A Dataset and Architecture for Visual Reasoning with a Working Memory

Guangyu Robert Yang[1,3(✉)], Igor Ganichev[2], Xiao-Jing Wang[1], Jonathon Shlens[2], and David Sussillo[2]

[1] Center for Neural Science, New York University, New York, USA
robert.yang@columbia.edu, xjwang@nyu.edu
[2] Google Brain, Mountain View, USA
iga@google.com, shlens@google.com, sussillo@google.com
[3] Department of Neuroscience, Columbia University, New York, USA

Abstract. A vexing problem in artificial intelligence is reasoning about events that occur in complex, changing visual stimuli such as in video analysis or game play. Inspired by a rich tradition of visual reasoning and memory in cognitive psychology and neuroscience, we developed an artificial, configurable visual question and answer dataset (COG) to parallel experiments in humans and animals. COG is much simpler than the general problem of video analysis, yet it addresses many of the problems relating to visual and logical reasoning and memory – problems that remain challenging for modern deep learning architectures. We additionally propose a deep learning architecture that performs competitively on other diagnostic VQA datasets (i.e. CLEVR) as well as easy settings of the COG dataset. However, several settings of COG result in datasets that are progressively more challenging to learn. After training, the network can zero-shot generalize to many new tasks. Preliminary analyses of the network architectures trained on COG demonstrate that the network accomplishes the task in a manner interpretable to humans.

Keywords: Visual reasoning · Visual question answering
Recurrent network · Working memory

1 Introduction

A major goal of artificial intelligence is to build systems that powerfully and flexibly reason about the sensory environment [1]. Vision provides an extremely rich and highly applicable domain for exercising our ability to build systems that

G. R. Yang—Work done as an intern at Google Brain.
G. R. Yang and I. Ganichev—Equal contribution.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01249-6_44) contains supplementary material, which is available to authorized users.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11214, pp. 729–745, 2018.
https://doi.org/10.1007/978-3-030-01249-6_44

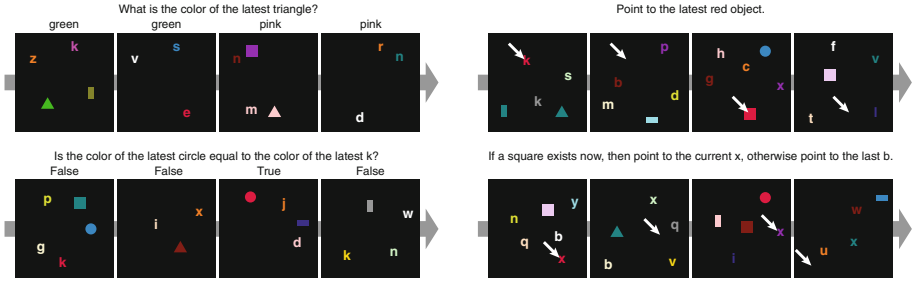

Fig. 1. Sample sequence of images and instruction from the COG dataset. Tasks in the COG dataset test aspects of object recognition, relational understanding and the manipulation and adaptation of memory to address a problem. Each task can involve objects shown in the current image and in previous images. Note that in the final example, the instruction involves the *last* instead of the *latest* "b". The former excludes the current "b" in the image. Target pointing response for each image is shown (white arrow). High-resolution image and proper English are used for clarity.

form logical inferences on complex stimuli [2–5]. One avenue for studying visual reasoning has been Visual Question Answering (VQA) datasets where a model learns to correctly answer challenging natural language questions about static images [6–9]. While advances on these multi-modal datasets have been significant, these datasets highlight several limitations to current approaches. First, it is uncertain the degree to which models trained on VQA datasets merely follow statistical cues inherent in the images, instead of reasoning about the logical components of a problem [10–13]. Second, such datasets avoid the complications of time and memory – both integral factors in the design of intelligent agents [1,14–16] and the analysis and summarization of videos [17–19].

To address the shortcomings related to logical reasoning about spatial relationships in VQA datasets, Johnson and colleagues [10] recently proposed CLEVR to directly test models for elementary visual reasoning, to be used in conjunction with other VQA datasets (e.g. [6–9]). The CLEVR dataset provides artificial, static images and natural language questions about those images that exercise the ability of a model to perform logical and visual reasoning. Recent work has demonstrated networks that achieve impressive performance with near perfect accuracy [4,5,20].

In this work, we address the second limitation concerning time and memory in visual reasoning. A reasoning agent must remember relevant pieces of its visual history, ignore irrelevant detail, update and manipulate a memory based on new information, and exploit this memory at later times to make decisions. Our approach is to create an artificial dataset that has many of the complexities found in temporally varying data, yet also to eschew much of the visual complexity and technical difficulty of working with video (e.g. video decoding, redundancy across temporally-smooth frames). In particular, we take inspiration from decades of research in cognitive psychology [21–25] and modern systems neuroscience (e.g.

[26–31]) – fields which have a long history of dissecting visual reasoning into core components based on spatial and logical reasoning, memory compositionality, and semantic understanding. Towards this end, we build an artificial dataset – termed COG – that exercises visual reasoning in time, in parallel with human cognitive experiments [32–34].

The COG dataset is based on a programmatic language that builds a battery of task triplets: an image sequence, a verbal instruction, and a sequence of correct answers. These randomly generated triplets exercise visual reasoning across a large array of tasks and require semantic comprehension of text, visual perception of each image in the sequence, and a working memory to determine the temporally varying answers (Fig. 1). We highlight several parameters in the programmatic language that allow researchers to modulate the problem difficulty from easy to challenging settings.

