Learning Multi-Step Spatio-Temporal Reasoning with Selective Attention Memory Network

T.S. Jayram, Tomasz Kornuta, Vincent Albouy, Emre Sevgen, Ahmet Ozcan
IBM Research AI
Almaden Research Center, San Jose, CA 95120, USA

Abstract

Visual reasoning in videos requires understanding temporal concepts in addition to the objects and their relations in a given frame. To that end, we present Selective Attention Memory Network (SAMNet), an end-to-end differentiable recurrent model equipped with external memory. In analogy with human reasoning, SAMNet can perform multi-step reasoning on a frame-by-frame basis, and dynamically control information flow to the memory to store context-relevant representations and from the memory to answer questions. We tested our model on the COG dataset (a multi-frame visual question answering test). SAMNet outperforms the the original COG model, especially on the hardest version of the dataset with longer sequences and a maximum number of distractors. We also demonstrate that our model has extraordinary generalization capabilities going from easy to hard tasks, without and with additional fine-tuning.

1 Introduction

Integration of vision and language in deep neural network models allows the system to learn joint representations of objects, concepts, and relations. Potentially, this approach can lead us towards Harnad's *symbol grounding problem* [8] but we are quite far from achieving the full capabilities of visually grounded language learning.

Recently, there is a growing interest in neuro-symbolic approaches, which can combine the power of representation learning and symbolic logic that is interpretable [14]. These approaches focus on *symbol manipulation* rather than learning *grounded symbols*. Furthermore, symbolic priors (e.g., domain knowledge) and integration of logic depend on hand-crafted modules. In the near term, this direction is certainly promising and can address some of the shortcomings of machine learning (i.e., the lack of explainability)[23]. However, in the long run, the desire is to learn grounded representations, which may lead to the emergence of symbols [20].

Starting with Image Question Answering [13, 1] and Image Captioning [11], a variety of tasks that integrate vision and language have appeared in the past several years [16]. Those directions include e.g., Video QA [21] and Video Action Recognition [17], that provide an additional challenge of understanding *temporal* aspects, and Video Reasoning [19, 25], that tackles both spatial (comparison of object attributes, counting and other relational question) and temporal aspects and relations (e.g. object disappearance). To deal with the temporal aspect most studies typically cut the whole video into clips; e.g., in [19] the model extracts visual features from each frame and aggregates features first into clips, followed by aggregation over clips to form a single video representation. Still, when reasoning and producing the answer, the system in fact has *access to all frames*. Similarly, in Visual Dialog [3] the system memorizes the whole dialog history. However, in real-time dialog or video monitoring, it is not always possible to keep the entire history of conversation nor all frames from the beginning of the recording.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

As evident from human cognition, attention and memory are the key competencies required to solve these problems, and unsurprisingly, the AI research is rapidly growing in these areas. The ability to deal with temporal casuality can pose a challenge for also in pure natural language processing (NLP) settings, e.g. in question answering (QA) and dialog applications. Current NLP solutions, in many problem settings, work around this challenge by processing the entire text input and reason over it multiple times using attention [22] or other mechanisms. For example, typical solutions to the bAbI reasoning task, such as Memory Networks [24], involve processing all the supporting facts at once and keeping them in memory all the time while searching for the answer.

Contributions. In this paper, we introduce a new model for visual reasoning that can dynamically process video input frame-by-frame, reason over each frame and store the salient concepts in memory so as to order to answer questions. Our experiments based on the COG dataset [25] indicate that the model can: (1) form temporal associations, i.e., grounding the time-related words with meaning; (2) learn complex, multi-step reasoning that involves grounding of words and visual representations of objects/attributes; (3) selectively control the flow of information to and from the memory to answer questions; and (4) update the memory only with relevant visual information depending on the temporal context.

2 Selective Attention Memory (SAM) Network

SAM Network (SAMNet for short) is an end-to-end differentiable recurrent model equipped with an external memory (Figure 1). At the conceptual level SAMNet draws from two core ideas: compositional reasoning as proposed e.g. in MAC (Memory-Attention-Composition) Network [9, 15] and utilization of external memory in Memory-Augmented Neural Networks such as NTM (Neural Turing Machine) [6], DNC (Differentiable Neural Computer) [7] or DWM (Differentiable Working Memory) [10].

A distinctive feature of the SAM Network is that it makes a single pass over the frames in temporal order, accessing one frame at a time. The memory locations store relevant objects representing contextual information about words in text and visual objects extracted from video. Each location of the memory stores a d-dimensional vector. The memory can be accessed through either content-based addressing, via dot-product attention, or location-based addressing. Using gating mechanisms, correct objects can be retrieved in order to perform multi-step spatio-temporal reasoning over text and video. A notable feature of this design is that the number of addresses N can be set to different values during training and testing to fit the characteristics of data.

