

# Interpretable Visual Reasoning via Induced Symbolic Space

Zhonghao Wang<sup>1</sup>, Mo Yu<sup>2</sup>, Kai Wang<sup>3</sup>, Jinjun Xiong<sup>2</sup>,  
Wen-mei Hwu<sup>1</sup>, Mark Hasegawa-Johnson<sup>1</sup>, Humphrey Shi<sup>3,1</sup>

<sup>1</sup>UIUC, <sup>2</sup>IBM Research, <sup>3</sup>University of Oregon

## Abstract

We study the problem of concept induction in visual reasoning, i.e., identifying concepts and their hierarchical relationships from question-answer pairs associated with images; and achieve an interpretable model via working on the induced symbolic concept space. To this end, we first design a new framework named object-centric compositional attention model (OCCAM) to perform the visual reasoning task with object-level visual features. Then, we come up with a method to induce concepts of objects and relations using clues from the attention patterns between objects' visual features and question words. Finally, we achieve a higher level of interpretability by imposing OCCAM on the objects represented in the induced symbolic concept space. Experiments on the CLEVR dataset demonstrate: 1) our OCCAM achieves a new state of the art without human-annotated functional programs; 2) our induced concepts are both accurate and sufficient as OCCAM achieves an on-par performance on objects represented either in visual features or in the induced symbolic concept space. Our code will be made available at <https://github.com/SHI-Labs/Interpretable-Visual-Reasoning>.

## 1. Introduction

Recent advances in Visual Question Answering (VQA) [1, 29, 16, 3, 15, 19, 14, 36, 25] usually rely on carefully designed neural attention models over images, and rely on pre-defined lists of concepts to enhance the compositional reasoning ability of the attention modules. Human prior knowledge plays an essential role in the success of the model design.

We focus on a less-studied problem in this field – given only question-answer pairs and images, whether it is possible to induce the visual concepts that are sufficient for completing the visual reasoning tasks. By sufficiency, we hope to maintain the predictive accuracy of the visual reasoning process, when using the induced concepts in place of the original visual features. We consider concepts that

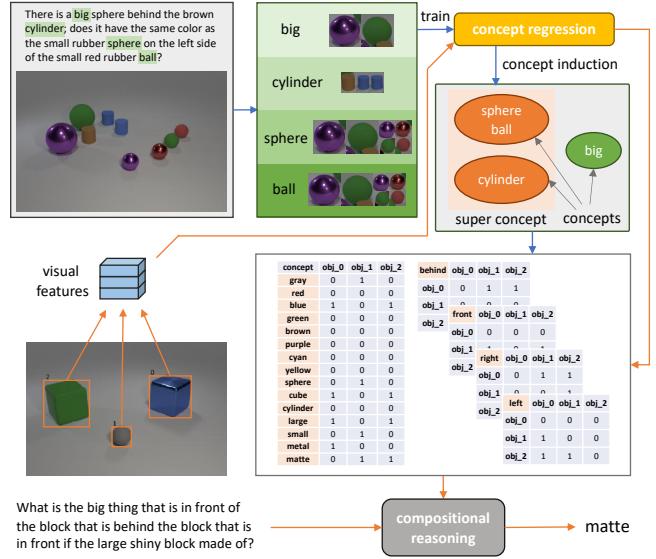


Figure 1. Illustration of our framework. Our model induces the concepts and super concepts with the attention correlation between the objects and question words in image-question pairs as the paths shown in blue arrows. Then, it answers a question about an image via compositional reasoning on the induced symbolic representations of objects and object relations, shown as the orange paths.

are important for visual reasoning, including properties of objects (e.g., *red, cube*) and relations between objects (e.g., *left, front*). The aforementioned scope and sufficiency criterion require accurately associating the induced symbols of concepts to both visual features and words that can express the meanings of concepts, so that each new instance of question-image pair can be transformed into the induced concept space for further computations. Additionally, it is necessary to identify super concepts, i.e., hypernyms of concept subsets (e.g., *shape*). The concepts inside a super concept are exclusive, so that the system knows each object can only possess one value in each subset. This introduces structural information to the concept space (multiple one-hot vectors for each visual object) and further guarantees the accuracy of the aforementioned transformation.