Finally, we introduce a multi-modal recurrent architecture for visual reasoning with memory. This network combines semantic and visual modules with a stateful *controller* that modulates visual attention and memory in order to correctly perform a visual task. We demonstrate that this model achieves near state-of-the-art performance on the CLEVR dataset. In addition, this network provides a strong baseline that achieves good performance on the COG dataset across an array of settings. Through ablation studies and an analysis of network dynamics, we find that the network employs human-interpretable, attention mechanisms to solve these visual reasoning tasks. We hope that the COG dataset, corresponding architecture, and associated baseline provide a helpful benchmark for studying reasoning in time-varying visual stimuli[1].

2 Related Work

It is broadly understood in the AI community that memory is a largely unsolved problem and there are many efforts underway to understand this problem, e.g. studied in [35–37]. The ability of sequential models to compute in time is notably limited by memory horizon and memory capacity [37] as measured in synthetic sequential datasets [38]. Indeed, a large constraint in training network models to perform generic Turing-complete operations is the ability to train systems that compute in time [37,39].

Developing computer systems that comprehend time-varying sequence of images is a prominent interest in video understanding [18,19,40] and intelligent video game agents [1,14,15]. While some attempts have used a feed-forward architecture (e.g. [14], baseline model in [16]), much work has been invested in building video analysis and game agents that contain a memory component [16,41]. These types of systems are often limited by the flexibility of network memory systems, and it is not clear the degree to which these systems reason based on complex relationships from past visual imagery.

[1] The COG dataset and code for the network architecture are open-sourced at https://github.com/google/cog.

Let us consider Visual Question Answering (VQA) datasets based on single, static images [6–9]. These datasets construct natural language questions to probe the logical understanding of a network about natural images. There has been strong suggestion in the literature that networks trained on these datasets focus on statistical regularities for the prediction tasks, whereby a system may "cheat" to superficially solve a given task [10,11]. Towards that end, several researchers proposed to build an auxiliary diagnostic, synthetic datasets to uncover these potential failure modes and highlight logical comprehension (e.g. attribute identification, counting, comparison, multiple attention, and logical operations) [10,13,42–44]. Further, many specialized neural network architectures focused on multi-task learning have been proposed to address this problem by leveraging attention [45], external memory [35,36], a family of feature-wise transformations [5,46], explicitly parsing a task into executable sub-tasks [2,3], and inferring relations between pairs of objects [4].

Our contribution takes direct inspiration from this previous work on single images but focuses on the aspects of time and memory. A second source of inspiration is the long line of cognitive neuroscience literature that has focused on developing a battery of sequential visual tasks to exercise and measure specific attributes of visual working memory [21,26,47]. Several lines of cognitive psychology and neuroscience have developed multitudes of visual tasks in time that exercise attribute identification, counting, comparison, multiple attention, and logical operations [26,28–34] (see references therein). This work emphasizes compositionality in task generation – a key ingredient in generalizing to unseen tasks [48]. Importantly, this literature provides measurements in humans and animals on these tasks as well as discusses the biological circuits and computations that may underlie and explain the variability in performance [27–31].

3 The COG Dataset

We designed a large set of tasks that requires a broad range of cognitive skills to solve, especially working memory. One major goal of this dataset is to build a compositional set of tasks that include variants of many cognitive tasks studied in humans and other animals [26,28–34] (see also Introduction and Related Work).

The dataset contains triplets of a task instruction, sequences of synthetic images, and sequences of target responses (see Fig. 1 for examples). Each image consists of a number of simple objects that vary in color, shape, and location. There are 19 possible colors and 33 possible shapes (6 geometric shapes and 26 lower-case English letters). The network needs to generate a verbal or pointing response for every image.

To build a large set of tasks, we first describe all potential tasks using a common, unified framework. Each task in the dataset is defined abstractly and constructed compositionally from basic building blocks, namely *operators*. An operator performs a basic computation, such as selecting an object based on attributes (color, shape, etc.) or comparing two attributes (Fig. 2A). The operators are defined abstractly without specifying the exact attributes involved.

A task is formed by a directed acyclic graph of operators (Fig. 2B). Finally, we instantiate a task by specifying all relevant attributes in its graph (Fig. 2C). The task instance is used to generate both the verbal task instruction and minimally-biased image sequences. Many image sequences can be generated from the same task instance.

There are 8 operators, 44 tasks, and more than 2 trillion possible task instances in the dataset (see Appendix for more sample task instances). We vary the number of images (F), the maximum memory duration ($M_{\max}$), and the maximum number of distractors on each image ($D_{\max}$) to explore the memory and capacity of our proposed model and systematically vary the task difficulty. When not explicitly stated, we use a canonical setting with $F = 4$, $M_{\max} = 3$, and $D_{\max} = 1$ (see Appendix for the rationale).

