Learning Visual Reasoning Without Strong Priors

Ethan Perez1,2, Harm de Vries3, Florian Strub3, Vincent Dumoulin4, Aaron Courville1

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

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

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 general intelligence.

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 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.

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.

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: Deep Learning, Language and Vision

Note: A full paper extending this study is available at [http://arxiv.org/abs/1709.07871](http://arxiv.org/abs/1709.07871) with additional references, experiments, and analysis.

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], speech recognition [17], 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 [15]. To explain BN, we define \( \mathcal{B} = \{ P_i \}_{i=1}^{N} \) as a mini-batch of \( N \) samples, where \( P \) corresponds to input feature maps whose subscripts \( c, h, w \) refer to the \( c \)-th feature map at the spatial location \( (h, w) \). We also define \( \gamma_c \) and \( \beta_c \) as per-channel, trainable

\[
F_i = \frac{F_i - \mu_i}{\sigma_i} \gamma_i + \beta_i,
\]

where \( \mu_i \) and \( \sigma_i \) are the mean and standard deviation across the mini-batch, respectively.

This process is repeated for each mini-batch, and the result is a normalized distribution of activations. CBN extends this idea by conditioning the normalization on the input features, allowing the network to learn different normalization parameters for each input feature map.

The full CBN process is as follows:

\[
F_i = \frac{F_i - \mu_{i,c}}{\sigma_{i,c}} \gamma_{c} + \beta_{c},
\]

where \( \mu_{i,c} \) and \( \sigma_{i,c} \) are the per-channel mean and standard deviation, respectively.

\[
\gamma_{c} = \frac{1}{N} \sum_{i=1}^{N} \frac{F_i - \mu_{i,c}}{\sigma_{i,c}}
\]

\[
\beta_{c} = \frac{1}{N} \sum_{i=1}^{N} F_i - \gamma_{c} \mu_{i,c}
\]

This allows the network to learn different normalization parameters for each input feature map, which can be particularly useful for multi-modal tasks like visual reasoning.

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

(a) What number of cylinders are small purple things or yellow rubber things?

Predicted: 2

(b) What color is the other object that is the same shape as the large brown matte thing?

Predicted: Brown
The visual pipeline extracts $14 \times 14$ image features using the $conv_4$ layer of a ResNet-101 [2] pre-trained on ImageNet [20], 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] [21], we concatenate coordinate feature maps indicating relative spatial position (scaled from $-1$ to $1$) to the image features, each residual block’s input, and the classifier’s input. We train our model end-to-end from scratch with Adam (learning rate $3e^{-4}$) [22], early stopping on the validation set, weight decay ($1e^{-4}$), 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 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 1. 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. 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, CLEVR outperforms Stacked Attention networks (CNN+LSTM+SA in [11]) by 21.0%. Stacked Attention networks are highly effective for visual question answering with simpler questions [23] 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.

![Figure 2: The linguistic pipeline (left), visual pipeline (middle), and CBN residual block architecture (right) of our model.](image)
equal, query

The (γ, β) parameters of the first and last CBN layers are grouped in the last layer but split in the first layer. The (γ, β) 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 module, hand-crafted neural architectures for specific functions.

Additionally, we observe that many points that break the previously mentioned clustering patterns do so in meaningful ways. For example, Figure 3 shows that some count questions have last layer CBN parameters far from those of other count questions but close to those of exist questions. Close 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 3, we show an example 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, 25, 26]. 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.
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 [27], 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 [28] and Hierarchical Mixtures of Experts [29], 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 [27,28,29,30,31,32,33] 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 Merrienboer, 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, “Virtual testing for computer vision systems,” vol. 112, no. 12. National Acad Sciences, 2012.

[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. Courville, O. Pietquin, H. Larochelle, and A. Courville, “Question!” 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/1706.01433

[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] T. Kim, I. Song, and Y. Bengio, “Dynamic layer normalization for adaptive neural acoustic modeling in speech recognition,” in Proc. of Interspeech. 2017.

[18] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in Proc. of ICMIL. 2015.

[19] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” in Deep Learning workshop at NIPS. 2014.

[20] 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 L. Fei-Fei, “Imagenet large scale visual recognition challenge,” International Journal of Computer Vision, vol. 115, no. 3, pp. 211–252, 2015.