The value of the study has two folds. First, our pro-

posed problem aims to identify visual concepts, their argument patterns (properties or relations) and their hierarchy (super concepts) *without* using any concept-level supervision. Solving the problem frees both the efforts of human annotations and human designs of concept schema required in previous visual reasoning works. At the same time, the problem is technically more challenging compared to the related existing problem like unsupervised or weakly-supervised visual grounding [35]. Second, by constraining the visual reasoning models to work over the induced concepts, the ability of concept induction improves the interpretability of visual reasoning models. Unlike previous interpretable visual reasoning models that rely on human-written rules to associate neural modules with given concept definitions [14, 25, 31], our method resolves the concept definitions and associations interpretability automatically in the learning process, without the need of trading off for hand-crafted model designs.

We achieve the proposed challenge in three steps. First, we propose a new model architecture, object-centric compositional attention model (OCCAM), that performs object-level visual reasoning instead of pixel-level by extracting object-level visual features with ResNet [12] and pooling the features according to each object’s bounding box. The object-level reasoning not only improves over the state-of-the-art methods, but also provides a higher-level interpretability for concept association and identification. In our second step, we benefit from this model’s attention values over objects to create classifiers mapping visual objects to words; then derive the concepts and super concepts from the object-word cooccurrence matrices as shown in Figure 1. Finally, our concept-based visual reasoning framework predicts the concepts of objects and object relations and performs compositional reasoning using the predicted symbolic concept embeddings instead of the original visual features as shown in Figure 1.

Experiments on the CLEVR dataset demonstrate that our overall approach improves the level of interpretability of neural visual reasoning models, and maintains the predictive accuracy: (1) our object-level visual reasoning model improves over the previous state-of-the-art methods; (2) our induced concepts and concept hierarchy is accurate in human study; and (3) our induced concepts are sufficient for visual reasoning – replacing visual features with concepts leads to only 0.7% performance drop.

## 2. Related Works

**Visual Question Answering (VQA)** requires models to reason a question about an image to infer an answer. Recent works on this task can be partitioned into two groups: holistic models [34, 32, 1, 29, 16] and modular models [2, 3, 15, 19, 14, 36, 25], according to whether the approach has explicit sub-task structures. A typical holistic model,

MAC [16], perform iterative reasoning steps with an attention mechanism on the image. A modular framework, NS-CL [25], designs multiple principle functions over the extracted features to explain the reasoning process.

**Model interpretability.** Plenty of previous works attempt to explain the decision rules of the learned models. For instance, Bau et al. [5] proposed network dissection to quantify interpretability of CNNs. Zhang et al. [38] explained a CNN prediction at the semantic level with decision trees. Shi et al. [31] generated scene graphs from images to explicitly trace the reasoning-flow. Whereas some works [28, 14] focused on visual attentions to provide enhanced interpretability. Our work is closely related to *the self-explaining systems via rationalization* [23, 6, 37]. Different from previous works that extract subsets of inputs as explanations, our work moves one-step further by learning parts of the structural explanation definitions (i.e., the concepts hierarchy) together with explanations.

**Visual concept learning** contributes to many visual-linguistic applications, such as cross-modal retrieval [22], visual captioning [20], and visual-question answering [24, 4]. Some recent papers [36, 25] attempt to disentangle visual concept learning and reasoning. Based on the visual concepts learned from VQA, Han et al. [10] learned metaconcepts, i.e., relational concepts about concepts, with augmented questions and answers. In contrast, our model learns super concepts without external knowledge.

## 3. Object-Centric Compositional Reasoning

In this section, we introduce a new neural architecture for neural reasoning. Our model adopts an object-centric approach, which performs compositional reasoning over the extracted object-level visual features. This proposed object-centric reasoning approach not only achieves state-of-the-art performance, but also plays a key role in the task of inducing object-wise or relational concepts as will be described in section 4.

