

# Learning Visual Reasoning Without Strong Priors

Ethan Perez<sup>1,2</sup>, Harm de Vries<sup>1</sup>, Florian Strub<sup>3</sup>,  
Vincent Dumoulin<sup>1</sup>, Aaron Courville<sup>1,4</sup>

<sup>1</sup>MILA, Université de Montréal, Canada; <sup>2</sup>Rice University, U.S.A.  
<sup>3</sup>Univ. Lille, CNRS, Centrale Lille, Inria, UMR 9189 CRISAL France  
<sup>4</sup>CIFAR Fellow, Canada

ethanperez@rice.edu, mail@harmdevries.com, florian.strub@inria.fr  
dumouliv@iro.umontreal.ca, courvila@iro.umontreal.ca

## Abstract

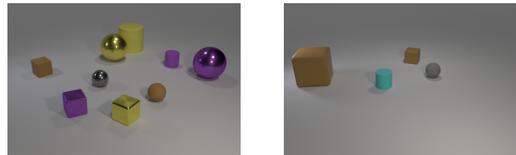
Achieving artificial visual reasoning — the ability to answer image-related questions which require a multi-step, high-level process — is an important step towards artificial general intelligence. This multi-modal task requires learning a question-dependent, structured reasoning process over images from language. Standard deep learning approaches tend to exploit biases in the data rather than learn this underlying structure, while leading methods learn to visually reason successfully but are hand-crafted for reasoning. We show that a general-purpose, Conditional Batch Normalization approach achieves state-of-the-art results on the CLEVR Visual Reasoning benchmark with a 2.4% error rate. We outperform the next best end-to-end method (4.5%) and even methods that use extra supervision (3.1%). We probe our model to shed light on how it reasons, showing it has learned a question-dependent, multi-step process. Previous work has operated under the assumption that visual reasoning calls for a specialized architecture, but we show that a general architecture with proper conditioning can learn to visually reason effectively.

**Index Terms:** visual reasoning, visual question answering, multi-modal data, deep learning

## 1. Introduction

The ability to use language to reason about every-day visual input is a fundamental building block to human intelligence. Achieving this capacity to visually reason is thus a meaningful step towards artificial agents that truly understand the world. Advances in both image-based learning and language-based learning using deep neural networks have made huge strides in difficult tasks such as object recognition [1, 2] and machine translation [3, 4]. These advances have in turn fueled research on the intersection of visual and linguistic learning [5, 6, 7, 8, 9].

To this end, [9] recently proposed the CLEVR dataset to test multi-step reasoning from language about images, as traditional visual question-answering datasets such as [5, 7] ask simpler questions on images that can often be answered in a single glance. Examples from CLEVR are shown in Figure 1. Structured, multi-step reasoning is quite difficult for standard deep learning approaches [10, 11], including those successful on traditional visual question answering datasets. Previous work highlights that standard deep learning approaches tend to exploit biases in the data rather than reason [9, 12]. To overcome this, recent efforts have built new learning architectures that explicitly model reasoning or relational associations [10, 13, 11], some of which even outperform humans [10, 11].



(a) *What number of cylinders are small purple things or yellow rubber things?* (b) *What color is the other object that is the same shape as the large brown matte thing?*

**Predicted: 2**

**Predicted: Brown**

Figure 1: Examples from CLEVR and our model’s answer.

In this paper, we show that a general model can achieve strong visual reasoning from language. We use Conditional Batch Normalization [14, 15, 16] with a Recurrent Neural Network (RNN) and a Convolutional Neural Network (CNN) to show that deep learning architectures built without strong priors can learn underlying structure behind visual reasoning, directly from language and images. We demonstrate this by achieving state-of-the-art visual reasoning on CLEVR and finding structured patterns while exploring the internals of our model.

## 2. Method

Our model processes the multi-modal question-image input using a RNN and CNN combined via Conditional Batch Normalization (CBN). CBN has proven highly effective for image stylization [14, 16] and traditional visual question answering tasks [15]. We start by explaining CBN in Section 2.1 and then describe our model in Section 2.2.