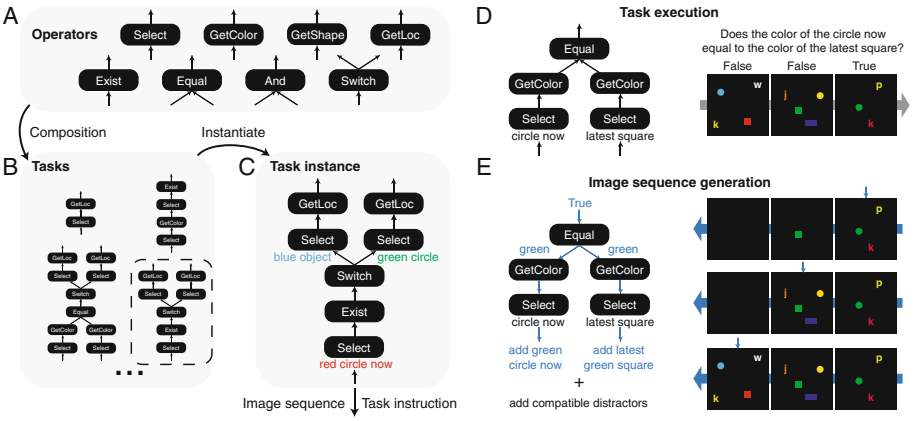

Fig. 2. Generating the compositional COG dataset. The COG dataset is based on a set of operators (A), which are combined to form various task graphs (B). (C) A task is instantiated by specifying the attributes of all operators in its graph. A task instance is used to generate both the image sequence and the semantic task instruction. (D) Forward pass through the graph and the image sequence for normal task execution. (E) Generating a consistent, minimally biased image sequence requires a backward pass through the graph in a reverse topological order and through the image sequence in the reverse chronological order.

The COG dataset is in many ways similar to the CLEVR dataset [10]. Both contain synthetic visual inputs and tasks defined as operator graphs (functional programs). However, COG differs from CLEVR in two important ways. First, all tasks in the COG dataset can involve objects shown in the past, due to the sequential nature of their inputs. Second, in the COG dataset, visual inputs with minimal response bias can be generated on the fly.

An operator is a simple function that receives and produces abstract data types such as an attribute, an object, a set of objects, a spatial range, or a Boolean. There are 8 operators in total: *Select*, *GetColor*, *GetShape*, *GetLoc*,

Exist, *Equal*, *And*, and *Switch* (see Appendix for details). Using these 8 operators, the COG dataset currently contains 44 tasks, with the number of operators in each task graph ranging from 2 to 11. Each task instruction is obtained from a task instance by traversing the task graph and combining pieces of text associated with each operator. It is straightforward to extend the COG dataset by introducing new operators.

Response bias is a major concern when designing a synthetic dataset. Neural networks may achieve high accuracy in a dataset by exploiting its bias. Rejection sampling can be used to ensure an *ad hoc* balanced response distribution [10]. We developed a method for the COG dataset to generate minimally-biased synthetic image sequences tailored to individual tasks.

In short, we first determine the minimally-biased responses (target outputs), then we generate images (inputs) that would lead to these specified responses. The images are generated in the reversed order of normal task execution (Fig. 2D, E). During generation, images are visited in the reverse chronological order and the task graph traversed in a reverse topological order (Fig. 2E). When visiting an operator, if its target output is not already specified, we randomly choose one from all allowable outputs. Based on the specified output, the image is modified accordingly and/or the supposed input is passed on to the next operator(s) as their target outputs (see details in Appendix). In addition, we can place a uniformly-distributed $D \sim U(1, D_{\max})$ distractors, then delete those that interfere with the normal task execution.

4 The Network

4.1 General Network Setup

Overall, the network contains four major systems (Fig. 3). The visual system processes the images. The semantic system processes the task instructions. The visual short-term memory system maintains the processed visual information, and provides outputs that guide the pointing response. Finally, the control system integrates converging information from all other systems, uses several attention and gating mechanisms to regulate how other systems process inputs and generate outputs, and provides verbal outputs. Critically, the network is allowed multiple time steps to "ponder" about each image [49], giving it the potential to solve multi-step reasoning problems naturally through iteration.

4.2 Visual Processing System

The visual system processes the raw input images. The visual inputs are 112×112 images and are processed by 4 convolutional layers with 32, 64, 64, 128 feature maps respectively. Each convolutional layer employs 3×3 kernels and is followed by a 2×2 max-pooling layer, batch-normalization [50], and a rectified-linear activation function. This simple and relatively shallow architecture was shown to be sufficient for the CLEVR dataset [4,10].

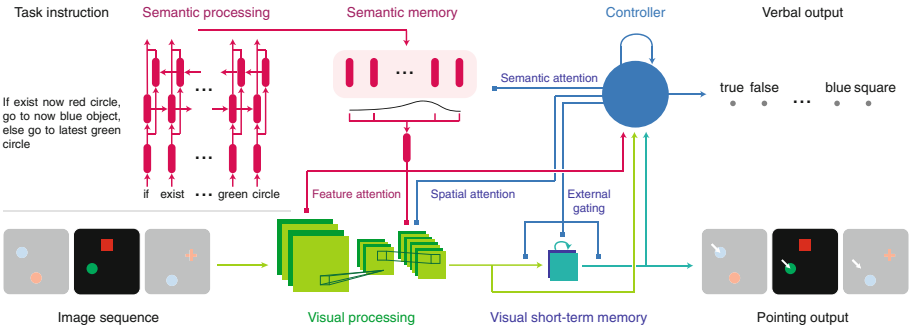

Fig. 3. Diagram of the proposed network. A sequence of images are provided as input into a convolutional neural network (green). An instruction in the form of English text is provided into a sequential embedding network (red). A visual short-term memory (vSTM) network holds visual-spatial information in time and provides the pointing output (teal). The vSTM module can be considered a convolutional LSTM network with external gating. A stateful controller (blue) provides all attention and gating signals directly or indirectly. The output of the network is either discrete (verbal) or 2D continuous (pointing). (Color figure online)

The last two layers of the convolutional network are subject to feature and spatial attention. Feature attention scales and shifts the batch normalization parameters of individual feature maps, such that the activity of all neurons within a feature map are multiplied and added by two scalars. This particular implementation of feature attention has been termed conditional batch-normalization or feature-wise linear modulation (FiLM) [5,46]. FiLM is a critical component for the model that achieved near state-of-the-art performance on the CLEVR dataset [5]. Soft spatial attention [51] is applied to the top convolutional layer following feature attention and the activation function. It multiplies the activity of all neurons with the same spatial preferences using a positive scalar.