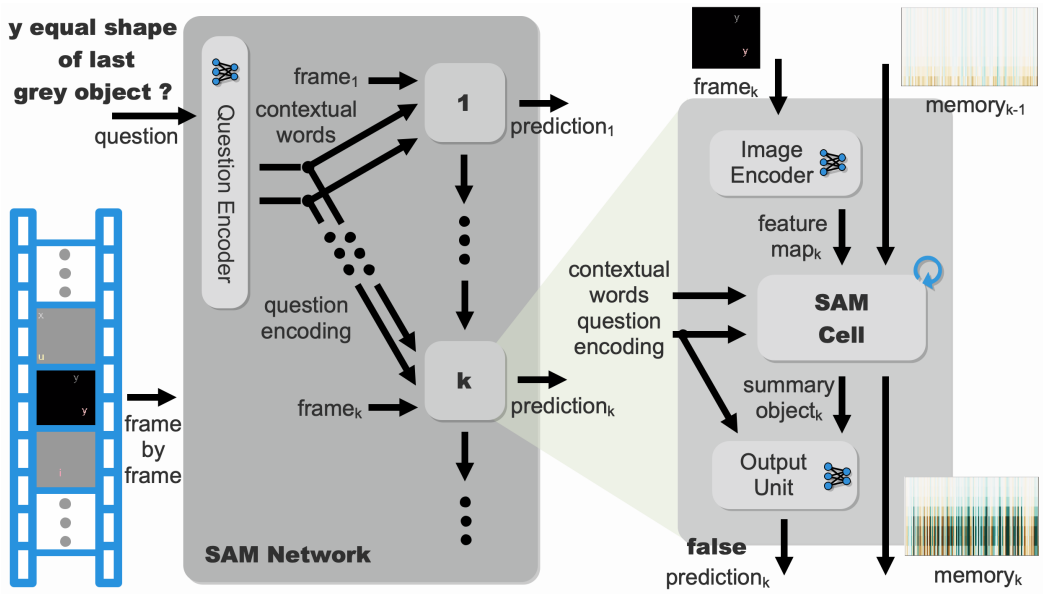


Figure 1: General architecture of SAMNet

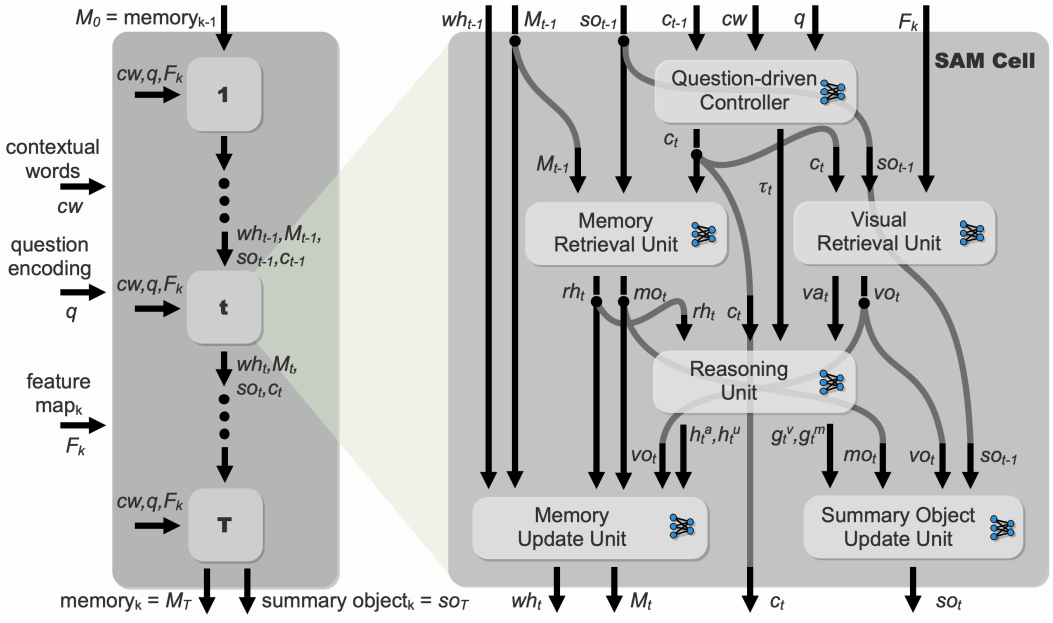


Figure 2: Unfolded reasoning steps with operations performed by the SAMCell

The core of SAMNet is a recurrent cell called a SAM Cell (Figure 2). Unrolling a new series of T cells for every frame enables T steps of compositional reasoning, similar to [9]. Information flows between frames through the external memory. During the t-th reasoning step, for $t = 1, 2, \ldots, T$, SAM Cell maintains the following information as part of its recurrent state: (a) $\mathbf{c}_t \in \mathbb{R}^d$, the control state used to drive the reasoning over objects in the frame and memory; and (b) $\mathbf{so}_t \in \mathbb{R}^d$, the summary visual object representing the relevant object for step t. Let $\mathbf{M}_t \in \mathbb{R}^{N \times d}$ denote the external memory with N slots at the end of step t. Let $\mathbf{wh}_t \in \mathbb{R}^N$ denote an attention vector over the memory locations; in a trained model, $\mathbf{wh}_t$ points to the location of first empty slot in memory for adding new objects.

Question-driven Controller. This module drives attention over the question to produce k control states, one per reasoning operation. The control state $\mathbf{c}_t$ at step t is then fed to a *temporal classifier*, a two-layer feedforward network with ELU activation used in the hidden layer of d units. The output τ_t of the classifier is intended to represent the different temporal contexts (or lack thereof) associated with the word in focus for that step of reasoning. For the COG dataset we pick 4 classes to capture the terms labeled "last", "latest", "now", and "none".

The visual retrieval unit uses the information generated above to extract a relevant object $\mathbf{vo}_t$ from the frame. A similar operation on memory yields the object $\mathbf{mo}_t$. The memory operation is based on an attention mechanism, and resembles content-based addressing on memory. Therefore, we obtain an attention vector over memory addresses that we interpret to be the *read head*, denoted by $\mathbf{rh}_t$. Note that the returned objects may be invalid, e.g., if the current reasoning step focuses on the phrase "last red square", $\mathbf{vo}_t$ is invalid even if the current frame contains a red square.

Reasoning Unit. This module is the backbone of SAMNet that determines what gating operations need to be performed on the external memory, as well as determining the location of the correct object for reasoning. To determine whether we have a valid object from the frame (and similarly for memory), we execute the following reasoning procedure. First, we take the visual attention vector $\mathbf{va}_t$ of dimension L, where L denotes the number of feature vectors for the frame, and compute a simple aggregate[1]: $vs_t = \sum_{i=1}^{L} [\mathbf{va}_t(i)]^2$. It can be shown that the more localized the attention vector is, the higher is the aggregate value. We perform a similar computation on the read head $\mathbf{rh}_t$ over memory locations. We feed these two values along with the temporal class weights τ_t to a 3-layer feedforward classifier with hidden ELU units to extract 4 gating values in $[0, 1]$ modulated for the current reasoning step: (a) g_t^{v}, which determines whether there is a valid visual object; (b)

[1]This is closely related to Tsallis entropy of order 2 and to Rényi entropy.

g_t^{m}, which determines whether there is a valid memory object. (c) h_t^{r}, which determines whether the memory should be updated by replacing a previously stored object with a new one; and (d) h_t^{a}, which determines whether a new object should be added to memory. We stress that the network has to learn via training how to correctly implement these behaviors.

