Abstract

Achieving artificial visual reasoning — the ability to answer image-related questions which require a multi-step, high-level process — is an important step towards artificial general intelligence. We introduce a FiBN model which is based on the FiLM model. We show how replacing the normal BN layer in the model with a CBN layer affects the performance and spatial reasoning of the model. We have tested various configurations and hyperparameters to see the effect of the CBN on the FiLM model, then we show how: different batch size, dropout percentage, model depth and the removal of both CBN and FiLM - affect the model. Lastly, we compare the spatial reasoning of the original FiLM and our model. According to the findings our model has shown similar results to FiLM and we believe it has a potential to provide better results with further testing.

1. Introduction

The ability to use language to reason about every-day visual input is a fundamental building block of human intelligence. Achieving this capacity to visually reason is thus a meaningful step towards artificial agents that truly understand the world (Perez et al., 2017).

A model which is made from general-purpose components and can learn to visually reason, will likely be more widely applicable across domains (Perez et al., 2018).

One of the ways to evaluate such a model is by using a diagnostic dataset for Compositional Language and Elementary Visual Reasoning (CLEVR), which is used to test visual reasoning via question answering (Johnson et al., 2017).

Visual question answering is a general task of asking questions about images, has its own line of datasets which generally focus on asking a diverse set of simple questions on images, often answerable in a single glance. From these datasets, several effective general-purpose deep learning models have emerged for visual question answering (Anderson et al., 2017; Lu, Yang, Batra, & Parikh, 2016; Malinowski, Rohrbach, & Fritz, 2015; Yang, He, Gao, Deng, & Smola, 2016). However, tests on CLEVR show that these general deep learning approaches struggle to learn structured, multi-step reasoning (Johnson et al., 2017).

These models tend to exploit biases in the data rather than capture complex underlying structure behind reasoning (Goyal et al., 2017).In order to overcome this problem, Perez and his colleagues (2018) developed a general model architecture that can achieve strong visual reasoning which they termed as FiLM: Feature-wise Linear Modulation.

Film is a general-purpose conditioning method that is highly effective for visual reasoning. However, one of its drawbacks is that it makes some logical mistakes that humans won't do, for example: a case where FiLM model correctly counts one gray object and two cyan objects but simultaneously answers that there are the same number of gray and cyan objects. In fact, it answers that the number of gray objects is both less than and equal to the number of yellow blocks (Perez et al., 2018).

In this project we want to observe whether adding a CBN layer, which has proven highly effective for traditional visual question answering tasks (De Vries et al., 2017) without exploiting biases to a FiLM model, can improve the performance and solve the above mentioned FiLM's drawback.

2. Method and Implementation

Our model processes the multi-modal question-image input using a RNN and CNN combined via FiLM and Conditional Batch Normalization (CBN).
Firstly, we will start by explaining FiLM and CBN and next in order we will describe our model with its modifications and additions.

2.1 FiLM: Feature-wise Linear Modulation

FiLM learns to adaptively influence the output of a neural network by applying an affine transformation, to the network's intermediate features, based on some input. More formally, FiLM learns functions f and h which output
$\gamma_{i,c}$ and $\beta_{i,c}$ as a function of input x_i:

(1) $\gamma_{i,c} = f_c(x_i) \qquad \beta_{i,c} = h_c(x_i)$

where $\gamma_{i,c}$ and $\beta_{i,c}$ modulate a neural network's activations $F_{i,c}$ whose subscripts refer to the i^{th} input's c^{th} feature or feature map, via a feature-wise affine transformation:

(2) $FiLM\left(F_{i,c}|\gamma_{i,c}, \beta_{i,c}\right) = \gamma_{i,c}F_{i,c} + \beta_{i,c}$

f and h can be arbitrary functions such as neural networks.
(Perez et al., 2018)