### 2.1. Conditional batch normalization

Batch normalization (BN) is a widely used technique to improve neural network training by normalizing activations throughout the network with respect to each mini-batch. BN has been shown to accelerate training and improve generalization by reducing covariate shift throughout the network [17]. To explain BN, we define  $\mathcal{B} = \{\mathbf{F}_i, \dots\}_{i=1}^N$  as a mini-batch of  $N$  samples, where  $\mathbf{F}$  corresponds to input feature maps whose subscripts  $c, h, w$  refers to the  $c^{\text{th}}$  feature map at the spatial location  $(h, w)$ . We also define  $\gamma_c$  and  $\beta_c$  as per-channel, trainable scalars and  $\epsilon$  as a constant damping factor for numerical stabil-

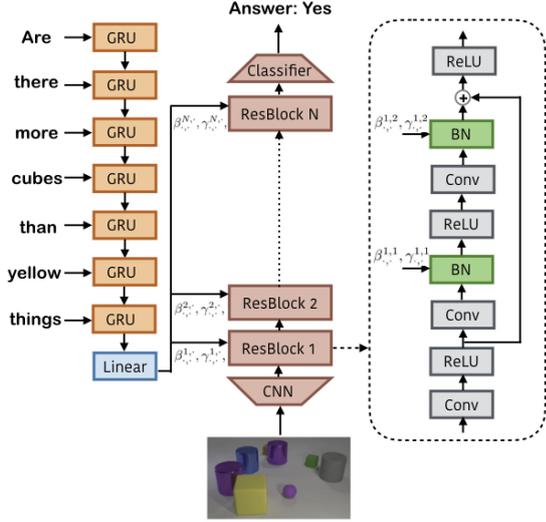


Figure 2: The linguistic pipeline (left), visual pipeline (middle), and CBN residual block architecture (right) of our model.

ity. BN is defined at training time as follows:

$$BN(\mathbf{F}_{i,c,h,w} | \gamma_c, \beta_c) = \gamma_c \frac{\mathbf{F}_{i,c,h,w} - \mathbb{E}_{\mathcal{B}}[\mathbf{F}_{i,c,h,w}]}{\sqrt{\text{Var}_{\mathcal{B}}[\mathbf{F}_{i,c,h,w}] + \epsilon}} + \beta_c. \quad (1)$$

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  $\mathbf{x}_i$ :

$$\hat{\gamma}_{i,c} = f_c(\mathbf{x}_i) \quad \hat{\beta}_{i,c} = h_c(\mathbf{x}_i), \quad (2)$$

where  $f$  and  $h$  are arbitrary functions such as neural networks. Thus,  $f$  and  $h$  can learn to control the distribution of CNN activations based on  $\mathbf{x}_i$ .

Combined with ReLU non-linearities, CBN empowers a conditioning model to manipulate feature maps of a target CNN by scaling them up or down, negating them, shutting them off, selectively thresholding them, and more. Each feature map is modulated independently, giving the conditioning model an exponential (in the number of feature maps) number of ways to affect the feature representation.

Rather than output  $\hat{\gamma}_{i,c}$  directly, we output  $\Delta\hat{\gamma}_{i,c}$ , where:

$$\hat{\gamma}_{i,c} = 1 + \Delta\hat{\gamma}_{i,c}, \quad (3)$$

since initially zero-centered  $\hat{\gamma}_{i,c}$  can zero out CNN feature map activations and thus gradients. In our implementation, we opt to output  $\Delta\hat{\gamma}_{i,c}$  rather than  $\hat{\gamma}_{i,c}$ , but for simplicity, in the rest of this paper, we will explain our method using  $\hat{\gamma}_{i,c}$ .