Memory Update Unit. Unit first determines the memory location where an object could be added:

$$\mathbf{w}_t = h^{\mathrm{r}} \cdot \mathbf{rh}_t + h^{\mathrm{a}} \cdot \mathbf{wh}_{t-1}$$

Above, $\mathbf{w}_t$ denotes the pseudo-attention vector that represents the "location" where the memory update should happen. The sum of components of $\mathbf{w}_t$ is at most equal to 1; and $\mathbf{w}_t$ can even equal 0, e.g., whenever neither condition of adding a new object nor replacing an existing object holds true. We then update the memory accordingly as:

$$\mathbf{M}_t = \mathbf{M}_{t-1} \odot (\mathbf{J} - \mathbf{w}_t \otimes \mathbf{1}) + \mathbf{w}_t \otimes \mathbf{vo}_t,$$

where $\mathbf{vo}_t$ denotes the object returned by the visual retrieval unit. Here $\mathbf{J}$ denotes the all ones matrix, $\odot$ denotes the Hadamard product and $\otimes$ denotes the Kronecker product. Note that the memory is unchanged in the case where $\mathbf{w}_t = 0$, i.e., $\mathbf{M}_t = \mathbf{M}_{t-1}$. We finally update the write head so that it points to the succeeding address if an object was added to memory or otherwise stay the same. Let $\mathbf{wh}'_{t-1}$ denote the circular shift to the right of $\mathbf{wh}_{t-1}$ which corresponds to the soft version of the head update. Then:

$$\mathbf{wh}_t = h^{\mathrm{a}} \cdot \mathbf{wh}'_{t-1} + (1 - h^{\mathrm{a}}) \cdot \mathbf{wh}_{t-1}$$

Summary Update Unit. This unit updates the (recurrent) summary object to equal the outcome of the t-th reasoning step. We first determine whether the relevant object $\mathbf{ro}_t$ should be obtained from memory or the frame according to:

$$\mathbf{ro}_t = g_t^{\mathrm{v}} \cdot \mathbf{vo}_t + g_t^{\mathrm{m}} \cdot \mathbf{mo}_t$$

Note that $\mathbf{ro}_t$ is allowed to be a null object (i.e. 0 vector) in case neither of the gates evaluate to true. Finally, $\mathbf{so}_t$ is the output of a simple linear layer whose inputs are $\mathbf{ro}_t$ and the previous summary object $\mathbf{so}_{t-1}$. This serves to retain additional information that was in $\mathbf{so}_{t-1}$, e.g., if it held the partial result of a complex query with Boolean connectives.

3 Experiments

We have implemented and trained our SAMNet model using MI-Prometheus [12], a framework based on Pytorch [18]. We evaluated the model on the COG dataset [25], a video reasoning [16] dataset developed for the purpose of research on relational and temporal reasoning. Our experiments were designed to study SAMNet's performance as well as its generalization abilities in different settings. For this purpose, we used two different variants of the COG dataset (Table 1): an easy one (Canonical) and a Hard version to explore a wide range of difficulties. The main differences are the number of frames in the input sequence (4 vs. 8) and the maximum number of distractors (i.e., objects not relevant for the answer) per a single frame (1 vs. 10).

Table 1: Details of the Canonical and Hard variants of the COG dataset

Variant	number of frames	maximum memory duration	maximum number of distractors	size of training set	size of validation set	size of test set
Canonical	4	3	1	10000320	500016	500016
Hard	8	7	10	10000320	500016	500016

3.1 Performance comparison with the COG baseline

In our experiments we have trained SAMNet using 8 reasoning steps and external memory having 8 address locations, each being a vector of 128 floats. We have also carried out experiments with different numbers of reasoning steps and memory size, but this goes beyond the scope of this paper. In here, we have focused on 22 classification tasks and compared our results with the baseline model.

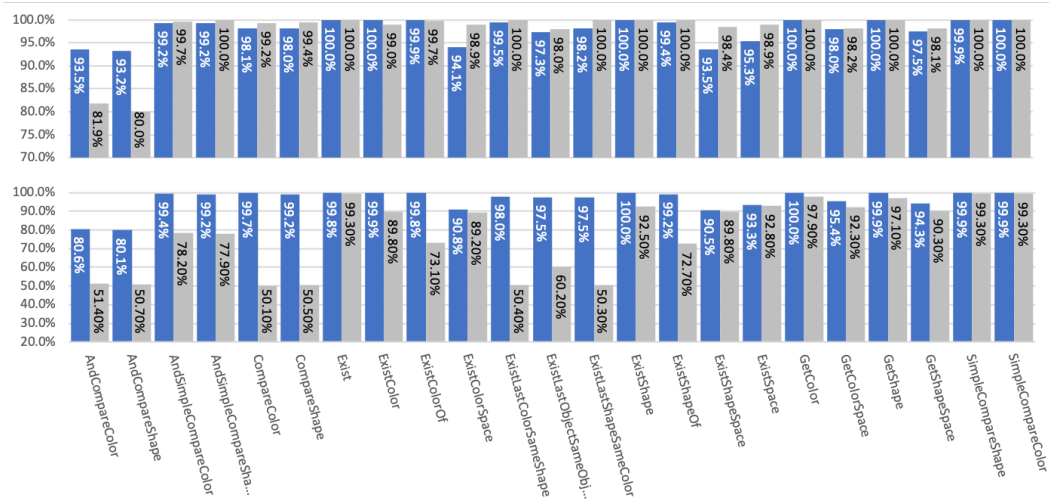


Figure 3: Comparison of test set accuracies of SAMNet (blue) with original results achieved by the COG model (gray) on Canonical (top) and Hard (bottom) variants of the COG dataset.