Figure 2 shows our general framework. It includes two training phases involving the process from feeding the input images and questions to attain the final answers. Phase 1 (black-colored paths) corresponds to the training of our object-centric neural model, in which we train the object-level feature extractor, the compositional reasoning module and the question embedding LSTM; Phase 2 (red-colored paths) corresponds to the induction of symbolic concepts based on the aforementioned trained neural modules, as well as the training of a concept projection module so that the induced concepts can be accommodated in our reasoning pipeline. The figure shows the central role that the object-centric model plays in our framework.

In the following sections, we will first review the background of the compositional reasoning framework which was first proposed by Hudson and Manning [16] (section

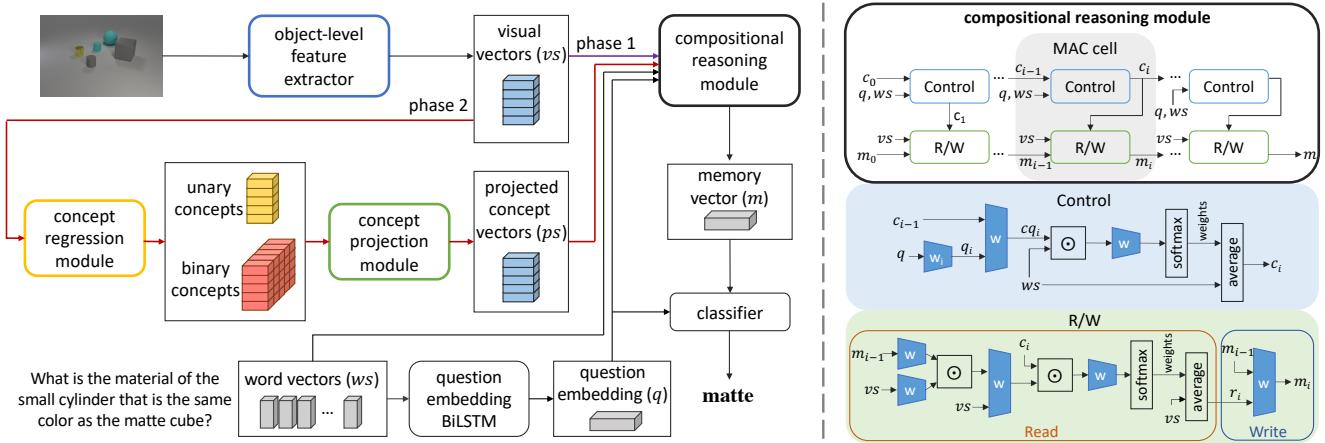


Figure 2. The framework and the compositional reasoning module. The left graph shows the general framework; The phase 1 training path is drawn in purple and the phase 2 training paths are drawn in red. The black paths are shared for both training phases. The structures of our proposed object-level feature extractor, concept regression module and concept projection module are shown in Figures 3, 4 and 7.

3.1); then, we will introduce the proposed object-level compositional attention network (section 3.2).

### 3.1. Background on compositional reasoning

**Notations** As shown in Figure 2, we name the visual vectors as  $vs$ , the output memory vector from the compositional reasoning module as  $m$ , the embedded word vectors for questions as  $ws$ , and the question embedding as  $q$ .

**Baseline visual reasoning framework** The original compositional reasoning framework [16] is similar to the phase 1 of our framework in Figure 2, except that it works on pixel-level instead of object-level features. To generate  $vs$ , it feeds the image to a ResNet101 [12] pretrained on ImageNet [8] and flatten the last feature maps across the width and height as  $vs$ . For the question inputs, we first convert each question word to its word embedding vector ( $ws$ ), then input  $ws$  to a bidirectional LSTM [13, 9] to extract the question embedding vector  $q$ . The compositional reasoning module takes  $vs$ ,  $ws$  and  $q$  as inputs and performs multi-step reasoning to attain  $m$ , the final step memory output. Finally, the classifier outputs the probability for each answer choice with a linear classifier over the concatenation of  $m$  and  $q$ . The loss function can thus be written as:

$$\mathcal{L}(ws, vs, q) = - \sum_{k \in K} y_k \log \mathcal{F}(q_k, \mathcal{G}(ws_k, vs_k, q_k)) \quad (1)$$

$$q_k = \mathcal{Q}(ws_k), vs_k = \mathcal{I}(im_k).$$

$K$  is the total number of image-question pairs,  $y$  is the one-hot ground truth vector,  $\mathcal{F}$  is the classifier,  $\mathcal{G}$  is the compositional module,  $\mathcal{Q}$  is the question embedding LSTM,  $\mathcal{I}$  is the visual feature extractor and  $im$  is the image input.