## 2.2. Model

Our model consists of a linguistic pipeline and a visual pipeline as depicted in Figure 2. The linguistic pipeline processes a question  $q$  using a Gated Recurrent Unit (GRU) [18] with 4096 hidden units that takes in learned, 200-dimensional word embeddings. The final GRU hidden state is a question embedding  $e_q$ . From this embedding, the model predicts the CBN parameters  $(\gamma_{i,c}^{m,n}, \beta_{i,c}^{m,n})$  for the  $n^{\text{th}}$  CBN layer of the  $m^{\text{th}}$  residual block via linear projection with a trainable weight matrix  $\mathbf{W}$  and bias vector  $\mathbf{b}$ :

$$(\gamma_{i,c}^{m,n}, \beta_{i,c}^{m,n}) = \mathbf{W}^{m,n} e_q + \mathbf{b}^{m,n} \quad (4)$$

The visual pipeline extracts  $14 \times 14$  image features using the *conv4* layer of a ResNet-101 [2] pre-trained on ImageNet [19], as done in [10] for CLEVR. Image features are processed by a  $3 \times 3$  convolution followed by several — 3 for our model — CBN residual blocks with 128 feature maps, and a final classifier. The classifier consists of a  $1 \times 1$  convolution to 512 feature maps, global max-pooling, and a two-layer MLP with 1024 hidden units that outputs a distribution over final answers.

Each CBN residual block starts with a  $1 \times 1$  convolution followed by two  $3 \times 3$  convolutions with CBN as depicted in Figure 2. Drawing from [11, 20], we concatenate coordinate feature maps indicating relative spatial position (scaled from  $-1$  to  $1$ ) to each residual block’s input. We train our model end-to-end from scratch with Adam (learning rate  $3e^{-4}$ ) [21], early stopping on the validation set, weight decay ( $1e^{-5}$ ), batch size 64, and BN and ReLU throughout the visual pipeline, using only image-question-answer triplets from the training set.

## 3. Experiments

### 3.1. CLEVR dataset

CLEVR is a generated dataset consisting 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 illustrated in Figure 1. They range from counting questions such as “How many green objects have the same size as the green metallic block?” to comparison questions such as “Are there fewer tiny yellow cylinders than yellow metal cubes?” Answers are each one word from a set of 28 possible answers. Programs are an additional supervisory signal consisting of step-by-step instructions, such as `filter_shape[cube]`, `relate[right]`, and `count`, on how to answer the question. Program labels are difficult to generate or come by for real world datasets. Our model avoids using this extra supervision, learning to reason effectively directly from linguistic and visual input.

### 3.2. Results

Our results on CLEVR are shown in Table 1. Our model achieves a new overall state-of-the-art, outperforming humans and previous, leading models, which often use additional program supervision. Notably, CBN outperforms Stacked Attention networks (CNN+LSTM+SA in 1) by 21.0%. Stacked Attention networks are highly effective for visual question answering with simpler questions [22] and are the previously leading model for visual reasoning that does not build in reasoning, making them a relevant baseline for CBN. We note also that our model’s pattern of performance more closely resembles that of humans than other models do. Strong performance ( $< 1\%$  error) in `exist` and `query_attribute` categories is perhaps explained by our model’s close resemblance to standard CNNs, which traditionally excel at these classification-type tasks. Our model also demonstrates strong performance on more complex categories such as `count` and `compare_attribute`.

Comparing numbers of objects gives our model more difficulty, understandably so; this question type requires more high-level reasoning steps — querying attributes, counting, and comparing — than other question type. The best model from [10] beats our model here but is trained with extra supervision via 700K program labels. As shown in Table 1, the equivalent, more comparable model from [10] which uses 9K program labels significantly underperforms our method in this category.

