On transfer learning using a MAC model variant

Vincent Marois **T.S. Jayram** **Vincent Albouy**
Tomasz Kornuta **Younes Bouhadjar** **Ahmet S. Ozcan**
IBM Research AI, Almaden Research Center, San Jose, USA
{vmarois,jayram,tkornut,byounes,asozcan}@us.ibm.com
{vincent.albouy}@ibm.com

Abstract

We introduce a variant of the MAC model (Hudson and Manning, ICLR 2018) with a simplified set of equations that achieves comparable accuracy, while training faster. We evaluate both models on CLEVR and CoGenT, and show that, transfer learning with fine-tuning results in a 15 point increase in accuracy, matching the state of the art. Finally, in contrast, we demonstrate that improper fine-tuning can actually reduce a model's accuracy as well.

1 Introduction

Reasoning over visual inputs is a fundamental characteristic of human intelligence. Reproducing this ability with artificial systems is a challenge that requires learning relations and compositionality [HAR+17, JHvdM+17b]. The Visual Question Answering (VQA) [AAL+15, MF14b, WTW+17] task has been tailored to benchmark this type of reasoning, combining natural language processing and visual recognition.

Many approaches have been explored, such as modular networks, that combine modules coming from a predefined collection [ARDK16, JHvdM+17b, MTSM18]. Attention mechanisms ([BCB14], [XBK+15]) are also used to guide the focus of the system over the image and the question words.

Several VQA datasets have been proposed (e.g. DAQUAR [MF14a], VQA [AAL+15]); nonetheless, these datasets contain several biases (e.g. unbalanced questions or answers) that are often exploited by systems during learning [GKSS+17]. The CLEVR dataset [JHvdM+17a] was designed to address these issues. The synthetic nature of its images & questions enables detailed analysis of visual reasoning, and allows for variations, to test a particular ability such as generalization or transfer learning. One variation of CLEVR, called CoGenT (Compositional Generalization Test), measures whether models learn separate representations for color and shape instead of memorizing all possible combinations. For details, see Appendix A (all appendices are in the supplementary material).

One of the most recent exciting models aiming at solving VQA is called Memory, Attention, and Composition (MAC) [HM18], which performs a sequence of attention-based reasoning operations. Although the performance of MAC has been proven, several questions arise: Does the model really learn relations between objects? How does the model represent these relations in its reasoning steps? Is the model representing concepts like objects attributes (shape, color, size)?

In this work, we further investigate the interpretability and generalization capabilities of the MAC model. We propose a new set of equations that simplifies the core of the model (S-MAC). It trains faster and achieves comparable accuracy on CLEVR. Second, we show that both models achieve comparable performance with zero-shot learning, when trained on CoGenT-A and tested on CoGenT-B. With fine-tuning however, we obtained a significant 15 points increase in the accuracy, matching state-of-the-art results [PSDV+17, MTSM18]. Last, we illustrate using CoGenT-B that, without adequate care, fine-tuning can actually reduce a model's accuracy.

Visually Grounded Interaction and Language (ViGIL) Workshop, NIPS 2018, Montréal, Canada.

Table 1: Comparing the number of position-independent parameters between MAC & S-MAC cells.

Model	Read Unit	Write Unit	Control Unit
MAC	787,969	524,800	525,313
simplified MAC	263,168	262,656	263,168
Reduction by [%]	67%	50%	50%

2 MAC network and our proposed simplification

The MAC network [HM18] is a recurrent model that performs sequential reasoning, where each step involves analyzing a part of the question followed by shifting the attention over the image. The core of the model is the MAC cell, supported with an input unit that processes the question and image pair, and output unit which produces the answer. The input unit uses an LSTM [HS97] to process the question in a word-by-word manner producing a sequence of *contextual words* and a final *question representation*. Additionally, the input unit uses a pre-trained ResNet [HZRS16] followed by two CNN layers to extract a feature map (referred to as *knowledge base*) from the image.

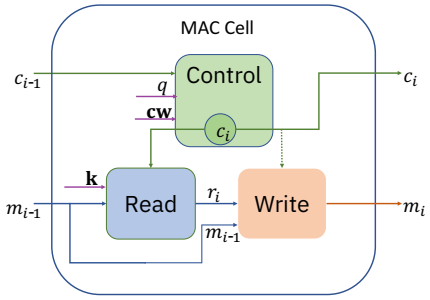

Figure 1: The MAC cell, reproduced on the basis of [HM18].