4.3 Semantic Processing System

The semantic processing system receives a task instruction and generates a semantic memory that the controller can later attend to. Conceptually, it produces a semantic memory – a contextualized representation of each word in the instruction – before the task is actually being performed. At each pondering step when performing the task, the controller can attend to individual parts of the semantic memory corresponding to different words or phrases.

Each word is mapped to a 64-dimensional trainable embedding vector, then sequentially fed into an 128-unit bidirectional Long Short-Term Memory (LSTM) network [38,52]. The outputs of the bidirectional LSTM for all words form a semantic memory of size $(n_{\text{word}}, n_{\text{rule}}^{(\text{out})})$, where n_{word} is the number of words in the instruction, and $n_{\text{rule}}^{(\text{out})} = 128$ is the dimension of the output vector.

Each $n_{\text{rule}}^{(\text{out})}$-dimensional vector in the semantic memory forms a key. For semantic attention, a query vector of the same dimension $n_{\text{rule}}^{(\text{out})}$ is used to retrieve the semantic memory by summing up all the keys weighted by their similarities to the query. We used Bahdanau attention [53], which computes the similarity between the query $\mathbf{q}$ and a key $\mathbf{k}$ as $\sum_{i=1}^{n_{\text{rule}}^{(\text{out})}} v_i \cdot \tanh(q_i + k_i)$, where $\mathbf{v}$ is trained.

4.4 Visual Short-Term Memory System

To utilize the spatial information preserved in the visual system for the pointing output, the top layer of the convolutional network feeds into a visual short-term memory module, which in turn projects to a group of pointing output neurons. This structure is also inspired by the posterior parietal cortex in the brain that maintains visual-spatial information to guide action [54].

The visual short-term memory (vSTM) module is an extension of a 2-d convolutional LSTM network [55] in which the gating mechanisms are conditioned on external information. The vSTM module consists of a number of 2-D feature maps, while the input and output connections are both convolutional. There is currently no recurrent connections within the vSTM module besides the forget gate. The state c_t and output h_t of this module at step t are

$$c_t = f_t * c_{t-1} + i_t * x_t, \tag{1}$$

$$h_t = o_t * \tanh(c_t), \tag{2}$$

where * indicates a convolution. This vSTM module differs from a convolutional LSTM network mainly in that the input i_t, forget f_t, and output gates o_t are not self-generated. Instead, they are all provided externally from the controller. In addition, the input x_t is not directly fed into the network, but a convolutional layer can be applied in between.

All convolutions are currently set to be 1×1. Equivalently, each feature map of the vSTM module adds its gated previous activity with a weighted combination of the post-attention activity of all feature maps from the top layer of the visual system. Finally, the activity of all vSTM feature maps is combined to generate a single spatial output map h_t.

4.5 Controller

To synthesize information across the entire network, we include a controller that receives feedforward inputs from all other systems and generates feedback attention and gating signals. This architecture is further inspired by the prefrontal cortex of the brain [27]. The controller is a Gated Recurrent Unit (GRU) network. At each pondering step, the post-attention activity of the top visual layer is processed through a 128-unit fully connected layer, concatenated with the retrieved semantic memory and the vSTM module output, then fed into the controller. In addition, the activity of the top visual layer is summed up across space and provided to the controller.

The controller generates queries for the semantic memory through a linear feedforward network. The retrieved semantic memory then generates the feature attention through another linear feedforward network. The controller generates the 49-dimensional soft spatial attention through a two layer feedforward network, with a 10-unit hidden layer and a rectified-linear activation function, followed by a softmax normalization. Finally, the controller state is concatenated with the retrieved semantic memory to generate the input, forget, and output gates used in the vSTM module through a linear feedforward network followed by a sigmoidal activation function.

4.6 Output, Loss, and Optimization

The verbal output is a single word, and the pointing output is the (x, y) coordinates of pointing. Each coordinate is between 0 and 1. A loss function is defined for each output, and only one loss function is used for every task. The verbal output uses a cross-entropy loss. To ensure the pointing output loss is comparable in scale to the verbal output loss, we include a group of pointing output neurons on a 7×7 spatial grid, and compute a cross-entropy loss over this group of neurons. Given a target (x, y) coordinates, we use a Gaussian distribution centered at the target location with $\sigma = 0.1$ as the target probability distribution of the pointing output neurons.

For each image, the loss is based on the output at the last pondering step. No loss is used if there is no valid output for a given image. We use a L2 regularization of strength 2e-5 on all the weights. We clip the gradient norm at 10 for `COG` and at 80 for CLEVR. We clip the controller state norm at 10000 for `COG` and 5000 for CLEVR. We also trained all initial states of the recurrent networks. The network is trained end-to-end with Adam [56], combined with a learning rate decay schedule.

5 Results

5.1 Intuitive and Interpretable Solutions on the CLEVR Dataset

To demonstrate the reasoning capability of our proposed network, we trained it on the CLEVR dataset [10], even though there is no explicit need for working memory in CLEVR. The network achieved an overall test accuracy of 96.8% on CLEVR, surpassing human-level performance and comparable with other state-of-the-art methods [4,5,20] (Table 1, see Appendix for more details).