Model	Overall	Count	Exist	Compare Numbers	Query Attribute	Compare Attribute
Human [10]	92.6	86.7	96.6	86.5	95.0	96.0
Q-type baseline [10]	41.8	34.6	50.2	51.0	36.0	51.3
LSTM [10]	46.8	41.7	61.1	69.8	36.8	51.8
CNN+LSTM [10]	52.3	43.7	65.2	67.1	49.3	53.0
CNN+LSTM+SA [11]	76.6	64.4	82.7	77.4	82.6	75.4
N2NMN* [13]	83.7	68.5	85.7	84.9	90.0	88.7
PG+EE (9K prog.)* [10]	88.6	79.7	89.7	79.1	92.6	96.0
PG+EE (700K prog.)* [10]	96.9	92.7	97.1	<b>98.7</b>	98.1	98.9
CNN+LSTM+RN† [11]	95.5	90.1	97.8	93.6	97.9	97.1
CNN+GRU+CBN	<b>97.6</b>	<b>94.5</b>	<b>99.2</b>	93.8	<b>99.2</b>	<b>99.0</b>

Table 1: CLEVR accuracy by baseline methods, competing methods, and our method (CBN). Methods denoted with (\*) use extra supervisory information through program labels. Methods denoted with (†) use data augmentation and no pre-trained CNN.

### 3.3. What does conditional batch norm learn?

To understand what our model learns, we use t-SNE [23] to visualize the CBN parameter vectors  $(\gamma, \beta)$ , of 2,000 random validation points, modulating first and last CBN layers in our model, as shown in Figure 4. The  $(\gamma, \beta)$  parameters of the first and last CBN layers are grouped by the low-level and high-level reasoning functions necessary to answer CLEVR questions, respectively. For example, the CBN parameters for `equal_color` and `query_color` are close for the first layer but apart for the last layer, and the same is true for `equal_shape` and `query_shape`, `equal_size` and `query_size`, and `equal_material` and `query_material`. Conversely, `equal_shape`, `equal_size`, and `equal_material` CBN parameters are grouped in the last layer but split in the first layer. Similar patterns emerge when visualizing residual block activations. Thus, we see that CBN learns a sort of function-based modularity, directly from language and image inputs and without an architectural prior on modularity. Simply with end-to-end training, our model learns to handle not only different types of questions differently, but also different types of question sub-parts differently, working from low-level to high-level processes as is the proper approach to answer CLEVR questions.

Additionally, we observe that many points that break the previously mentioned clustering patterns do so in meaningful ways. For example, Figure 4 shows that some `count` questions have last layer CBN parameters far from those of other `count` questions but close to those of `exist` questions. Closer examination reveals that these `count` questions have answers of either 0 or 1, making them similar to `exist` questions.

### 3.4. Error analysis

An analysis of our model’s errors reveals that 94% of its counting mistakes are off-by-one errors, indicating our model has learned underlying concepts behind counting, such as close relationships between close numbers.

As shown in Figure 3, our CBN model struggles more on questions that require more steps, as indicated by the length of the corresponding CLEVR programs; error rates for questions requiring 10 or fewer steps are around 1.5%, while error rates for questions requiring 17 or more steps are around 5.5%, more than three times higher.

Furthermore, the model sometimes makes curious reasoning mistakes a human would not. In Figure 5, we show an exam-

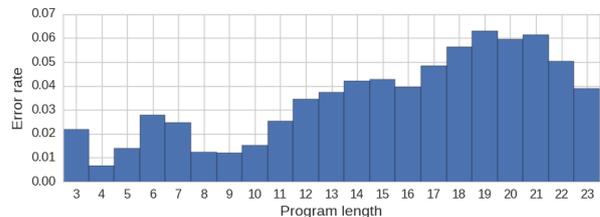


Figure 3: Validation error rate by program length.

ple where our model correctly counts two cyan objects and two yellow objects but simultaneously does not answer that there are the same number of cyan and yellow objects. In fact, it does not answer that the number of cyan blocks is more, less, or equal to the number of yellow blocks. These errors could be prevented by directly minimizing logical inconsistency, which is an interesting avenue for future work orthogonal to our approach.

These types of mistakes in a state-of-the-art visual reasoning model suggest that more work is needed to truly achieve human-like reasoning and logical consistency. We view CLEVR as a curriculum of tasks and believe that the key to the most meaningful and advanced reasoning lies in tackling these last few percentage points of error.