**The MAC reasoning module** The aforementioned framework employs the MAC (i.e., *Memory*, *Attention*, and *Composition*) cells [16] as the compositional reasoning

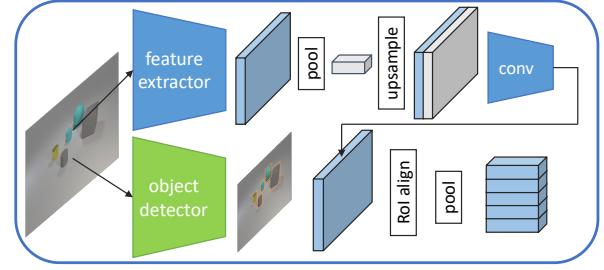


Figure 3. The structure of **object-level feature extractor**.

module. This module processes visual and language inputs in a sequential way. Shown in Figure 2 (right), each MAC cell contains a control unit and an R/W (Read/Write) unit; the blue diagrams labeled with  $w$  stand for fully connected layers and the symbol  $\odot$  stands for Hadamard product.

At each step, the  $i$ -th MAC cell receives the control signal  $c_{i-1}$  and the memory output from the previous step,  $m_{i-1}$ , and outputs the new memory vector  $m_i$ . The control unit computes the single  $c_i$  to control reading of  $vs$  in the R/W unit. Specifically, it computes the interactions among  $c_{i-1}$ ,  $q_i$ , and each vector in  $ws$  to produce the attention weights, and weighted averages  $ws$  to produce  $c_i$ . The control unit of each MAC cell has a unique question embedding projection layer, while all other layers are shared. The R/W unit aims to read the useful  $vs$  and store the read information into  $m_i$ . It first computes the interactions among  $m_{i-1}$ ,  $c_{i-1}$  and each vector in  $vs$  to attain the attention weights, weighted averages  $vs$  to produce a read vector  $r_i$ , and finally computes the interaction of  $r_i$  and  $m_{i-1}$  to produce  $m_i$ . The weights of the R/W units are shared across all MAC cells. The initial control signal and memory  $c_0$  and  $m_0$  are learnable parameters.

### 3.2. Object-level compositional attention network

The object-level compositional attention network is shown in Figure 2 with phase 1 path, and optimizes Eqn

(1) with  $vs$  generated by our object-level feature extractor.

**Object-level feature extraction** Fed with an image, the object-level feature extractor produces a set of vectors, each of which encodes both a single object’s unary visual features and its interactions with other objects. The structure of this module is shown in Figure 3. Following [25], we use Mask-RCNN [11] to detect all objects in an image and output the bounding boxes for them. The image is fed to a ResNet34 network [12] pretrained on ImageNet [8] to generate the feature maps in parallel.

On top of the ResNet34 feature maps, we apply a global average pooling to get a single **global feature vector** (the gray vector in the figure). We concatenate this global vector with the feature map at each location, followed by **three convolution layers**. This global vector is crucial since it allows the visual features to encode the interaction among objects; and the three convolution layers fuse the local and global features into a single visual vector at each position.

Finally, to get object-level features from the above pixel-level fused features, we utilize ROI align [11] to project the objects’ bounding boxes onto the fused feature vectors to generate the ROI feature maps, and average pool the ROI feature maps for each object to produce the object-level  $vs$ .

## 4. Concept Induction and Reasoning

In this section, we describe how we achieve our goal of inducing symbolic concepts for objects and performing compositional reasoning on the induced concepts. We first formalize the problem of concept induction (section 4.1). Then building on the learned object-centric reasoning model introduced in the previous section, we propose to induce concepts of both unary object properties or the binary relations between objects (section 4.2). Finally we introduce how to achieve compositional reasoning over symbolic concepts by substituting the object-level features with the induced concepts (section 4.3).