2.2 CBN: Conditional Batch Normalization

BN has been shown to accelerate training and improve generalization by reducing covariate shift throughout the network (Watters et al., 2017). To explain BN, we define $B = \{F_i, ., ., .\}_{i=1}^{N}$ as a mini batch of N samples, where F corresponds to input feature maps whose subscripts c, h, w refers to the c^{th} feature map at the spatial location (h, w). We also define $\gamma_{i,c}$ and $\beta_{i,c}$ as per-channel, trainable scalars and ϵ as a constant damping factor for numerical stability.
BN is defined at training time as follows:

(1) $BN(F_{i,c,h,w}|\gamma_c, \beta_c) = \gamma_c \frac{F_{i,c,w,h} - \mathbb{E}_B[F_{\cdot,c,\cdot,\cdot}]}{\sqrt{Var_B[F_{\cdot,c,\cdot,\cdot}] + \epsilon}} + \beta_c$

Conditional Batch Normalization (CBN) [14, 15, 16] instead learns to output new BN parameters $\hat{\gamma}_{i,c}$ and $\hat{\beta}_{i,c}$ as a function of some input x_i:

(2) $\hat{\gamma}_{i,c} = f_c(x_i) \qquad \hat{\beta}_{i,c} = h_c(x_i)$

where f and h are arbitrary functions such as neural networks.
(Perez et al., 2017)

2.3 Our Model

Our model consists of a linguistic pipeline and a visual pipeline as depicted in Figure 1. The linguistic pipeline processes a question q using a Gated Recurrent Unit (GRU) (Chung, Gulcehre, Cho, & Bengio, 2014) with 4096 hidden units that takes in learned, 200-dimensional word embeddings. The final GRU hidden state is a question embedding, from which the model predicts $\left(\gamma_{i,\cdot}^{n}, \beta_{i,\cdot}^{n}\right)$ for each n^{th} residual block via affine projection, we have doubled the amount of weights the GRU provides so they can be used for both FiLM and CBN. We also wrapped the GRU model with Pytorch's Parallel model to increase processing speed. The visual pipeline extracts 128 14 x 14 image feature maps from a resized, 224 x 224 image input using either a CNN trained from scratch or a fixed, pre-trained feature extractor with a learned layer of 3 x 3 convolutions. The CNN trained from scratch consists of 4 layers with 128 4 x 4 kernels each, ReLU activations, conditional batch normalization and dropout. The classifier is implemented as was presented by Perez and his colleagues (2018). Furthermore, we had to do several necessary adjustments because of server's limitations. The server wouldn't let us run the model for more than 24hrs. Consequently, we had to adjust the dataset, in such a way, that we had reduced the data type to int32 down from int64, used only half of the dataset instead of all of it and ran only for 1% of the original number of iterations.