## 4. Related Work

One leading approach for visual reasoning is the Program Generator + Execution Engine model from [10]. This approach consists of a sequence-to-sequence Program Generator (PG), which takes in a question and outputs a sequence corresponding to a tree of composable Neural Modules, each of which is a two-layer residual block similar to ours. This tree of Neural Modules is assembled to form the Execution Engine (EE) that then predicts an answer from the image. The PG+EE model uses a strong prior by training with program labels and explicitly modeling the compositional nature of reasoning. Our approach learns to reason directly from textual input without using additional cues or a specialized architecture.

This modular approach is part of a recent line of work in Neural Module Networks [13, 24, 25]. Of these, End-to-End Module Networks (N2NMN) [13] also tackle visual reasoning but do not perform as well as other approaches. These methods also use strong priors by modeling the compositionality of reasoning, using program-level supervision, and building per-module, hand-crafted neural architectures for specific functions.

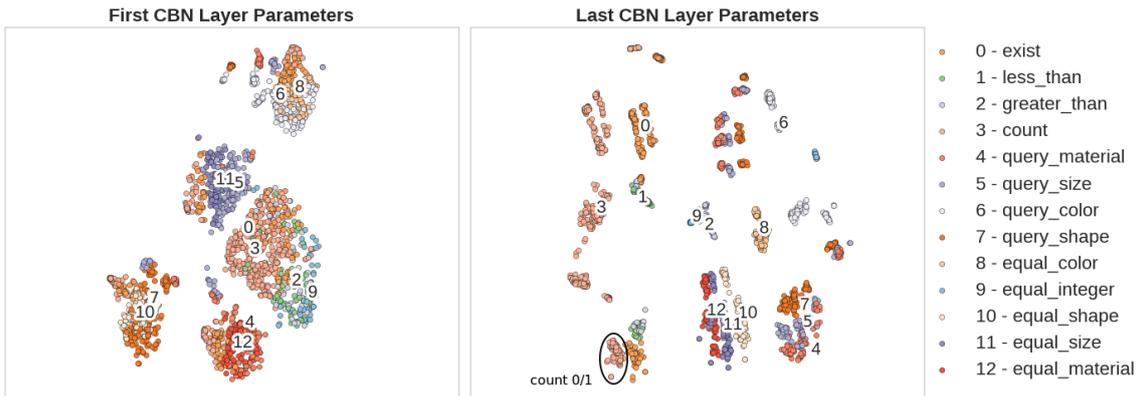
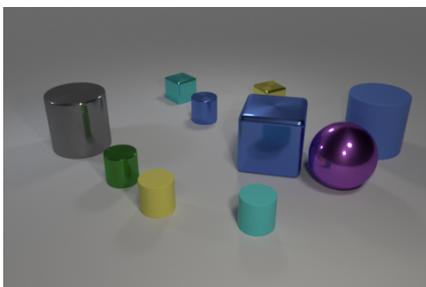


Figure 4:  $t$ -SNE plots of  $\gamma$ ,  $\beta$  of the first BN layer of the first residual block (left) and the last BN layer of the last residual block (right). CBN parameters are grouped by low-level reasoning functions for the first layer and by high-level reasoning functions for the last layer.



Question	Answer
How many yellow things are there?	2
How many cyan things are there?	2
Are there as many yellow things as cyan things?	No
Are there more yellow things than cyan things?	No
Are there fewer yellow things than cyan things?	No

Figure 5: An interesting failure example where our model counts correctly but compares counts erroneously. Its third answer is incorrect and inconsistent with its other answers.

Relation Networks (RNs) from [11] are another leading approach for visual reasoning. RNs use an MLP to carry out pairwise comparisons over each location of extracted convolutional features over an image, including LSTM-extracted question features as input to this MLP. RNs then element-wise sum over the resulting comparison vectors to form another vector from which a final classifier predicts the answer. This approach is end-to-end differentiable and trainable from scratch to high performance, as we show in Table 1. Our approach lifts the explicitly relational aspect of this model, freeing our approach from the use of a comparison-based prior, as well as the scaling difficulties of pairwise comparisons over spatial locations.

