez et al. (2018)'s FiLM, on the other hand, doesn't make explicit assumptions. It instead seeks to build an efficient interaction layer for the image and language features, and it achieves equally good performance. Yi et al. (2018)'s NS-VQA uses program generator similar to the PG + EE's approach, however, it assumes that the semantic structures of the questions are vital for successful and data-efficient VQA models. All these approaches have unique insights, and certainly there are trade-offs for each of these methods. Based on their performance on CLEVR dataset, we cannot tell which method is more promising.

Third, the VQA models, such as Johnson and Girshick (2017)'s PG + EE, still doesn't fully understand the questions and images. There are cases in which PG + EE mentions unseen objects/attributes in its answers, and in some cases PG + EE might produce totally irrelevant answers. These might suggest that PG + EE still use biases in the dataset, and it might still lack the real reasoning abilities. This observation might suggest that there is a gap between VQA algorithms' performance on the datasets and their true reasoning abilities. So, learning to reason is still a mission not completed for the VQA community, and it might require more insightful designs in the VQA algorithms. Since CLEVR dataset cannot fully reveal VQA algorithms' drawbacks, more advanced designs for VQA datasets are required to assess VQA algorithms' reasoning abilities.

7 Contributions

Sihui Wang is the sole author of this work. He is responsible for setting up the virtual environment to run the codes, fixing bugs and adapting codes to settle version conflicts and incompatibilities, doing the experiments, collecting the results, carrying out the analysis, and completing the final write-up.

References

Marcus Rohrbach Trevor Darrell Andreas, Jacob and Dan Klein. 2016. Neural module networks.

Aishwarya Agrawal Jiasen Lu Margaret Mitchell Dhruv Batra C. Lawrence Zitnick Antol, Stanislaw and Devi Parikh. 2015. Vqa: Visual question answering.

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

Oscar Beijbom Ning Zhang Gao, Yang and Trevor Darrell. 2016. Compact bilinear pooling.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory.

Bharath Hariharan Laurens Van Der Maaten Judy Hoffman Li Fei-Fei C. Lawrence Zitnick Johnson, Justin and Ross Girshick. 2017. Inferring and executing programs for visual reasoning.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Fei-Fei Li, C.Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. 2019. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision.

Jiayuan Mao, Haoyue Shi, Jiajun Wu, Roger P. Levy, and Joshua B. Tenenbaum. 2021. Grammar-based grounded lexicon learning.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. 2018. Film: Visual reasoning with a general conditioning layer.

Xiaodong He Jianfeng Gao Li Deng Yang, Zichao and Alex Smola. 2016. Stacked attention networks for image question answering.

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

Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. 2021. Cross-modal contrastive learning for text-to-image generation.

Yuandong Tian Sainbayar Sukhbaatar Arthur Szlam Zhou, Bolei and Rob Fergus. 2015. Simple baseline for visual question answering.