Images were first resized to 128×128, then randomly cropped or resized to 112×112 during training and validation/testing respectively. In the best-performing network, the controller used 12 pondering steps per image. Feature attention was applied to the top two convolutional layers. The vSTM module was disabled since there is no pointing output.

The output of the network is human-interpretable and intuitive. In Fig. 4, we illustrate how the verbal output and various attention signals evolved through

Table 1. CLEVR test accuracies for human, baseline, and top-performing models that relied only on pixel inputs and task instructions during training. (*) denotes use of pretrained models.

Model	Overall	Count	Exist	Compare numbers	Query attribute	Compare attribute
Human [10]	92.6	86.7	96.6	86.5	95.0	96.0
Q-type baseline [10]	41.8	34.6	50.2	51.0	36.0	51.3
CNN+LSTM+SA [4]	76.6	64.4	82.7	77.4	82.6	75.4
CNN+LSTM+RN [4]	95.5	90.1	97.8	93.6	97.9	97.1
CNN+GRU+FiLM [5]	97.6	94.3	99.3	93.4	99.3	99.3
MAC* [20]	98.9	97.2	99.5	99.4	99.3	99.5
Our model	96.8	91.7	99.0	95.5	98.5	98.8

pondering steps for an example image-question pair. The network answered a long question by decomposing it into small, executable steps. Even though training only relies on verbal outputs at the last pondering steps, the network learned to produce interpretable verbal outputs that reflect its reasoning process.

In Fig. 4, we computed effective feature attention as the difference between the normalized activity maps with or without feature attention. To get the post- (or pre-) feature-attention normalized activity map, we average the activity across all feature maps after (or without) feature attention, then divide the activity by its mean. The relative spatial attention is normalized by subtracting the time-averaged spatial attention map. This example network uses 8 pondering steps.

5.2 Training on the COG Dataset

Our proposed model achieved a maximum overall test accuracy of 93.7% on the COG dataset in the canonical setting (see Sect. 3). In the Appendix, we discuss potential strategies for measuring human accuracy on the COG dataset. We noticed a small but significant variability in the final accuracy even for networks with the same hyperparameters (mean $\pm$ std: $90.6 \pm 2.8\%$, 50 networks). We found that tasks containing more operators tend to take substantially longer to be learned or remain at lower accuracy (see Appendix for more results). We tried many approaches of reducing variance including various curriculum learning regimes, different weight and bias initializations, different optimizers and their hyperparameters. All approaches we tried either did not significantly reduce the variance or degraded performance.

The best network uses 5 pondering steps for each image. Feature attention is applied to the top layer of the visual network. The vSTM module contains 4 feature maps.

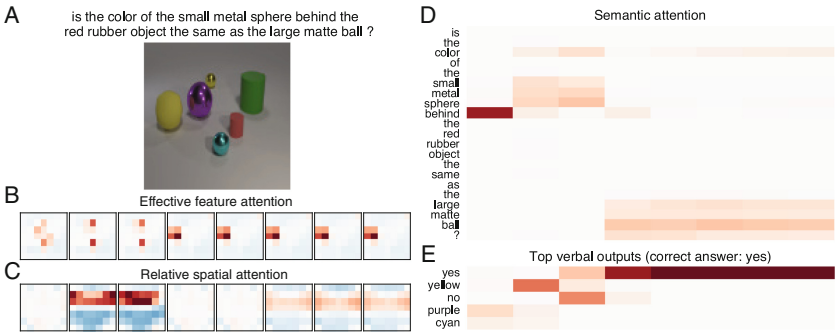

Fig. 4. Pondering process of the proposed network, visualized through attention and output for a single CLEVR example. (A) The example question and image from the CLEVR validation set. (B) The effective feature attention map for each pondering step. (C) The relative spatial attention maps. (D) The semantic attention. (E) Top five verbal outputs. Red and blue indicate stronger and weaker, respectively. After simultaneous feature attention to the "small metal spheres" and spatial attention to "behind the red rubber object", the color of the attended object (yellow) was reflected in the verbal output. Later in the pondering process, the network paid feature attention to the "large matte ball", while the correct answer (yes) emerged in the verbal output. (Color figure online)

5.3 Assessing the Contribution of Model Parts Through Ablation

The model we proposed contains multiple attention mechanisms, a short-term memory module, and multiple pondering steps. To assess the contribution of each component to the overall accuracy, we trained versions of the network on the CLEVR and the COG dataset in which one component was ablated from the full network. We also trained a baseline network with all components ablated. The baseline network still contains a CNN for visual processing, a LSTM network for semantic processing, and a GRU network as the controller. To give each ablated network a fair chance, we re-tuned their hyperparameters, with the total number of parameters limited at 110% of the original network, and reported the maximum accuracy.

We found that the baseline network performed poorly on both datasets (Fig. 5A, B). To our surprise, the network relies on a different combination of mechanisms to solve the CLEVR and the COG dataset. The network depends strongly on feature attention for CLEVR (Fig. 5A), while it depends strongly on spatial attention for the COG dataset (Fig. 5B). One possible explanation is that there are fewer possible objects in CLEVR (96 combinations compared to 608 combinations in COG), making feature attention on $\sim$ 100 feature maps better suited to select objects in CLEVR. Having multiple pondering steps is important for both datasets, demonstrating that it is beneficial to solve multi-step reasoning problems through iteration. Although semantic attention has a rather minor impact on the overall accuracy of both datasets, it is more useful for tasks with more operators and longer task instructions (Fig. 5C).