The most important results are highlighted in Figure 3, whereas full comparison with all accuracies can be found in Table 2 at the end of the section.

For the Canonical variant (top row), we have achieved similar accuracies for the majority of tasks (with the total average accuracy of 98.0% in comparison of 97.6% achieved by the COG model), with significant improvements (around 13 points) for *AndCompare* tasks. As those tasks focus on compositional questions referring to two objects, we hypothesize that our model achieved better accuracy due to the ability to selectively pick and store the relevant objects from the past frames in the memory. Despite there being some tasks in which our model reached slightly lower accuracies, when comparing performances on the Hard variant, it improves upon the COG baseline on all tasks, with improvements varying from 0.5 to more than 30 points.

3.2 Generalization capabilities

The goal of the next set of experiments was to test the generalization ability concerning the sequence length and number of distractors. For that purpose, we have compared the accuracies of both models when trained on the Canonical variant and tested on Hard (Figure 4). As the original paper does not include such experiments, we have performed them on our own. The light gray color indicates

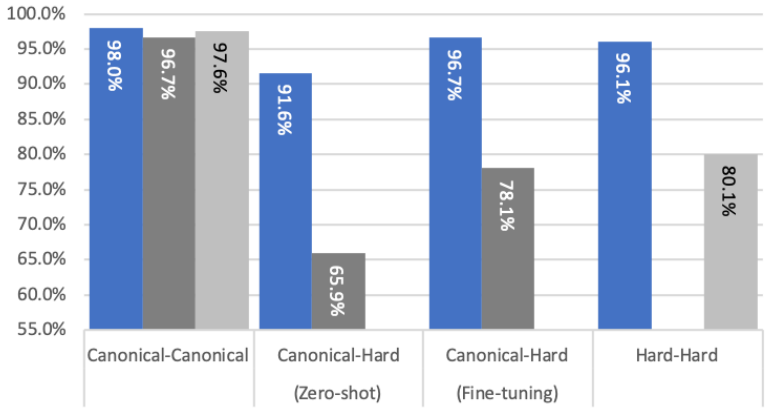


Figure 4: Total accuracies of SAMNet (blue) and COG models (light/dark gray) when testing generalization from Canonical to Hard variants of the dataset.

the original results, whereas dark gray indicates the accuracies of COG models that we have trained (fine-tuning/testing) using the original code provided by the authors. For sanity check, in the first column, we report both the best-achieved score and the score reported in the paper when training and testing on Canonical variant, without any fine-tuning. In a pure *transfer learning* setup (second column), our model shows enormous generalization ability, reaching 91.6% accuracy on the test set. We have also tested both models in a setup where the model trained on a Canonical variant underwent additional fine-tuning (for a single epoch) on the Hard variant (third column). In this case, the SAMNet model also reached much better performance, and, interestingly, achieved better scores from the model trained and tested exclusively on the Hard variant.

Table 2: COG test set accuracies for SAMNet & COG models. Below 'paper' denotes results from [25] while 'code' denotes results of our experiments using their implementation [4]

Model	SAMNet				COG			
					paper	ours	ours	paper
Trained on	canonical	canonical	canonical	hard	canonical	canonical	canonical	hard
Fine tuned on	-	-	hard	-	-	-	hard	-
Tested on	canonical	hard	hard	hard	canonical	hard	hard	hard
Overall accuracy	98.0	91.6	96.5	96.1	97.6	65.9	78.1	80.1
AndCompareColor	93.5	82.7	89.2	80.6	81.9	57.1	60.7	51.4
AndCompareShape	93.2	83.7	89.7	80.1	80.0	53.1	50.3	50.7
AndSimpleCompareColor	99.2	85.3	97.6	99.4	99.7	53.4	77.1	78.2
AndSimpleCompareShape	99.2	85.8	97.6	99.2	100.0	56.7	79.3	77.9
CompareColor	98.1	89.3	95.9	99.7	99.2	56.1	67.9	50.1
CompareShape	98.0	89.7	95.9	99.2	99.4	66.8	65.4	50.5
Exist	100.0	99.7	99.8	99.8	100.0	63.5	96.1	99.3
ExistColor	100.0	99.6	99.9	99.9	99.0	70.9	99	89.8
ExistColorOf	99.9	95.5	99.7	99.8	99.7	51.5	76.1	73.1
ExistColorSpace	94.1	88.8	91.0	90.8	98.9	72.8	77.3	89.2
ExistLastColorSameShape	99.5	99.4	99.4	98.0	100.0	65.0	62.5	50.4
ExistLastObjectSameObject	97.3	97.5	97.7	97.5	98.0	77.5	61.7	60.2
ExistLastShapeSameColor	98.2	98.5	98.8	97.5	100.0	87.8	60.4	50.3
ExistShape	100.0	99.5	100.0	100.0	100.0	77.1	98.2	92.5
ExistShapeOf	99.4	95.9	99.2	99.2	100.0	52.7	74.7	72.70
ExistShapeSpace	93.4	87.5	91.1	90.5	97.7	70	89.8	89.80
ExistSpace	95.3	89.7	93.2	93.3	98.9	71.1	88.1	92.8
GetColor	100.0	95.8	99.9	100.0	100.0	71.4	83.1	97.9
GetColorSpace	98.0	90.0	95.0	95.4	98.2	71.8	73.	92.3
GetShape	100.0	97.3	99.9	99.9	100.0	83.5	89.2	97.1
GetShapeSpace	97.5	89.4	93.9	94.3	98.1	78.7	77.3	90.3
SimpleCompareShape	99.9	91.4	99.7	99.9	100.0	67.7	96.7	99.3
SimpleCompareColor	100.0	91.6	99.80	99.9	100.0	64.2	90.4	99.3

