4 Supplementary Material

4.1 Training and Implementation Details

SAMNet is implemented on IBM's Mi-Prometheus [7] framework based on Pytorch. We trained all our models using NVIDIA's GeForce GTX TITAN X GPUs. SAMNet was trained using 8 reasoning steps and a hidden state size of 128. The external memory has 128-bit slots for all experiments. We trained our model until convergence but we also have set a training time limit of 80 hours.

4.1.1 Training and testing Methodology

We compared our model to the original COG model [14] using their implementation (https://github.com/google/cog) and scores provided by the authors through personal communications. We used the same training parameters detailed in the original paper and reproduced their results. For the generalization experiments from canonical to hard, we used the verified model and obtained new results that were not reported in the reference paper. In Table 1 COG section shows 4 columns divided into two parts: "paper" and "ours" which distinguish between the results reported in the paper vs. our own experiments.

Our experiments focused on the 22 classification tasks provided by the COG dataset. More details about the dataset are given in Table 2. First we evaluated SAMNet's performance on the canonical setting and compared it with the COG Model. As shown in Table 1 we could achieve a small improvement in accuracy, from 97.6% for the COG model to 98% for SAMNet. Next we focused on the hard setting of the dataset which increases the number of distractors from 1 to 10 and the number of frames from 4 to 8.

The first approach was to train a model on the hard training set, and test it on the hard test set. This is the same approach used by the COG paper [14] to evaluate performance on the hard dataset. We achieve a test accuracy of 96.1 % which represents a 16% improvement from the COG model score (see Table 1).

The second approach was to see if the models can generalize from the easy to the hard setting. For this experiment, we trained a model on the canonical dataset, and directly tested on the hard dataset. This experiment highlighted the most significant difference between SAMNet and the baseline COG model.

Finally we trained a model on the canonical data set, fine-tuned it on the hard data set using only 25k iterations, and tested on the hard dataset. Thanks to fine-tuning, we can observe a significant improvement from 91.6% to 96.5% test accuracy which represents the state of the art accuracy for the hard setting (classification tasks). After a short fine-tuning process, the transferred model could generalize well to harder tasks and even surpass the accuracy obtained in the first approach. We note that the third approach is also twice faster than the first one, and it is more effective in terms of accuracy.

A more granular analysis of accuracy per task shows a major improvement for the two hardest tasks, AndCompareShape and AndCompareColor. Those two tasks represent a higher level of difficulty due to the number of objects to be remembered in order to answer the question correctly. As we can see in Table 1 we could achieve a 12% improvement for the canonical data set and almost a 30% improvement for the hard dataset. The large improvement in these memory-intensive tasks indicate that the SAMNet's external memory plays a crucial role in our results. The training and implementation details are in appendix.

4.2 Visualization

We illustrate the key reasoning steps taken by the model with the following example. Let the question be: "Does color of u now equal the color of the latest circle?" and let the frames be as shown in Figure 5. The answer is true in the last frame because the latest circle can be found in Frame 2.

SAMNet gives the correct answer for this example and we can interpret its reasoning process via a movie that we can generate. The key steps in that reasoning process are shown below. Working backwards, the following shows what happens in the key steps of the reasoning process. In Step 1 of processing Frame 4 (see Figure 6), we see that the object representing the letter u is where the

Table 1: COG test set accuracies for SAMNet & COG models. Below 'paper' denotes results from [14] while 'code' denotes results of our experiments using their implementation [3]

Model	SAMNet				COG			
					paper	ours	ours	paper
Trained on Fine tuned on Tested on	canonical - canonical	canonical - hard	canonical hard hard	hard - hard	canonical - canonical	canonical - hard	canonical hard 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

Table 2: COG Dataset parameters for the canonical setting and the hard setting

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

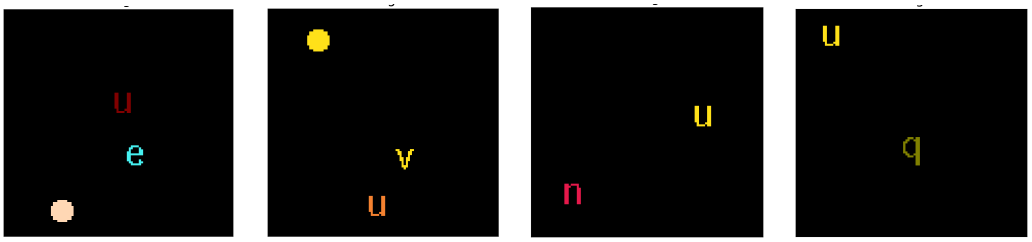

Figure 5: Sample sequence of input frames

attention is, both in text and in image. We also see that the temporal classifier gives the highest weight to the context "now".

Now there is no circle in this frame even though it is a valid candidate . This is detected by SAMNet, as shown in Figure 7. Even though the textual attention is on the right word and the temporal classification is "latest", there is no valid visual attention.

But in Step 6 of processing Frame 2 (see Figure 8), we see that the object representing the circle is where the attention is. Moreover, the "Add New" gate value is high enough that the valid object representing the visual encoding of the circle is stored in memory. Note that this is not perfect: this value should have been close to 1 but the model is still able to use it for correct prediction.

