Visually Grounded Video Reasoning in Selective Attention Memory

T.S. Jayram, Vincent Albouy, Tomasz Kornuta, 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. In analogy with human reasoning, we present Selective Attention Memory Network (SAMNet), an end-to-end differentiable recurrent model equipped with external memory. 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 to answer questions. We tested our model on the COG dataset (a multi-frame visual question answering test), and outperformed the state of the art baseline for hard tasks and in terms of generalization over video length and scene complexity.

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* [4] but we are quite far from achieving the full capabilities of visually grounded language learning. Starting with Image Question Answering [8, 1] and Image Captioning [6], a variety of tasks that integrate vision and language have emerged in the past several years [9]. Those directions include e.g., Video QA [13] and Video Action Recognition [10], that provide an additional challenge of understanding *temporal* aspects, and Video Reasoning [12, 14], 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 [12] 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 [2] 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.

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 [14] 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.

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

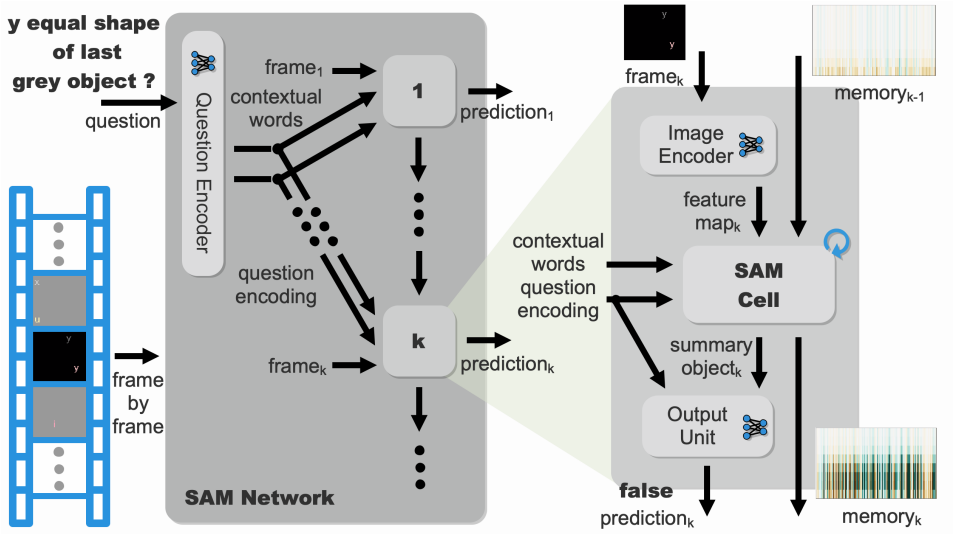

Figure 1: General architecture of SAMNet

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). The model 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.

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 [5]. 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

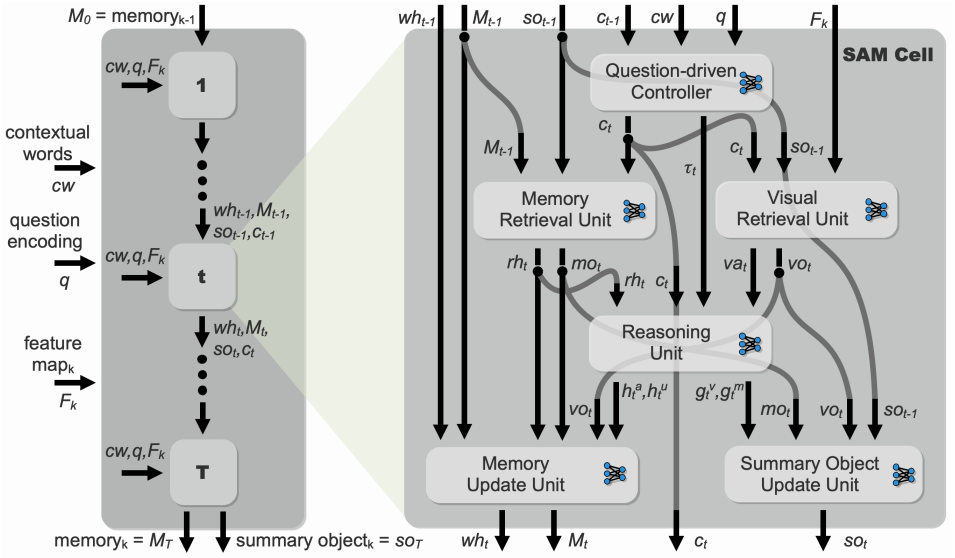

Figure 2: Unfolded reasoning steps with operations performed by the SAMCell

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) 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.