3.3 Illustrating the multi-step reasoning and grounding of concepts including time

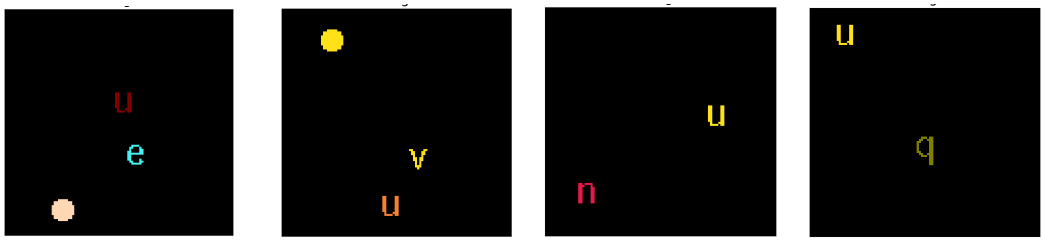
To illustrate the reasoning strategy developed by SAMNet including the grounding of concepts, we consider the example of **CompareColor** presented in Figure 5. As a full step-by-step analysis (8 reasoning steps for each of 4 frames) would be tedious, we picked the four key steps of reasoning performed by SAMNet for the same example that are important for Frame **4**. For more details on this example, please see Appendix A.

First, we analyze the **6**-th reasoning step of Frame **2**, presented Figure 6. Here the system clearly focuses its attention on the second object important from the point of view of the question: the **yellow circle**. Analogously, the question and visual attention are almost perfectly aligned with the word and object, besides, the system does a decent job of capturing the time context **Latest** of the object. Finally, the "Add New" gate is on, indicating that the system has stored the **yellow circle** in external memory.

Next, in the **1**-st reasoning step of Frame **4**, presented in Figure 7, we can once again observe that system correctly grounded the visual object **u** and detected the most likely temporal context **Now** (still not extremely high indicating that the distinction may not be significant for this example). Besides that, we can notice that memory content remained more or less unchanged.

In **6**-th reasoning step (Figure 8) there are several things worth discussing. First, question attention is pointing at the word **circle**, but the visual attention is clearly avoiding all the objects. This is because

Question: **Does color of u now equal the color of the latest circle?**



Frame: **1**
Ground Truth: **false**

Frame: **2**
Ground Truth: **false**

Frame: **3**
Ground Truth: **true**

Frame: **4**
Ground Truth: **true**

Figure 5: A sample from the COG dataset selected for the following analysis.

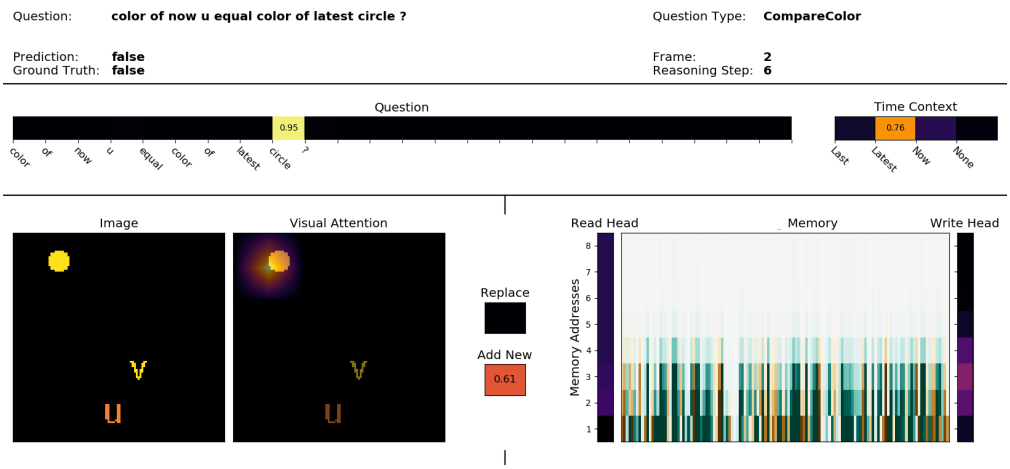


Figure 6: State of the SAM Cell after 6-th step of reasoning on Frame 2.

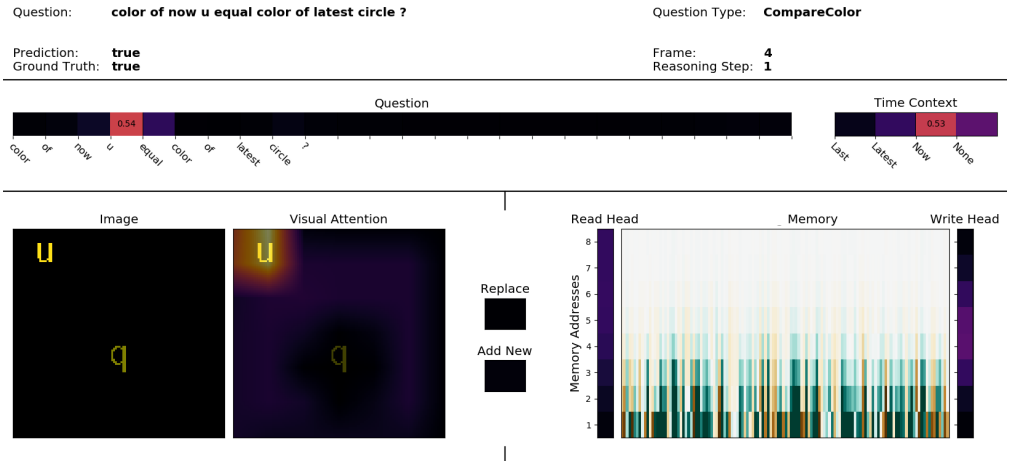


Figure 7: State of the SAM Cell after 1-st step of reasoning on Frame 4.

there is no circle. However, as the system properly identified the temporal context as **Latest**, instead of using the visual object, it uses the object retrieved from the memory—notice that the read head, despite not being perfectly crisp, is pointing at addresses 3-5, where it previously stored the **yellow circle** during the analysis of Frame **2**. Moreover, the "Add New" gate value is high enough so it once again updates the memory, with a void object. (This would have been considered for future frames if

they were to be present. Because the content of those memory addresses is negligible, SAMNet will still able return the correct prediction).