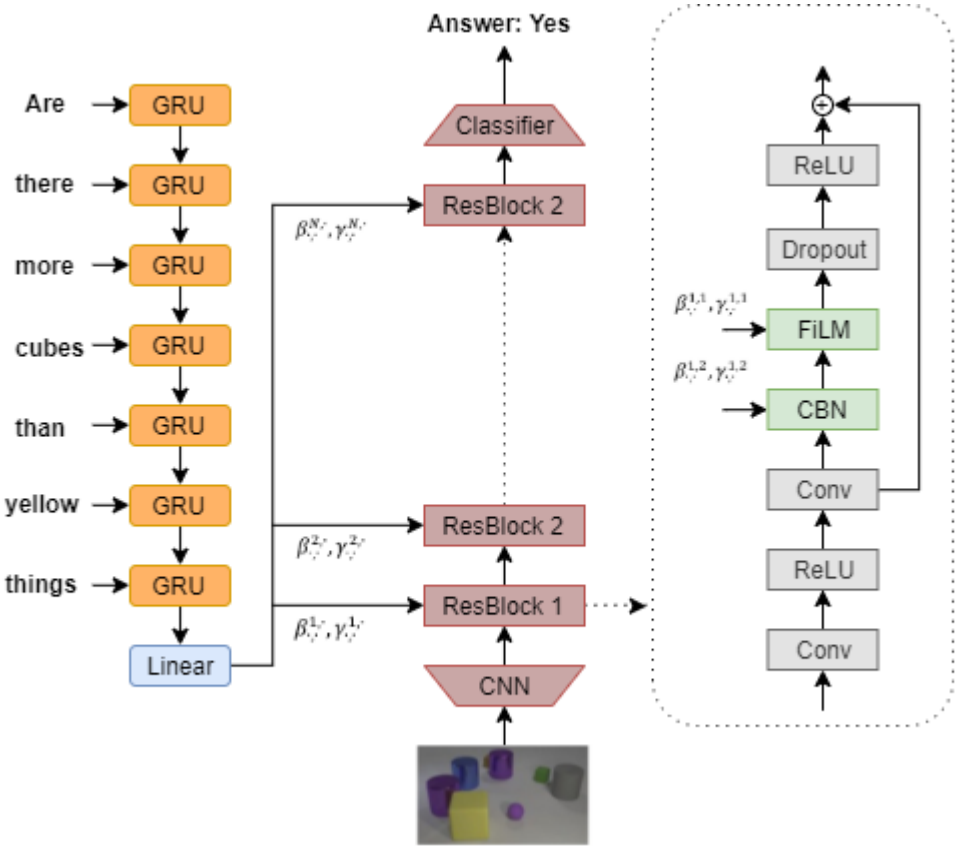
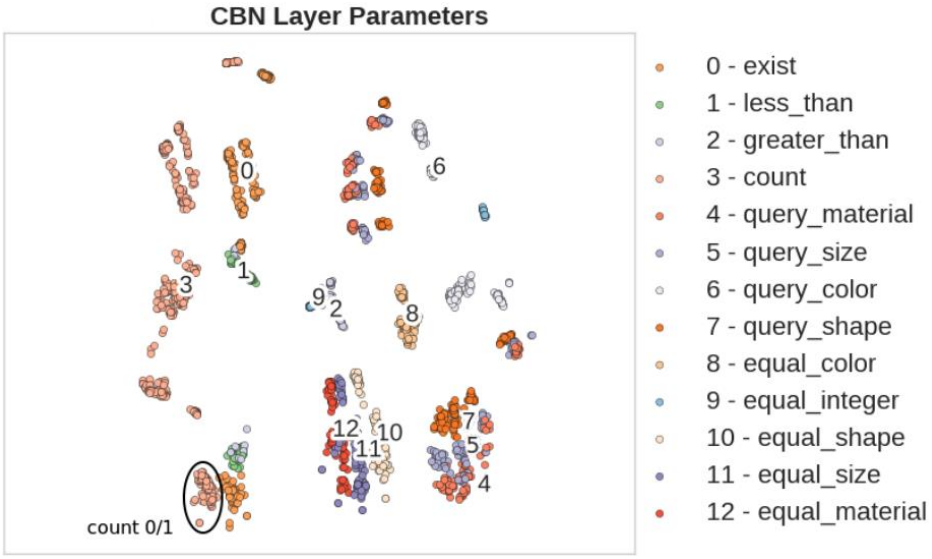
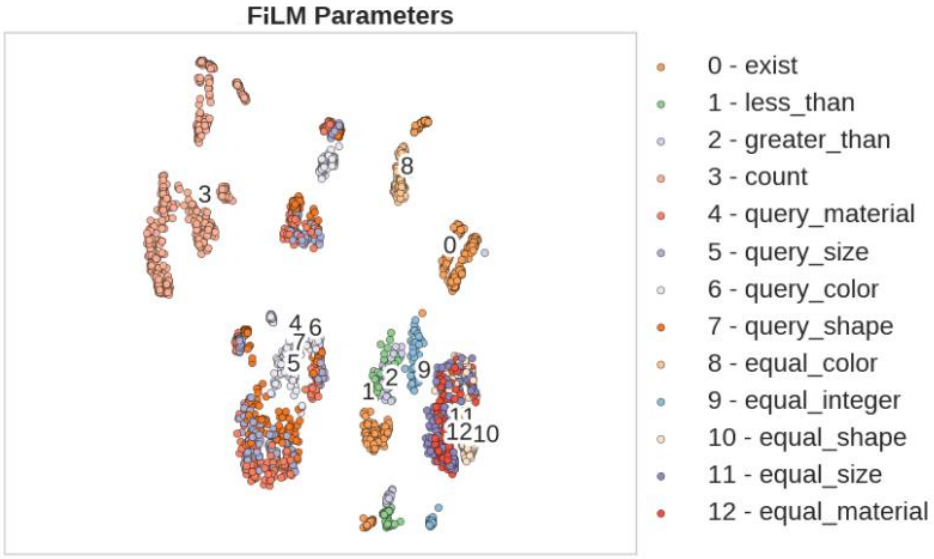