The MAC cell consists of a control unit, a read unit and a write unit (Fig. 1). The control unit is updating the control state c_i and drives the attention over the list of *contextual words* $\mathbf{cw}$ taking into account the *question representation* q. Guided by c_i, the read unit extracts information from the *knowledge base* $\mathbf{k}$ and combines it with the previous memory state m_{i-1} to produce the *read vector* r_i. Finally, the write unit integrates r_i and m_{i-1} to update the memory state. Detailed equations are described in the next section.

2.1 Simplified MAC network

Our proposed modification to the MAC network is based on two heuristic simplifications of the MAC cell. First, we observe that, taking the MAC cell equations as a whole, consecutive linear layers (with no activation in-between) can be combined as one linear layer. Next, we assume that dimension-preserving linear layers are invertible so as to avoid information loss. Applying this principle to the equations, with a careful reorganization, we can apply a single linear layer to the knowledge base *prior* to all the reasoning steps and work with this *projected* knowledge base as-is throughout the reasoning steps.

Notation. For each knowledge base $\mathbf{k}$ of dimension $H \times W \times d$, let $\mathbf{k}_{hw}$ be the d-dimensional vector indexed by $h \in \{1, 2, \ldots, H\}$ and $w \in \{1, 2, \ldots, W\}$. Let i denote the index (position) i of the reasoning step.

In the description below, the original MAC cell equations are shown on the *left* while our simplified equations are shown (in color) on the *right*. The equation numbering is the same as in [HM18].

Control unit: For both models, in the control unit, the question q is first transformed in each step of the reasoning using a *position-aware* linear layer depending on i: $q_i = U_i^{[d \times 2d]} q + b_i^{[d]}$.

$$cq_i = W_{cq}^{[d \times 2d]}[c_{i-1}, q_i] + b_{cq}^{[d]} \quad \text{(c1)}$$
$$ca_{is} = W_{ca}^{[1\times d]}(cq_i \odot \mathbf{cw}_s) + b_{ca}^{[1]} \quad \text{(c2.1)}$$
$$cv_{is} = \text{softmax}(ca_{is}) \quad \text{(c2.2)}$$
$$\mathbf{c}_i = \sum_s cv_{is}\, \mathbf{cw}_s \quad \text{(c2.3)}$$

$$cq_i = W_{cq}^{[d \times d]} c_{i-1} + q_i \quad \text{(c1)}$$
$$ca_{is} = W_{ca}^{[1 \times d]}(cq_i \odot \mathbf{cw}_s) \quad \text{(c2.1)}$$
$$cv_{is} = \text{softmax}(ca_{is}) \quad \text{(c2.2)}$$
$$\mathbf{c}_i = \sum_s cv_{is}\, \mathbf{cw}_s \quad \text{(c2.3)}$$

Read and write units:

$$I_{ihw} = (W_m^{[d\times d]}\mathbf{m}_{i-1} + b_m^{[d]}) \odot (W_k^{[d\times d]}\mathbf{k}_{hw} + b_k^{[d]}) \quad \text{(r1)}$$

$$I'_{ihw} = W_{I'}^{[d\times 2d]}[I_{ihw}, \mathbf{k}_{hw}] + b_{I'}^{[d]} \quad \text{(r2)}$$

$$ra_{ihw} = W_{ra}^{[1\times d]}(\mathbf{c}_i \odot I'_{ihw}) + b_{ra}^{[1]} \quad \text{(r3.1)}$$

$$rv_{ihw} = \text{softmax}(ra_{ihw}) \quad \text{(r3.2)}$$

$$\mathbf{r}_i = \sum_s rv_{ihw}\,\mathbf{k}_{hw} \quad \text{(r3.3)}$$

$$\mathbf{m}_i = W_{rm}^{[d\times d]}[\mathbf{r}_i, \mathbf{m}_{i-1}] + b_{rm}^{[d]} \quad \text{(w1)}$$