CBN itself has its own line of work. The results of [14, 16] show that the closely related Conditional Instance Normalization is able to successfully modulate a convolutional style-transfer network to quickly and scalably render an image in a huge variety of different styles, simply by learning to output a different set of BN parameters based on target style. For visual question answering, answering general questions often of natural images, de Vries *et al.* [15] show that CBN performs highly on real-world VQA and GuessWhat?! datasets, demonstrating CBN’s effectiveness beyond the simpler CLEVR images. Their

architecture conditions 50 BN layers of a pre-trained ResNet. We show that a few layers of CBN after a ResNet can also be highly effective, even for complex problems. We also show how CBN models can learn to carry out multi-step processes and reason in a structured way — from low-level to high-level.

Additionally, CBN is essentially a post-BN, feature-wise affine conditioning, with BN’s trainable scalars turned off. Thus, there are many interesting connections with other conditioning methods. A common approach, used for example in Conditional DCGANs [26], is to concatenate constant feature maps of conditioning information to the input of convolutional layers, which amounts to adding a post-convolutional, feature-wise conditional bias. Other approaches, such as LSTMs [27] and Hierarchical Mixtures of Experts [28], gate an input’s features as a function of that same input (rather than a separate, conditioning input), which amounts to a feature-wise, conditional scaling, restricted to between 0 and 1. CBN consists of both scaling and shifting, each unrestricted, giving it more capacity than many of these related approaches. We leave exploring these connections more in-depth for future work.

## 5. Conclusion

With a simple and general model based on CBN, we show it is possible to achieve state-of-the-art visual reasoning on CLEVR without explicitly incorporating reasoning priors. We show that our model learns an underlying structure required to answer CLEVR questions by finding clusters in the CBN parameters of our model; earlier parameters are grouped by low-level reasoning functions while later parameters are grouped by high-level reasoning functions. Simply by manipulating feature maps with CBN, a RNN can effectively use language to influence a CNN to carry out diverse and multi-step reasoning tasks over an image. It is unclear whether CBN is the most effective general way to use conditioning information for visual reasoning or other tasks, as well as what precisely about CBN is so effective. Other approaches [26, 27, 28, 29, 30, 31, 32] employ a similar, repetitive conditioning, so perhaps there is an underlying principle that explains the success of these approaches. Regardless, we believe that CBN is a general and powerful technique for multi-modal and conditional tasks, especially where more complex structure is involved.

## 6. Acknowledgements

We would like to thank the developers of PyTorch (<http://pytorch.org/>) for their elegant deep learning framework. Also, our implementation was based off the open-source code from [10]. We thank Mohammad Pezeshki, Dzmitry Bahdanau, Yoshua Bengio, Nando de Freitas, Joelle Pineau, Olivier Pietquin, Jérémie Mary, Chin-Wei Huang, Layla Asri, and Max Smith for helpful feedback and discussions, as well as Justin Johnson for CLEVR test set evaluations. We thank NVIDIA for donating a DGX-1 computer used in this work. We also acknowledge FRQNT through the CHIST-ERA IGLU project and CPER Nord-Pas de Calais and FEDER DATA Advanced data science and technologies 2015-2020 for funding our research.