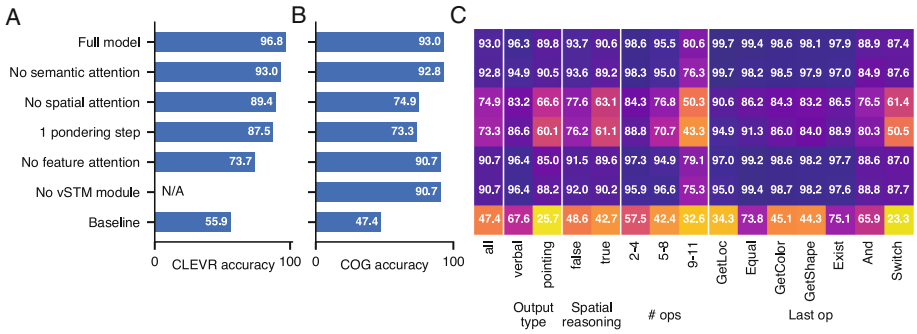

Fig. 5. Ablation studies. Overall accuracies for various ablation models on the CLEVR test set (A) and COG (B). vSTM module is not included in any model for CLEVR. (C) Breaking the COG accuracies down based on the output type, whether spatial reasoning is involved, the number of operators, and the last operator in the task graph.

5.4 Exploring the Range of Difficulty of the COG Dataset

To explore the range of difficulty in visual reasoning in our dataset, we varied the maximum number of distractors on each image ($D_{\max}$), the maximum memory duration ($M_{\max}$), and the number of images in each sequence (F) (Fig. 6). For each setting we selected the best network across 50–80 hyper-parameter settings involving model capacity and learning rate schedules. Out of all models explored, the accuracy of the best network drops substantially with more distractors. When there is a large number of distractors, the network accuracy also drops with longer memory duration. These results suggest that the network has difficulty filtering out many distractors and maintaining memory at the same time. However, doubling the number of images does not have a clear effect on the accuracy, which indicates that the network developed a solution that is invariant to the number of images used in the sequence. The harder setting of the COG dataset with $F = 8$, $D_{\max} = 10$ and $M_{\max} = 7$ can potentially serve as a benchmark for more powerful neural network models.

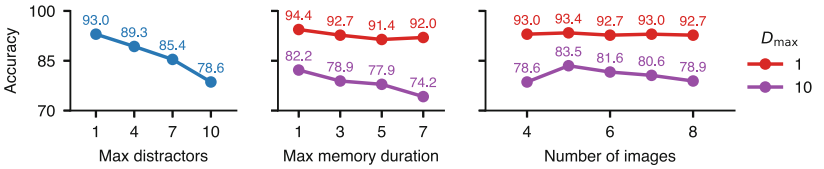

Fig. 6. Accuracies on variants of the COG dataset. From left to right, varying the maximum number of distractors ($D_{\max}$), the maximum memory duration ($M_{\max}$), and the number of images in each sequence (F).

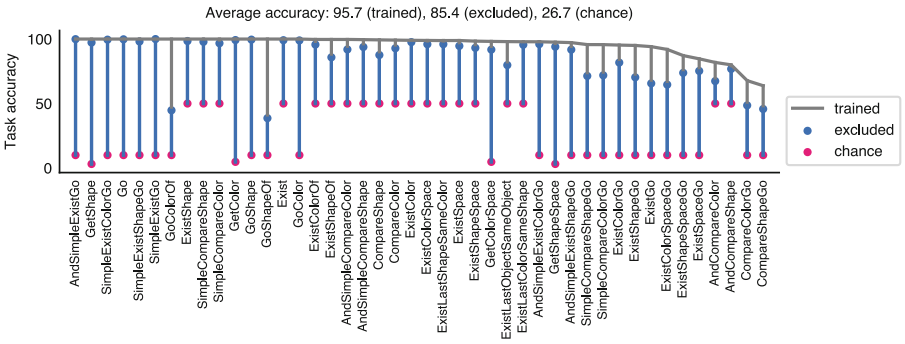

Fig. 7. The proposed network can zero-shot generalize to new tasks. 44 networks were trained on 43 of 44 tasks. Shown are the maximum accuracies of the networks on the 43 trained tasks (gray), the one excluded (blue) task, and the chance levels for that task (red). (Color figure online)

5.5 Zero-Shot Generalization to New Tasks

A hallmark of intelligence is the flexibility and capability to generalize to unseen situations. During training and testing, each image sequence is generated anew, therefore the network is able to generalize to unseen input images. On top of that, the network can generalize to trillions of task instances (new task instructions), although only millions of them are used during training.

The most challenging form of generalization is to completely new tasks not explicitly trained on. To test whether the network can generalize to new tasks, we trained 44 groups of networks. Each group contains 10 networks and is trained on 43 out of 44 COG tasks. We monitored the accuracy of all tasks. For each task, we report the highest accuracy across networks. We found that networks are able to immediately generalize to most untrained tasks (Fig. 7). The average accuracy for tasks excluded during training (85.4%) is substantially higher than the average chance level (26.7%), although it is still lower than the average accuracy for trained tasks (95.7%). Hence, our proposed model is able to perform zero-shot generalization across tasks with some success although not matching the performance as if trained on the task explicitly.

5.6 Clustering and Compositionality of the Controller Representation

To understand how the network is able to perform COG tasks and generalize to new tasks, we carried out preliminary analyses studying the activity of the controller. One suggestion is that networks can perform many tasks by engaging clusters of units, where each cluster supports one operation [57]. To address this question, we examined low-dimensional representations of the activation space of the controller and labeled such points based on the individual tasks. Figure 8A and B highlight the clustering behavior across tasks that emerges from training on the COG dataset (see Appendix for details).