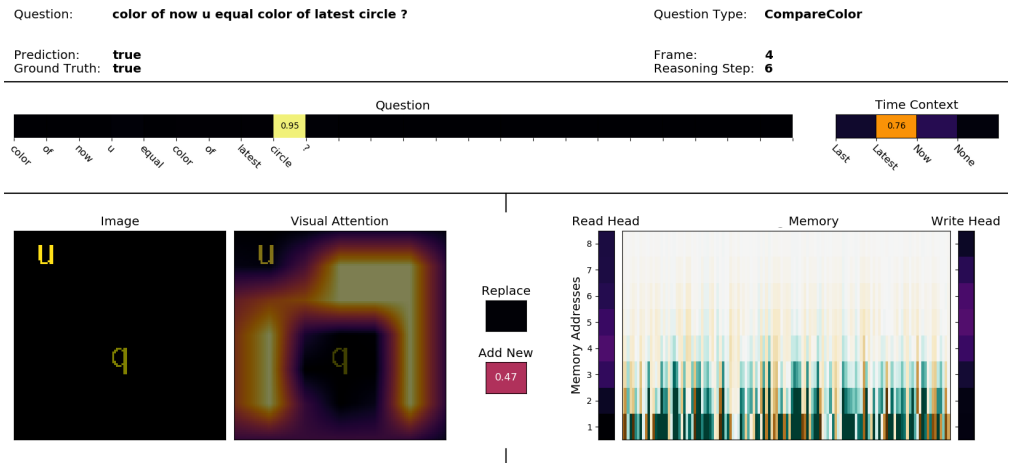


Figure 8: State of the SAM Cell after 6-th step of reasoning on Frame 4.

4 Conclusions

In this paper we have introduced a novel Memory-Augmented Neural Network model called SAMNet. SAMNet was designed with the goal of learning to reason over video sequences and to answer questions about the content of the frames. When applied to the COG video reasoning dataset, the model outperformed the baseline model, showing significant improvements on the Hard variant of the dataset. The results indicate that the mechanisms introduced in SAMNet enable it to operate independent of the total number of frames or the number of distractions, and allow it to generalize to longer videos and more complex scenes. Observing attention maps shows that SAMNet can effectively perform multi-step reasoning over questions and frames as intended. Despite being trained only on image-question pairs with complex, compositional questions, SAMNet clearly learns to associate visual symbols with words and accurately classify temporal contexts as designed. Besides, the model's reasoning using neural representations appears to be similar to how a human would operate on abstract symbols when solving the same task, including memorizing and recalling symbols (object embeddings) from the memory when needed. This is not perfect however and the system can sometimes store spurious objects despite the gating and reasoning mechanisms, but still give correct answers. This indicates at least two directions for possible further improvements. The first is to ameliorate content-based addressing with masking, similar to the improvements made for DNC proposed by [2]. Second is to implement variable number of reasoning steps, instead of hard-coded 8 steps, which could utilize Adaptive Computation Time (ACT) [5].

Acknowledgement

The authors would like to thank to the authors of COG paper (Igor Ganichev in particular) for sharing the detailed results with performances achieved by their COG baseline model.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015.

[2] R. Csordás and J. Schmidhuber. Improved addressing in the differentiable neural computer. 2019.

[3] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 326–335, 2017.

[4] I. Ganichev. Cog implementation. https://github.com/google/cog, 2018.

[5] A. Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

[6] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[7] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471, 2016.

[8] S. Harnad. Symbol grounding problem. 2003.

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

[10] T. Jayram, Y. Bouhadjar, R. L. McAvoy, T. Kornuta, A. Asseman, K. Rocki, and A. S. Ozcan. Learning to remember, forget and ignore using attention control in memory. *arXiv preprint arXiv:1809.11087*, 2018.

[11] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[12] T. Kornuta, V. Marois, R. L. McAvoy, Y. Bouhadjar, A. Asseman, V. Albouy, T. Jayram, and A. S. Ozcan. Accelerating machine learning research with mi-prometheus. In *NeurIPS Workshop on Machine Learning Open Source Software (MLOSS)*, volume 2018, 2018.

[13] M. Malinowski and M. 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.

[14] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision, 2019.

[15] V. Marois, T. Jayram, V. Albouy, T. Kornuta, Y. Bouhadjar, and A. S. Ozcan. On transfer learning using a MAC model variant. In *NeurIPS'18 Visually-Grounded Interaction and Language (ViGIL) Workshop*, 2018.

[16] A. Mogadala, M. Kalimuthu, and D. Klakow. Trends in integration of vision and language research: A survey of tasks, datasets, and methods. *arXiv preprint arXiv:1907.09358*, 2019.

[17] M. Monfort, A. Andonian, B. Zhou, K. Ramakrishnan, S. A. Bargal, Y. Yan, L. Brown, Q. Fan, D. Gutfreund, C. Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[18] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. 2017.

[19] X. Song, Y. Shi, X. Chen, and Y. Han. Explore multi-step reasoning in video question answering. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 239–247. ACM, 2018.

[20] T. Taniguchi, E. Ugur, M. Hoffmann, L. Jamone, T. Nagai, B. Rosman, T. Matsuka, N. Iwahashi, E. Oztop, J. Piater, et al. Symbol emergence in cognitive developmental systems: a survey. *IEEE Transactions on Cognitive and Developmental Systems*, 2018.