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

3 Experiments

We evaluated SAMNet on the COG dataset [14]. 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: 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 frame (1 vs. 10).

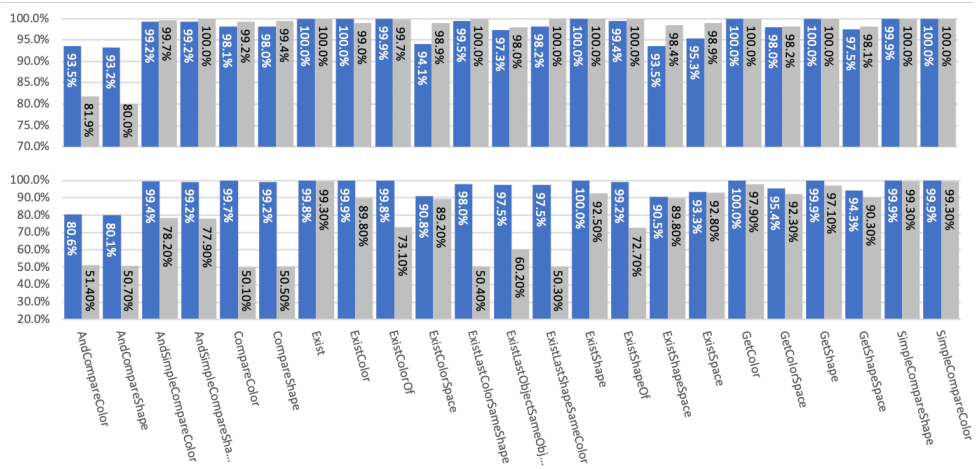

Figure 3: Comparison of test set accuracies of SAMNet (blue) with original results achieved by the COG model (gray) on Canonical (top) and Hard (bottom) variants of the COG dataset.

We have implemented and trained our SAMNet model using MI-Prometheus [7], a framework based on Pytorch [11]. In our experiments, we have focused on 22 classification tasks and compared our results with the baseline model, as presented in Figure 3. 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.

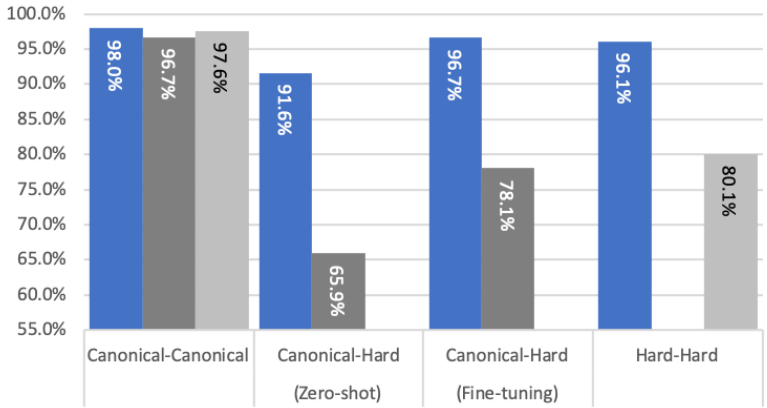

Figure 4: Total accuracies of SAMNet (blue) and COG models (light/dark gray) when testing generalization from Canonical to Hard variants of the dataset.

The 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 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. In summary, the results clearly indicate that the mechanisms introduced in SAMNet enable it to learn to operate independently of the total number of frames or number of distractions, and allow it to generalize to longer videos and more complex scenes. One other strength of SAMNet is its interpretability. Observing attention maps (see supplementary material) shows that SAMNet can effectively perform multi-step reasoning over questions and frames as intended. It also accurately classifies temporal contexts as designed. However we notice that the model can sometime discover alternative strategies that were not in the intended design but the answers are still correct.

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] 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.

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

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

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.