Figure 1: The linguistic pipeline (left), visual pipeline (middle), and residual block architecture (right) of our model.

2.4 Theoretical Motivation

Both FiLM and CBN have comparable performances on the CLEVR dataset. However, each of them is slightly better than the other in different questions, for example FiLM has better accuracy with comparing questions but on the other hand, CBN is better for counting. We believe the accuracy difference is caused by the difference in spatial reasoning of CBN and FiLM.

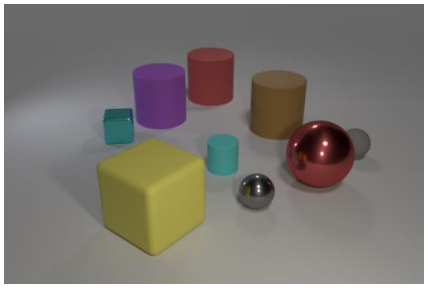




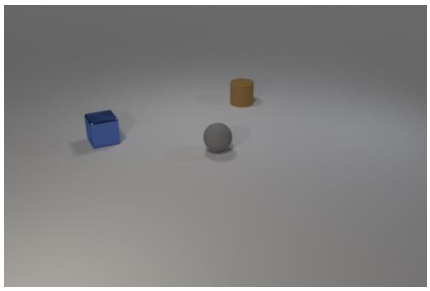
From that, we assume that the combination of the two layers might result in a different spatial reasoning all together and as a result there will be an improvement in performances.

2.5 CLEVR dataset

CLEVR presents a diagnostic dataset that tests a range of visual reasoning abilities. It contains minimal biases and has detailed annotations describing the kind of reasoning each question requires and can be used to analyze a variety of modern visual reasoning systems, providing novel insights into their abilities and limitations (Johnson et al., 2017).
It is a generated dataset of 700K (image, question, answer, program) tuples. Images contain 3D-rendered objects of various shapes, materials, colors, and sizes. Questions are multi-step and compositional in nature, as shown in Figure 2. They range from counting questions ("How many green objects have the same size as the green metallic block?") to comparison questions ("Are there fewer tiny yellow cylinders than yellow metal cubes?") and can be 40+ words long. Answers are each one word from a set of 28 possible answers (Perez et al., 2017).



(a) Q: Is there a ball made of the same material as the tiny cyan cube?
A: Yes



(b) Q: How many brown rubber objects are the same shape as the gray rubber object
A: 0

3 Experiments

We have tested our model with various tests to see how the addition of a CBN layer affects the model. The loss formulas that was used is the same as presented by Perez and his colleagues (2018). We have preprocessed the CLEVR pictures, for the images we have extracted ResNet-101 features. As for the questions, we have created a vocabulary file and encoded all questions and programs. The data preprocessing that was used is the same as was used by Perez and his colleagues (2018).

3.1 Reducing model's overfit

In order to reduce the model's overfit, we have tried different batch sizes and different dropout percentages.

4.1.1 Batch size test

We have used the same model architecture as described at section 2.3. The batch sizes that were tested are 64, 96, 128, 256.