[21] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. MovieQA: Understanding Stories in Movies through Question-Answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[23] R. Vedantam, K. Desai, S. Lee, M. Rohrbach, D. Batra, and D. Parikh. Probabilistic neural-symbolic models for interpretable visual question answering. *arXiv preprint arXiv:1902.07864*, 2019.

[24] J. Weston, S. Chopra, and A. Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

[25] G. R. Yang, I. Ganichev, X.-J. Wang, J. Shlens, and D. Sussillo. A dataset and architecture for visual reasoning with a working memory. In *European Conference on Computer Vision*, pages 729–745. Springer, 2018.

A More details on the reasoning strategy

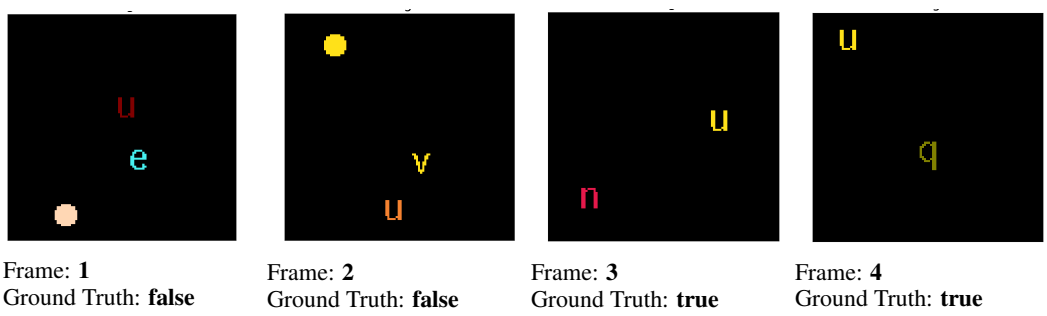


Figure 9: A sample from the COG dataset selected for the following analysis.

As a result of training the SAMNet developed a quite complex reasoning strategy. We will illustrate the key reasoning steps taken by the model on the example presented in Figure 9. We have decided to pick the sample from the **CompareColor** task.

First, let us analyze, independently of SAMNet, what are the key operations that one would need to perform in order to provide the correct answers. As the question concerns both the **now** and **latest** temporal contexts, in the Frame **1** one should look for both **u** and **circle** objects, look at their colors (different), and provide the answer **false**. Next, one should also memorize the **pink circle** and move to next Frame. Analogically, in Frame **2** there are both objects present, both with different colors, so the answer is once again **false**. However, from that point we do not care anymore about the **pink circle** and should remember the **yellow circle** instead. This is of utmost importance for the following two frames, as there are no circles present there. So when analyzing Frames **3** and **4** one need to recall the **yellow circle** from the memory and compare its color with the color of **u**'s – as in both cases they are yellow, thus both answers should be **true**.

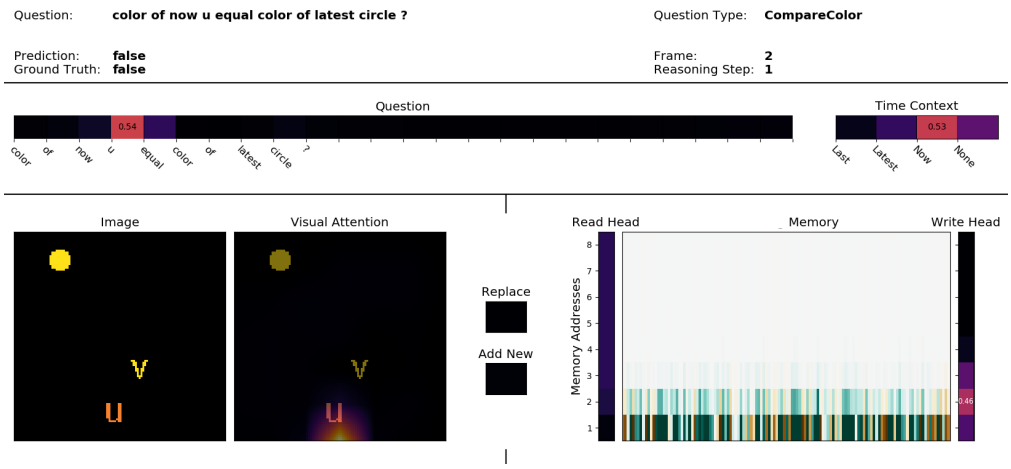


Figure 10: State of the SAM Cell after 1-st step of reasoning on Frame 2.

As full step-by-step analysis (8 reasoning steps for each of 4 frames) would significantly exceed the page limits, we have cherry picked four steps of reasoning performed by our SAMNet model. Let us start from **1**-st reasoning step on Frame **2**, as visualized in Figure 10 and focus on different aspects of the model. First, the attention over both the question and image is clearly focusing over the **u** – despite we have never trained the system with "word **u**"-"visual object **u**" pairs, the SAM Network managed to develop visual grounding on its own. Second, the time context is clearly focused on Now – once again, during the training we have never instructed that the system should pay attention to this (or any other) particular word and the system learned to catch the temporal context on its own (i.e. learned to use the provided mechanisms in the correct way). Third, as write head is always pointing

at "free memory slot", we can notice that there is an object already stored in memory under address 1. This is the **pink circle** object memorized during the analysis of the Frame **1**. However, we can also notice that the 2nd slot seems to be partially "polluted" by the object from slot 1, which indicates that the write head was not perfectly focused on a single memory address (similarly to the current state).