$$I_{ihw} = \mathbf{m}_{i-1} \odot \mathbf{k}_{hw} \quad \text{(r1)}$$

$$I'_{ihw} = W_{I'}^{[d\times d]} I_{ihw} + b_{I'}^{[d]} + \mathbf{k}_{hw} \quad \text{(r2)}$$

$$ra_{ihw} = W_{ra}^{[1\times d]}(\mathbf{c}_i \odot I'_{ihw}) \quad \text{(r3.1)}$$

$$rv_{ihw} = \text{softmax}(ra_{ihw}) \quad \text{(r3.2)}$$

$$\mathbf{r}_i = \sum_s rv_{ihw}\,\mathbf{k}_{hw} \quad \text{(r3.3)}$$

$$\mathbf{m}_i = W_{rm}^{[d\times 2d]}\mathbf{r}_i + b_{rm}^{[d]} \quad \text{(w1)}$$

As seen above, being noticeably simpler, the S-MAC obtains significant reduction in the number of position-independent parameters across all units (see Table 1). Our experiments demonstrate that this gives us noticeable savings in the training time. However, since the computation time is also dominated by the position-aware layers in the control unit, as well as the input unit, the speedup is not as large as we desire.

3 Experiments

Our experiments were intended to study MAC's and S-MAC's generalization as well as transfer learning abilities in different settings. We used CLEVR and CoGenT to address these different aspects. The first experiment studied the training time and the capability of the models to generalize on the same type of dataset that it was trained on. This was used mainly as a baseline for the further experiments that were intended to study how well the transfer learning performed in comparison to the baseline results. The second experiment studied the capability of the models to succeed in doing transfer learning from domain A to domain B when trained on different combinations of the respective domains. The third experiment was intended to see whether the performance improves if the model could be further trained on a small subset of the dataset from domain B. Table 2 presents the most important results, focusing on S-MAC in particular; see Appendix B for the entire table.

For all experiments, the initial training procedure is as follows: we train the given model for 20 epochs on 90% of the training sets of CLEVR & CoGenT separately. We keep the remaining 10% for validating the model at every epoch, and use the original validation sets as test sets.

Our implementation of both MAC models used PyTorch (v0.4.0) [PGC+17]. We relied on the MI-Prometheus [KMM+18] framework that enables fast experimentation of the cross product of models and datasets[1]. We used NVIDIA's GeForce GTX TITAN X GPUs. We followed the implementation details indicated in the supplemental material (Sec. A) of the original paper [HM18], to ensure a faithful comparison.

For the CLEVR dataset, the training wall time of MAC (row a) is consistent with what is reported in the original paper (roughly 30h of training for 20 epochs). S-MAC trains faster, showing a decrease of 10.5% in wall time (row b), due to the reductions in the number of parameters as shown earlier. This was consistently observed across the other experiments as well.

Turning to the generalization performance (row a), MAC on CLEVR yields an accuracy of 96.17%, which is taken as a reference experiment. S-MAC reaches an accuracy of 95.29% on CLEVR (row b), indicating that the simplifications did not hinder its generalization capability. Similar performance was observed for generalization on CoGenT-A (row c).

Before fine-tuning, we wanted to estimate the best upper bounds on accuracy that we could possibly get by doing transfer learning. As both CoGenT datasets contain complementary subsets of colors/shapes combinations present in CLEVR, we evaluated CLEVR-trained models on the CoGenT datasets. Even though the CoGenT datasets were generated using more restricted parameters, the models obtained nearly equal accuracy (rows d-e).

[1]To reproduce the presented research please follow: `https://github.com/IBM/mi-prometheus/`

Table 2: CLEVR & CoGenT accuracies for the MAC & S-MAC models. The [Training] column indicates wall times and final accuracies on the training set. For fine-tuning, we use 30k samples of the test set, and kept the remainder for testing. The [Fine-tuning] column reports the used sub-set (30k samples) and the final accuracies on this sub-set during training. The [Test] column reports the used set and the obtained test accuracies. If no fine-tuning was done, the whole indicated set was used for testing.