4.1.2 Dropout percentage test

We have used the same model architecture as described at section 2.3 and the best batch size from batch size test. We have tested dropout of 0, 3, 20, 50, 80. The batch size that was used is 96.

3.2 Changing model depth

We checked our model performance as a function of the number of ResBlocks in the model. At this test we only changed the model depth. Each block has the same architecture as described in section 2.3, the batch size and dropout that were used are 96 and 0 respectively. The number of ResBlocks that we tested are 2, 3, 4, 5, 6.

All the tests that are mentioned above are aimed to determine the best architecture and hyperparameters for our model. In each of them we have compared the train accuracy and validation accuracy between the different configurations and chose the configuration that yields the lowest overfit and highest accuracy.

3.3 Removing FiLM and CBN from ResBlock

We wanted to check how repetitive use of FiLM and CBN(FiBN) layer affects the model. In order to do that we have tested the model while removing the FiBN layer from the ResBlocks. The model architecture we used is the same architecture as presented in section 2.3. The hyperparameters that were used are batch size 96, dropout 0 and 3 ResBlocks. The tests include removal of FiBN from ResBlock number 3, Resblock number 2-3, Resblock number 1-3 and no removal at all.

This test purpose was to check how repetitive use of FiBN affects the model and not to check which architecture is best for the model. Hence the results were not used to determine the best model.

4. Results

All the tests were performed under the server's limitation. Hence, the model was running for 24hrs on every configuration and the results are based on 20hrs of training.

4.1 Batch size test

We tested which batch size will provide the best accuracy and the lowest overfit, the results are presented in the table below.

Batch size	Train accuracy	Validation accuracy
64	96.8	81.7
96	98.8	83.6
128	97.5	80.9
256	98.2	78.8

We can see that for batch size 96 the train accuracy is the highest and the validation is the highest which makes batch size 96 the best option for our model. The test was performed on our model only without comparing to the original FiLM model. The purpose of this test was to reduce the model overfit without hurting the performance as much

4.2 Dropout percentage test

We tested which dropout percentage will provide the best accuracy and the lowest overfit with the batch size that was provided from batch size test, the results are presented in the table below.

Dropout percentage	Train accuracy	Validation Accuracy
0	98.8	83.6
3	97.3	79.5
20	92.8	75
50	66.1	49.1
80	51.4	48.8

From the table above it is clear that using no dropout at all provides the highest accuracy and lowest overfit, 80% dropout provides less overfit, but the accuracy is too low, hence it hasn't been considered. We aim to reduce the overfit of the model without hurting the performance as much.

4.3 Changing model depth

We have tested which model depth will provide the best accuracy and the lowest overfit. We have used batch size and dropout percentage based on the previous tests we have performed.
The results are presented in the table below:

Model Depth (number of ResBlocks)	Train accuracy	Validation Accuracy
2	98	83.8
3	98.4	84.1
4	98.8	83.6
5	97.8	83.4
6	97.5	81

Both depth 2 and 3 yield the same overfit percentage, but depth 3 has higher train accuracy and higher validation accuracy, hence it was chosen for our model. Unlike FiLM, the FiBN model achieves better results with less than 4 ResBlocks as a result of the increase in dependency (caused by adding the CBN layer) between the output of the GRU and the images.

4.4 Removing FiLM and CBN from ResBlock

We have tested how the combination of FiLM and CBN (FiBN) affects the model in term of train accuracy and validation accuracy. We have used batch size, dropout percentage and depth based on the previous tests we have performed.
The results are presented in the table below:

ResBlock	Train accuracy	Validation Accuracy
No removal	98.4	84.1
1 to 3	27	19
2 to 3	97.9	84.5
3	98	85.5

From the results that are mentioned above, we can say that even a single FiBN combination, does not deviate far from the best model's performance, revealing that the model can reason and answer diverse questions successfully by modulating features even just once much like the FiLM model. However, more than 2 combinations of FiLM and CBN cause an increase in overfitting of roughly 1.5%, which is caused by the increased in dependency between the GRU weights and the images.