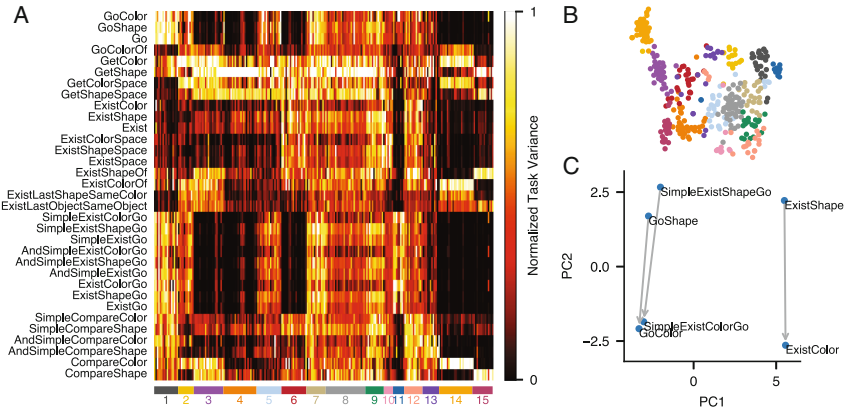

Fig. 8. Clustering and compositionality in the controller. (A) The level of task involvement for each controller unit (columns) in each task (rows). The task involvement is measured by task variance, which quantifies the variance of activity across different inputs (task instructions and image sequences) for a given task. For each unit, task variances are normalized to a maximum of 1. Units are clustered (bottom color bar) according to task variance vectors (columns). Only showing tasks with accuracy higher than 90%. (B) t-SNE visualization of task variance vectors for all units, colored by cluster identity. (C) Example compositional representation of tasks. We compute the state-space representation for each task as its mean controller activity vector, obtained by averaging across many different inputs for that task. The representation of 6 tasks are shown in the first two principal components. The vector in the direction of PC2 is a shared direction for altering a task to change from *Shape* to *Color* (Color figure online).

Previous work has suggested that humans may flexibly perform new tasks by representing learned tasks in a compositional manner [48,57]. For instance, the analysis of semantic embeddings indicates that network may learn shared directions for concepts across word embeddings [58]. We searched for signs of compositional behavior by exploring if directions in the activation space of the controller correspond to common sub-problems across tasks. Figure 8C highlights a direction that was identified that corresponds to axis of `Shape` to `Color` across multiple tasks. These results provide a first step in understanding how neural networks can understand task structures and generalize to new tasks.

6 Conclusions

In this work, we built a synthetic, compositional dataset that requires a system to perform various tasks on sequences of images based on English instructions. The tasks included in our `COG` dataset test a range of cognitive reasoning skills and, in particular, require explicit memory of past objects. This dataset is minimally-biased, highly configurable, and designed to produce a rich array of performance measures through a large number of named tasks.

We also built a recurrent neural network model that harnesses a number of attention and gating mechanisms to solve the COG dataset in a natural, human-interpretable way. The model also achieves near state-of-the-art performance on another visual reasoning dataset, CLEVR. The model uses a recurrent controller to pay attention to different parts of images and instructions, and to produce verbal outputs, all in an iterative fashion. These iterative attention signals provide multiple windows into the model's step-by-step pondering process and provide clues as to how the model breaks complex instructions down into smaller computations. Finally, the network is able to generalize immediately to completely untrained tasks, demonstrating zero-shot learning of new tasks.

References

1. Hassabis, D., Kumaran, D., Summerfield, C., Botvinick, M.: Neuroscience-inspired artificial intelligence. Neuron **95**(2), 245–258 (2017)
2. Hu, R., Andreas, J., Rohrbach, M., Darrell, T., Saenko, K.: Learning to reason: end-to-end module networks for visual question answering. CoRR, abs/1704.05526, vol. 3 (2017)
3. Johnson, J., et al.: Inferring and executing programs for visual reasoning. arXiv preprint arXiv:1705.03633 (2017)
4. Santoro, A., et al.: A simple neural network module for relational reasoning. In: Advances in Neural Information Processing Systems, pp. 4974–4983 (2017)
5. Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A.: Film: visual reasoning with a general conditioning layer. arXiv preprint arXiv:1709.07871 (2017)
6. Antol, S., et al.: VQA: visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2425–2433 (2015)
7. Gao, H., Mao, J., Zhou, J., Huang, Z., Wang, L., Xu, W.: Are you talking to a machine? Dataset and methods for multilingual image question. In: Advances in Neural Information Processing Systems, pp. 2296–2304 (2015)
8. Malinowski, M., Fritz, M.: A multi-world approach to question answering about real-world scenes based on uncertain input. In: Advances in Neural Information Processing Systems, pp. 1682–1690 (2014)
9. Zhu, Y., Groth, O., Bernstein, M., Fei-Fei, L.: Visual7W: grounded question answering in images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4995–5004 (2016)
10. Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C.L., Girshick, R.: CLEVR: a diagnostic dataset for compositional language and elementary visual reasoning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1988–1997. IEEE (2017)
11. Sturm, B.L.: A simple method to determine if a music information retrieval system is a horse. IEEE Trans. Multimed. **16**(6), 1636–1644 (2014)
12. Agrawal, A., Batra, D., Parikh, D.: Analyzing the behavior of visual question answering models. arXiv preprint arXiv:1606.07356 (2016)
13. Winograd, T.: Understanding Natural Language. Academic Press Inc., Orlando (1972)
14. Mnih, V., et al.: Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)
15. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529 (2015)