Model	Training			Fine-tuning		Test		Row
	Dataset	Time [h:m]	Acc [%]	Dataset	Acc [%]	Dataset	Acc [%]	
MAC	CLEVR	30:52	96.70	–	–	CLEVR	96.17	(a)
S-MAC	CLEVR	28:30	95.82	–	–	CLEVR	95.29	(b)
	CoGenT-A	28:33	96.09	–	–	CoGenT-A	95.91	(c)
	CLEVR	28:30	95.82	–	–	CoGenT-A	95.47	(d)
						CoGenT-B	95.58	(e)
	CoGenT-A	28:33	96.09	–	–	CogenT-B	78.71	(f)
				CoGenT-B	96.85	CoGenT-A	91.24	(g)
						CoGenT-B	94.55	(h)
	CLEVR	28:30	95.82	CoGenT-B	97.67	CoGenT-A	92.11	(i)
						CoGenT-B	92.95	(j)

Evaluating S-MAC on CoGenT shows that, similar to [JHvdM$^+$17b, MTSM18], the score is worse on CoGenT-B (zero-shot learning, row f) than CoGenT-A after training on CoGenT-A data only (generalization, row c).

Following [JHvdM$^+$17b, PSDV$^+$17]), we then fine-tune S-MAC using 3k images and 30k questions from the CoGenT-B data (for 10 epochs), and re-evaluate it on both conditions. This enables much higher accuracy on CoGenT-B, of at least a 15 points increase (row h). Performance on CoGenT-A is slightly worse, dropping by 4 points (row g). This seems to indicate that S-MAC is able to learn new combinations of shape & color without forgetting the ones it learned during the initial training.

To study the pitfalls of fine-tuning, we conducted a final set of experiments, where we fine-tuned a CLEVR-trained S-MAC model on CoGenT-B for 10 epochs, as before. Surprisingly, this operation handicapped the generalization of the model not only on CoGenT-A (row i), but also on CoGenT-B (row j, a 3 point drop compared to row e). This highlights the delicate nature of fine-tuning, with respect to the correlation between the datasets. This warrants further investigations.

We have analyzed cases of failure on CoGenT-B to illustrate this point. Please refer to Appendix D for more details. Aside, Appendix C presents a comparison of results obtained by the S-MAC model and selected state-of-the-art models [JHvdM$^+$17b, PSDV$^+$17, MTSM18] on CoGenT.

4 Conclusion

We have introduced S-MAC, a simplified variant of the MAC model. Because it has nearly half the number of parameters in the recurrent portion MAC cell, it trains faster while maintaining an equivalent test accuracy.

Our experiments on zero-short learning show that the MAC model has poor performance in line with the other models in the literature. Thus, this remains an interesting problem to investigate how we can train it to disentangle the concepts of shape and color.

With fine-tuning, the MAC model indeed achieves much improved performance, matching state of the art. However, we have showed that correlation between the different domains must be taken into account when fine-tuning, otherwise potentially leading to decreased performance.

References

[AAL+15] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[ARDK16] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. *arXiv preprint arXiv:1601.01705*, 2016.

[BCB14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[GKSS+17] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, volume 1, page 3, 2017.

[HAR+17] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. *CoRR, abs/1704.05526*, 3, 2017.

[HM18] Drew A. Hudson and Christopher D. Manning. Compositional attention networks for machine reasoning. *International Conference on Learning Representations*, 2018.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016.

[JHvdM+17a] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1988–1997. IEEE, 2017.

[JHvdM+17b] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross B Girshick. Inferring and executing programs for visual reasoning. In *ICCV*, pages 3008–3017, 2017.

[KMM+18] Tomasz Kornuta, Vincent Marois, Ryan L. McAvoy, Younes Bouhadjar, Alexis Asseman, Vincent Albouy, T.S. Jayram, and Ahmet S. Ozcan. Accelerating machine learning research with mi-prometheus. *NIPS 2018 Workshop MLOSS*, 2018.

[MF14a] Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in neural information processing systems*, pages 1682–1690, 2014.

[MF14b] Mateusz Malinowski and Mario Fritz. Towards a visual turing challenge. *arXiv preprint arXiv:1410.8027*, 2014.

[MTSM18] David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4942–4950, 2018.

[PGC+17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. 2017.

[PSDV+17] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. *arXiv preprint arXiv:1709.07871*, 2017.

[WTW+17] Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163:21–40, 2017.

[XBK+15] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.