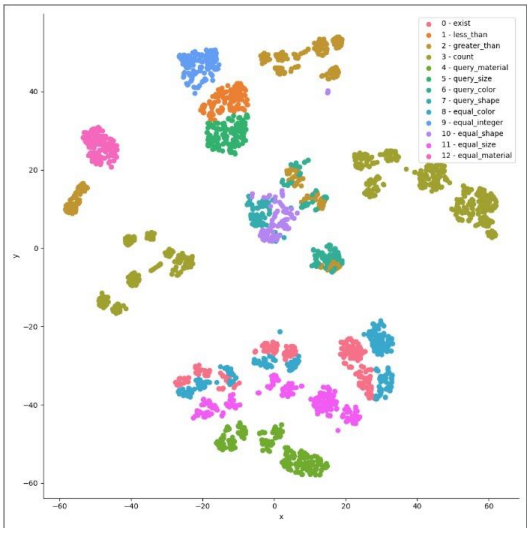
Lastly, after performing all the tests we have compared our model to the original FiLM model. The FiLM model ran under the same limitation as our model in term of data type size, dataset size and training time. Our model was using batch size 96, no dropout, 3 ResBlocks and we haven't removed FiBN from any ResBlock. Those parameters and architecture details were taken from the tests mentioned above.
The results are presented in the table below:

Model	Train accuracy	Validation Accuracy
FiBN	98.4	84.1
FiLM	98	89.77

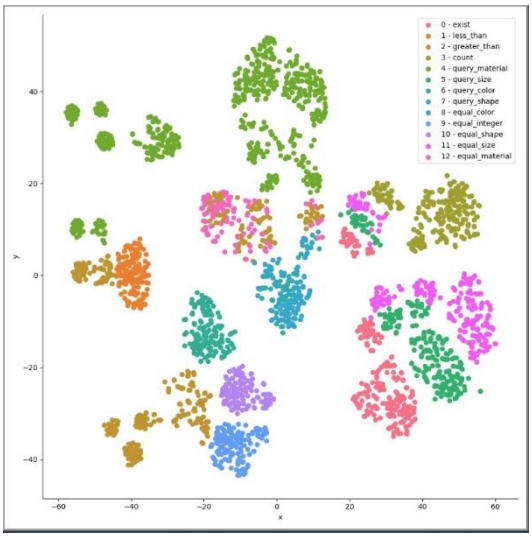
Our FiBN model achieve higher train accuracy than the FiLM model but less validation accuracy. The addition of CBN layer instead of the normal BN causes the model to be more overfitted, thus results in lower validation accuracy. Due to lack of time, we couldn't train the model properly. However, we believe that it might be possible to achieve better results with our model by doing more architecture and hyperparameters tuning and by increasing the dataset size but.

In addition, we have used t-SNE to visualize our model parameter vectors (γ, β) and compared them to the FiLM model's vectors. As one of our goals was to achieve a different spatial reasoning which might solve FiLM's problem that we mentioned. They are both presented in the pictures below:

FiLM Last ResBlock



FiBN Last ResBlock

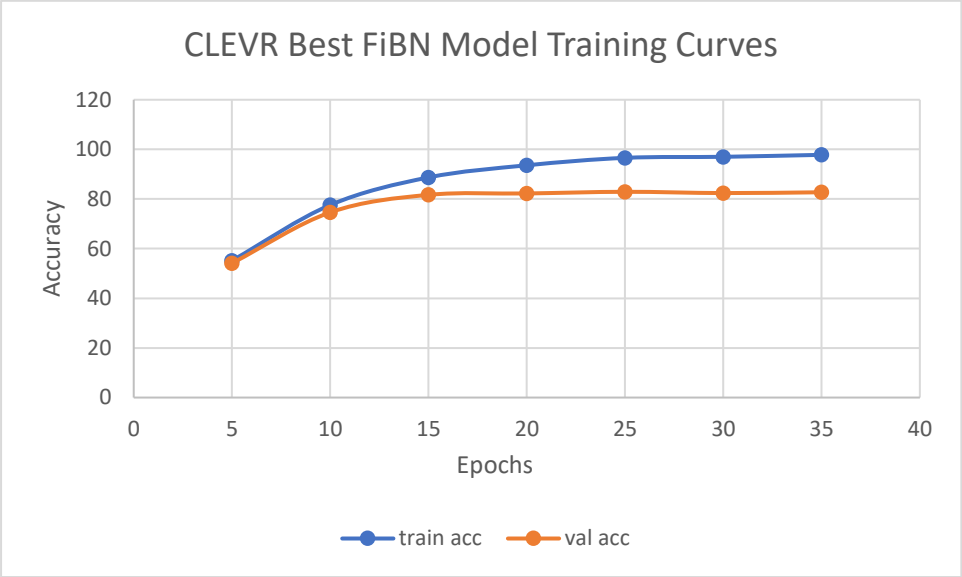


*Parameter vectors (γ, β) for 3000 random samples from the validation dataset

From the scatter images we can gather that FiLM and FiBN reason differently about the input. For example, the parameters for query_shape and equal_shape are very close to one another in FilM but they are far away in FiBN, and the opposite is true for query_size and equal_size.

5. Conclusion

Our model has similar architecture as the Original FiLM model. However, we replaced the normal BN with CBN. We came to know that our model is overfitted due to the fact we have both FiLM and CBN updating the weights that the GRU uses, consequently the model is more dependent on the train data. Despite the various limitations we had to address during training, the FiBN model achieved good results. We believe that training the model for more than 24hrs and with the full dataset will solve the overfit problem and might surpass the FiLM model.



References

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2017). Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 6077–6086. https://doi.org/10.1109/CVPR.2018.00636

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 1–9. Retrieved from http://arxiv.org/abs/1412.3555

De Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O., & Courville, A. (2017). Modulating early visual processing by language. *Advances in Neural Information Processing Systems*, *2017-Decem*, 6595–6605.

Goyal, Y., Khot, T., Agrawal, A., Summers-Stay, D., Batra, D., & Parikh, D. (2017). Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. *International Journal of Computer Vision*, *127*(4), 398–414. https://doi.org/10.1007/s11263-018-1116-0

Johnson, J., Fei-Fei, L., Hariharan, B., Zitnick, C. L., Van Der Maaten, L., & Girshick, R. (2017). CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 1988–1997. https://doi.org/10.1109/CVPR.2017.215

Lu, J., Yang, J., Batra, D., & Parikh, D. (2016). Hierarchical question-image co-attention for visual question answering. *Advances in Neural Information Processing Systems*, (c), 289–297.

Malinowski, M., Rohrbach, M., & Fritz, M. (2015). Ask your neurons: A neural-based approach to answering questions about images. *Proceedings of the IEEE International Conference on Computer Vision*, *2015 Inter*, 1–9. https://doi.org/10.1109/ICCV.2015.9

Perez, E., de Vries, H., Strub, F., Dumoulin, V., & Courville, A. (2017). Learning Visual Reasoning Without Strong Priors. Retrieved from http://arxiv.org/abs/1707.03017

Perez, E., Strub, F., De Vries, H., Dumoulin, V., & Courville, A. (2018). FiLM: Visual reasoning with a general conditioning layer. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 3942–3951. Retrieved from https://arxiv.org/abs/1709.07871

Watters, N., Tacchetti, A., Weber, T., Pascanu, R., Battaglia, P., & Zoran, D. (2017). *Visual Interaction Networks*. 1–14. Retrieved from http://arxiv.org/abs/1706.01433

Yang, Z., He, X., Gao, J., Deng, L., & Smola, A. (2016). Stacked attention networks for image question answering. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*(1), 21–29. https://doi.org/10.1109/CVPR.2016.10