16. Vinyals, O., et al.: StarCraft II: a new challenge for reinforcement learning. arXiv preprint arXiv:1708.04782 (2017)
17. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR (2014)
18. Abu-El-Haija, S., et al.: YouTube-8M: a large-scale video classification benchmark. arXiv preprint arXiv:1609.08675 (2016)
19. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: ActivityNet: a large-scale video benchmark for human activity understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 961–970 (2015)
20. Hudson, D.A., Manning, C.D.: Compositional attention networks for machine reasoning. In: International Conference on Learning Representations (2018)
21. Diamond, A.: Executive functions. Ann. Rev. Psychol. **64**, 135–168 (2013)
22. Miyake, A., Friedman, N.P., Emerson, M.J., Witzki, A.H., Howerter, A., Wager, T.D.: The unity and diversity of executive functions and their contributions to complex frontal lobe tasks: a latent variable analysis. Cogn. Psychol. **41**(1), 49–100 (2000)
23. Berg, E.A.: A simple objective technique for measuring flexibility in thinking. J. Gen. Psychol. **39**(1), 15–22 (1948)
24. Milner, B.: Effects of different brain lesions on card sorting: the role of the frontal lobes. Arch. Neurol. **9**(1), 90–100 (1963)
25. Baddeley, A.: Working memory. Science **255**(5044), 556–559 (1992)
26. Miller, E.K., Erickson, C.A., Desimone, R.: Neural mechanisms of visual working memory in prefrontal cortex of the macaque. J. Neurosci. **16**(16), 5154–5167 (1996)
27. Miller, E.K., Cohen, J.D.: An integrative theory of prefrontal cortex function. Ann. Rev. Neurosci. **24**(1), 167–202 (2001)
28. Newsome, W.T., Britten, K.H., Movshon, J.A.: Neuronal correlates of a perceptual decision. Nature **341**(6237), 52 (1989)
29. Romo, R., Salinas, E.: Cognitive neuroscience: flutter discrimination: neural codes, perception, memory and decision making. Nat. Rev. Neurosci. **4**(3), 203 (2003)
30. Mante, V., Sussillo, D., Shenoy, K.V., Newsome, W.T.: Context-dependent computation by recurrent dynamics in prefrontal cortex. Nature **503**(7474), 78 (2013)
31. Rigotti, M., et al.: The importance of mixed selectivity in complex cognitive tasks. Nature **497**(7451), 585 (2013)
32. Yntema, D.B.: Keeping track of several things at once. Hum. Factors **5**(1), 7–17 (1963)
33. Zelazo, P.D., Frye, D., Rapus, T.: An age-related dissociation between knowing rules and using them. Cogn. Dev. **11**(1), 37–63 (1996)
34. Owen, A.M., McMillan, K.M., Laird, A.R., Bullmore, E.: N-back working memory paradigm: a meta-analysis of normative functional neuroimaging studies. Hum. Brain Mapp. **25**(1), 46–59 (2005)
35. Graves, A., Wayne, G., Danihelka, I.: Neural turing machines. CoRR abs/1410.5401 (2014)
36. Joulin, A., Mikolov, T.: Inferring algorithmic patterns with stack-augmented recurrent nets. CoRR abs/1503.01007 (2015)
37. Collins, J., Sohl-Dickstein, J., Sussillo, D.: Capacity and trainability in recurrent neural networks. Stat **1050**, 28 (2017)
38. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
39. Graves, A., et al.: Hybrid computing using a neural network with dynamic external memory. Nature **538**(7626), 471–476 (2016)

40. Kay, W., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
41. Ng, J.Y.H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: deep networks for video classification. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4694–4702. IEEE (2015)
42. Weston, J., et al.: Towards AI-complete question answering: a set of prerequisite toy tasks. arXiv preprint arXiv:1502.05698 (2015)
43. Zitnick, C.L., Parikh, D.: Bringing semantics into focus using visual abstraction. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3009–3016. IEEE (2013)
44. Kuhnle, A., Copestake, A.: ShapeWorld-a new test methodology for multimodal language understanding. arXiv preprint arXiv:1704.04517 (2017)
45. Xu, H., Saenko, K.: Ask, attend and answer: exploring question-guided spatial attention for visual question answering. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 451–466. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_28
46. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. In: International Conference on Learning Representations (ICLR) (2017)
47. Luck, S.J., Vogel, E.K.: The capacity of visual working memory for features and conjunctions. Nature **390**(6657), 279 (1997)
48. Cole, M.W., Laurent, P., Stocco, A.: Rapid instructed task learning: a new window into the human brains unique capacity for flexible cognitive control. Cogn. Affect. Behav. Neurosci. **13**(1), 1–22 (2013)
49. Graves, A.: Adaptive computation time for recurrent neural networks. arXiv preprint arXiv:1603.08983 (2016)
50. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456 (2015)
51. Xu, K., et al.: Show, attend and tell: neural image caption generation with visual attention. In: International Conference on Machine Learning, pp. 2048–2057 (2015)
52. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Sig. Process. **45**(11), 2673–2681 (1997)
53. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
54. Andersen, R.A., Snyder, L.H., Bradley, D.C., Xing, J.: Multimodal representation of space in the posterior parietal cortex and its use in planning movements. Ann. Rev. Neurosci. **20**(1), 303–330 (1997)
55. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.C.: Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Advances in Neural Information Processing Systems, pp. 802–810 (2015)
56. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
57. Yang, G.R., Song, H.F., Newsome, W.T., Wang, X.J.: Clustering and compositionality of task representations in a neural network trained to perform many cognitive tasks. bioRxiv, p. 183632 (2017)
58. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)