## 7. References

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. of NIPS*, 2012.
- [2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. of CVPR*, 2016.
- [3] K. Cho, B. van Merriënboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in *Proc. of EMNLP*, vol. abs/1406.1078, 2014.
- [4] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Proc. of NIPS*, 2014.
- [5] M. Malinowski and M. Fritz, “A multi-world approach to question answering about real-world scenes based on uncertain input,” in *Proc. of NIPS*, 2014.
- [6] D. Geman, S. Geman, N. Hallonquist, and L. Younes, “Visual Turing test for computer vision systems,” vol. 112, no. 12. National Acad Sciences, 2015, pp. 3618–3623.
- [7] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, “VQA: Visual Question Answering,” in *Proc. of ICCV*, 2015.
- [8] H. de Vries, F. Strub, S. Chandar, O. Pietquin, H. Larochelle, and A. Courville, “GuessWhat?! Visual object discovery through multi-modal dialogue,” in *Proc. of CVPR*, 2017.
- [9] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick, “CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning,” in *Proc. of CVPR*, 2017.
- [10] J. Johnson, B. Hariharan, L. van der Maaten, J. Hoffman, F. Li, C. L. Zitnick, and R. B. Girshick, “Inferring and executing programs for visual reasoning,” 2017. [Online]. Available: <http://arxiv.org/abs/1705.03633>
- [11] A. Santoro, D. Raposo, D. G. T. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, , and T. Lillicrap, “A simple neural network module for relational reasoning,” *CoRR*, vol. abs/1706.01427, 2017. [Online]. Available: <http://arxiv.org/abs/1706.01427>
- [12] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, “Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering,” in *Proc. of CVPR*, 2017.
- [13] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko, “Learning to reason: End-to-end module networks for visual question answering,” *CoRR*, vol. abs/1704.05526, 2017. [Online]. Available: <http://arxiv.org/abs/1704.05526>
- [14] V. Dumoulin, J. Shlens, and M. Kudlur, “A learned representation for artistic style,” in *Proc. of ICLR*, 2017.
- [15] H. de Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. C. Courville, “Modulating early visual processing by language,” *arXiv preprint arXiv:1707.00683*, 2017. [Online]. Available: <http://arxiv.org/abs/1707.00683>
- [16] G. Ghiasi, H. Lee, M. Kudlur, V. Dumoulin, and J. Shlens, “Exploring the structure of a real-time, arbitrary neural artistic stylization network,” *CoRR*, vol. abs/1705.06830, 2017. [Online]. Available: <http://arxiv.org/abs/1705.06830>
- [17] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. of ICML*, 2015.
- [18] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” in *Deep Learning workshop at NIPS*, 2014.
- [19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [20] N. Watters, A. Tachetti, T. Weber, R. Pascanu, P. Battaglia, , and D. Zoran, “Visual interaction networks,” *CoRR*, vol. abs/1706.01433, 2017. [Online]. Available: <http://arxiv.org/abs/1706.01433>
- [21] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. of ICLR*, 2015.
- [22] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola, “Stacked attention networks for image question answering,” in *Proc. of CVPR*, 2016.
- [23] L. van der Maaten and G. Hinton, “Visualizing data using t-sne,” *JMLR*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- [24] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, “Neural module networks,” in *Proc. of CVPR*, 2016.
- [25] J. Andreas, R. Marcus, T. Darrell, and D. Klein, “Learning to compose neural networks for question answering,” in *Proc. of NAACL*, 2016.
- [26] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” 2015. [Online]. Available: <http://arxiv.org/abs/1511.06434>
- [27] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: <http://dx.doi.org/10.1162/neco.1997.9.8.1735>
- [28] M. I. Jordan and R. A. Jacobs, “Hierarchical mixtures of experts and the em algorithm,” *Neural Comput.*, vol. 6, no. 2, pp. 181–214, Mar. 1994. [Online]. Available: <http://dx.doi.org/10.1162/neco.1994.6.2.181>
- [29] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” 2016. [Online]. Available: <http://arxiv.org/abs/1609.03499>
- [30] A. van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves *et al.*, “Conditional image generation with pixelcnn decoders,” in *Proc. of NIPS*, 2016.
- [31] S. E. Reed, A. van den Oord, N. Kalchbrenner, S. G. Colmenarejo, Z. Wang, D. Belov, and N. de Freitas, “Parallel multiscale autoregressive density estimation,” 2017. [Online]. Available: <http://arxiv.org/abs/1703.03664>
- [32] S. Reed, A. van den Oord, N. Kalchbrenner, V. Bapst, M. Botvinick, and N. de Freitas, “Generating interpretable images with controllable structure,” in *Proc. of ICLR*, 2017.