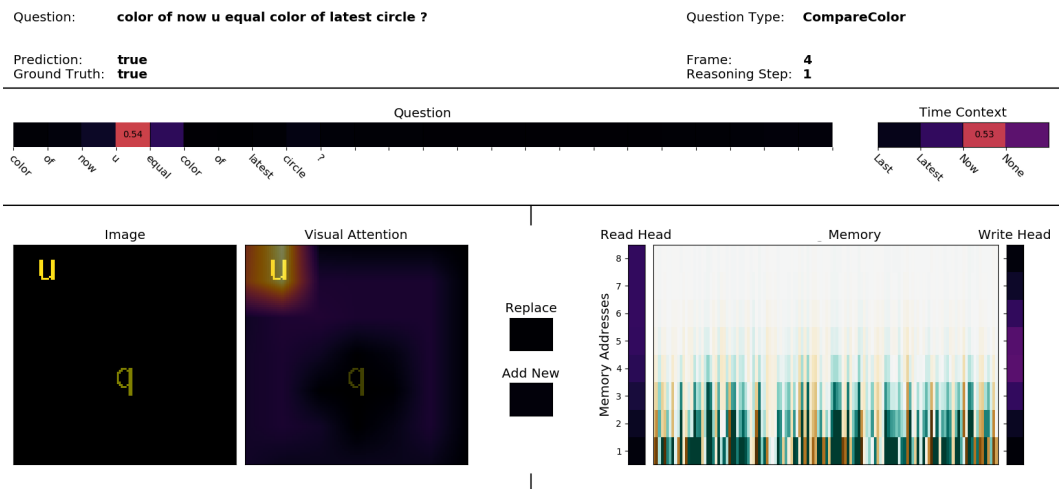

Figure 6: State of SAMNet in a particular reasoning step

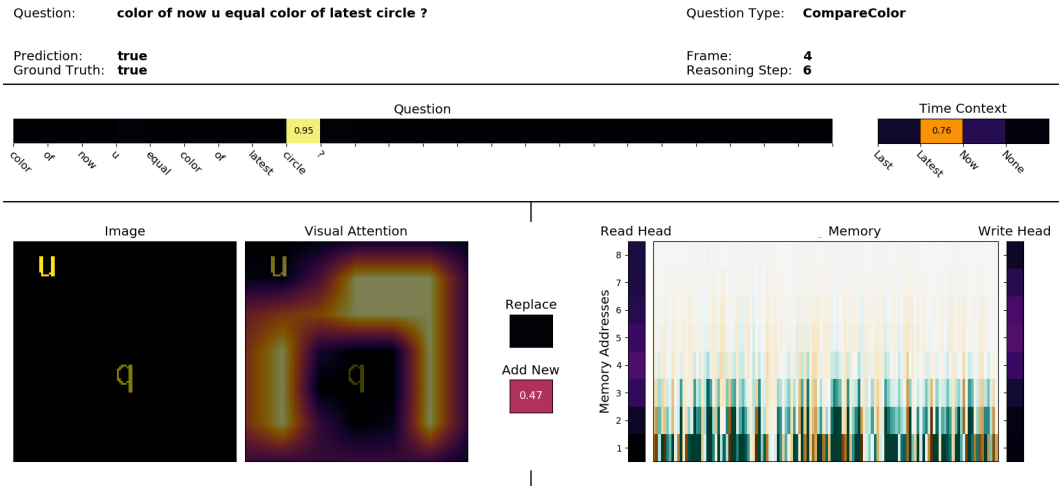

Figure 7: State of SAMNet in a particular reasoning step

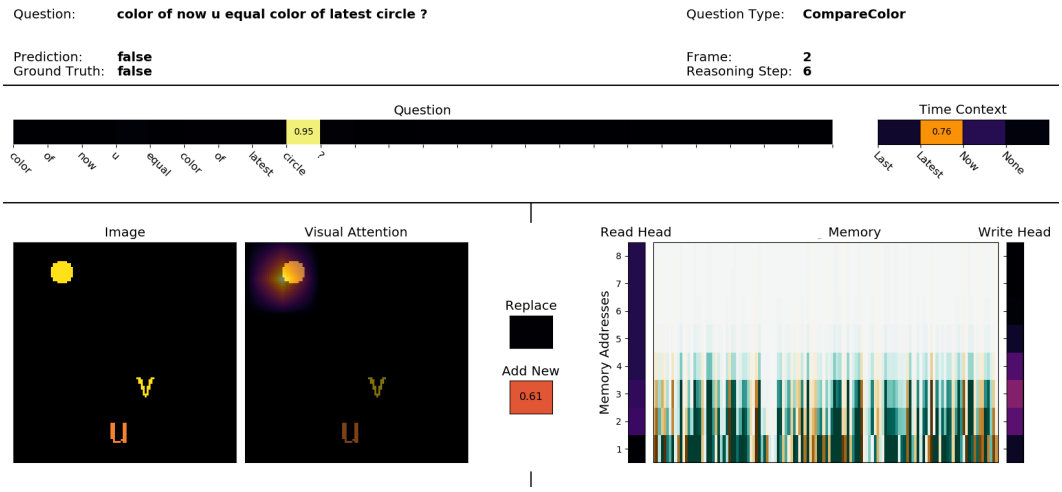

Figure 8: State of SAMNet in a particular reasoning step