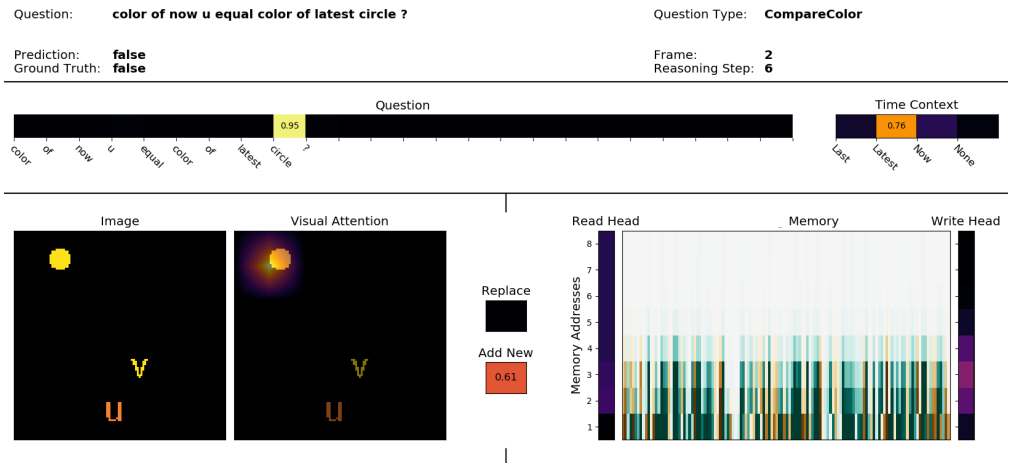


Figure 11: State of the SAM Cell after 6-th step of reasoning on Frame 2.

Next, let us analyze the **6**-th reasoning step presented Figure 11. Here the system clearly focuses its attention on the second object important from the point of view of the question: the **yellow circle**. Analogically, the question and visual attention are almost perfectly aligned with the word and object, and, besides, the system properly caught the time context of the object: **Latest**. Additionally, the "Add New" gate is on, the memory address 2 has a different content and write head moved to the next address. Those are clear indicators that the system has stored the **yellow circle** in the external memory. Sadly, also in here we observe the "pollution" of addresses 3 and 4.

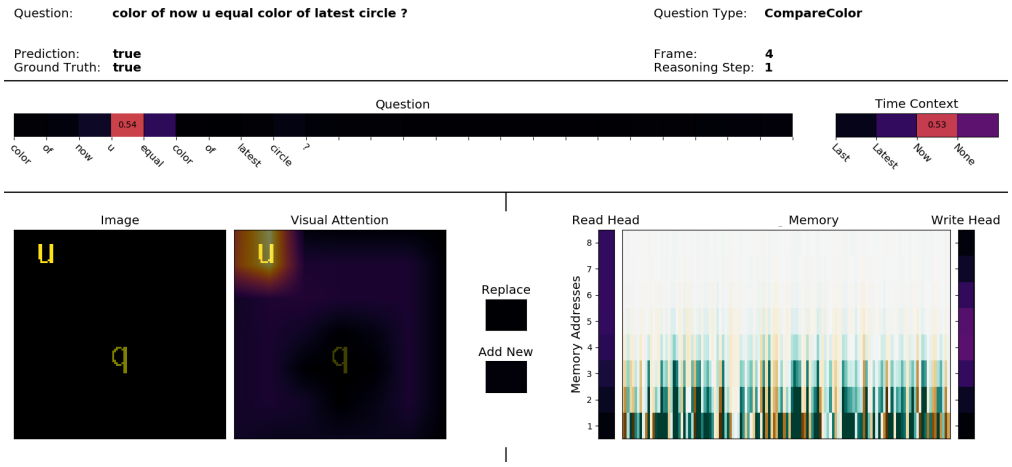


Figure 12: State of the SAM Cell after 1-st step of reasoning on Frame 4.

Let us now fast-forward to analogical reasoning steps in Frame **6**. In the **1**-st reasoning step (Figure 12) we can once again observe that system correctly grounded the visual object **u** and detected the correct temporal context **Now**. Besides that, we can notice that memory content remained more or less unchanged, despite the fact that write head shifted to the next memory address (it also became more soft, pointing at addresses 4 and 5).

In **6**-th reasoning step (Figure 13) there are several things worth discussing. First, question attention is pointing at the word **circle**, but the visual attention is clearly avoiding all the objects. This is because there is no circle. However, as the system properly identified the temporal context as **Latest**, instead

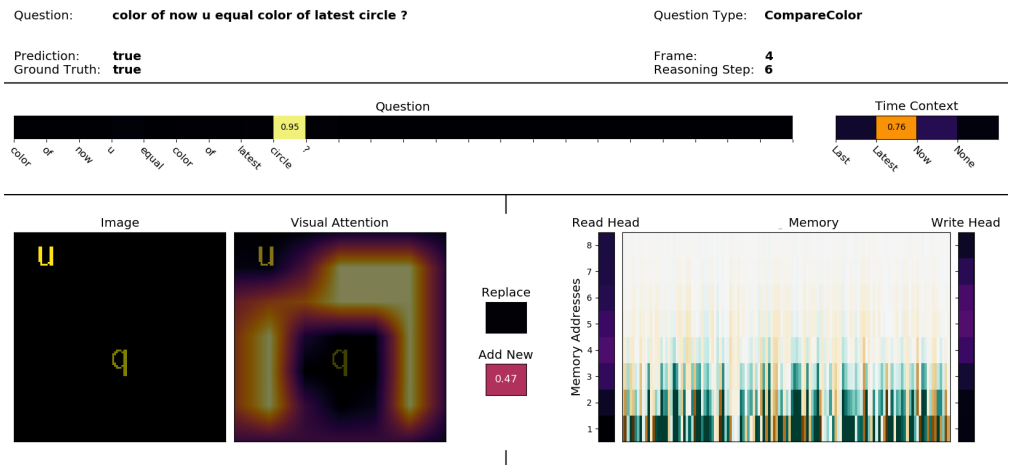


Figure 13: State of the SAM Cell after 6-th step of reasoning on Frame 4.

of using the visual object, it uses the object retrieved from the memory – please notice that the read head, despite not being perfectly crisp, is pointing at addresses 3-5, where it previously stored the **yellow circle** during the analysis of Frame **2**. Moreover, the "Add New" gate value is high enough so it once again updates the memory, with void object. But please notice that the content of those memory addresses is negligible, and at the end the SAMNet model is still able return the correct prediction.