### 4.1. Problem definition

We consider identifying three types of concepts: (1) the **unary concepts**  $\mathcal{C}^u$  that are properties of objects (e.g., *red*, *cube*, etc.); (2) the **binary concepts**  $\mathcal{C}^b$  that are relation descriptions between any two objects (e.g., *left*, *front* etc.); and (3) the **super concepts**  $\mathcal{C}^{sup}$  that are hypernyms of certain subsets of concepts (e.g., *color*, *shape*, etc.), subject to that each object can only possess one concept under each super concept, e.g., *cube* and *sphere*.

As questions refer to objects and describe object relations in images and, more importantly, include all the semantic information to reach an answer, it is natural to induce the concepts from question words. Therefore we assume that all the unary and binary concepts have their corresponding words; and these words are a subset of the nouns

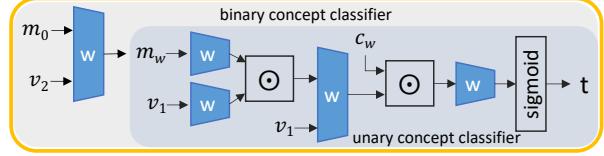


Figure 4. The structure of the **concept regression module**.  $v_1$  and  $v_2$  are the object-level visual vectors representing two objects respectively, and  $c_w$  is the word vector.  $m_0$  is a fixed vector and  $m_w$  equals to  $m_0$  for the unary concept classifier.

or adjectives from all the training questions. We denote the sets of words that describe unary concepts and binary concepts as  $M^u$  and  $M^b$  respectively. Therefore, the goal of concept induction consists of the following tasks:

- **Visual mapping:** for each concept  $c \in \mathcal{C}^u$  or  $\mathcal{C}^b$ , learning a mapping from the visual feature  $v$  to  $c$ . In other words, a prediction function  $f_c(v) \in \{0, 1\}$  is learned to predict the existence of concept  $c$  from the visual feature  $v$  of an object.
- **Word mapping:** for each concept  $c \in \mathcal{C}^u$  or  $\mathcal{C}^b$ , identifying a subset of words  $S_c \subset M^u$  or  $M^b$  that are synonyms representing the same concept, e.g., the concept of ‘*cube*’ corresponds to set of words {*cube*, *cubes*, *block*, *blocks*, ...}.
- **Super concept induction:** clustering of concepts to form super concepts. Each super concept  $c$  contains a set of concepts  $\{c_1, \dots, c_k\} \subset \mathcal{C}^u$  or  $\mathcal{C}^b$ .

### 4.2. Concept induction

This section describes how we achieve the aforementioned tasks of concept induction. The key idea of our approach includes: (1) benefiting from the R/W unit from the trained MAC cells to achieve the visual mapping to textual words; (2) utilizing the inclusiveness of words’ visual mapping to induce each concept’s multiple word descriptions; (3) clustering super concepts from the mutual exclusiveness between concepts. To achieve the above, we first train two binary classifiers that can determine if a word correctly describes an object’s unique feature and if a word correctly describes a relation between two objects respectively. Then, with the help of these classifiers, we produce zero-one vectors for words that properly describe the unique features for each object and the relations between any pair of objects in single images across the dataset. Finally, we perform a clustering method on the word vectors to generalize unary and binary concepts, and the super concept sets.

**Visual mapping via regression from MAC cells** The concept regression module is shown in Figure 4. It is composed of a classifier for the unary concept word regression,  $\mathcal{B}^u(v_1, c_w) \in [0, 1]$ , and a classifier for the binary concept word regression,  $\mathcal{B}^b(v_1, v_2, c_w) \in [0, 1]$ .  $\mathcal{B}^u$  is expected to produce 1 if  $v_1$  can be described by the word vector  $c_w$ . Likewise,  $\mathcal{B}^b$  is expected to produce 1 if the relation of  $v_1$  to  $v_2$  can be described by the word vector  $c_w$ .

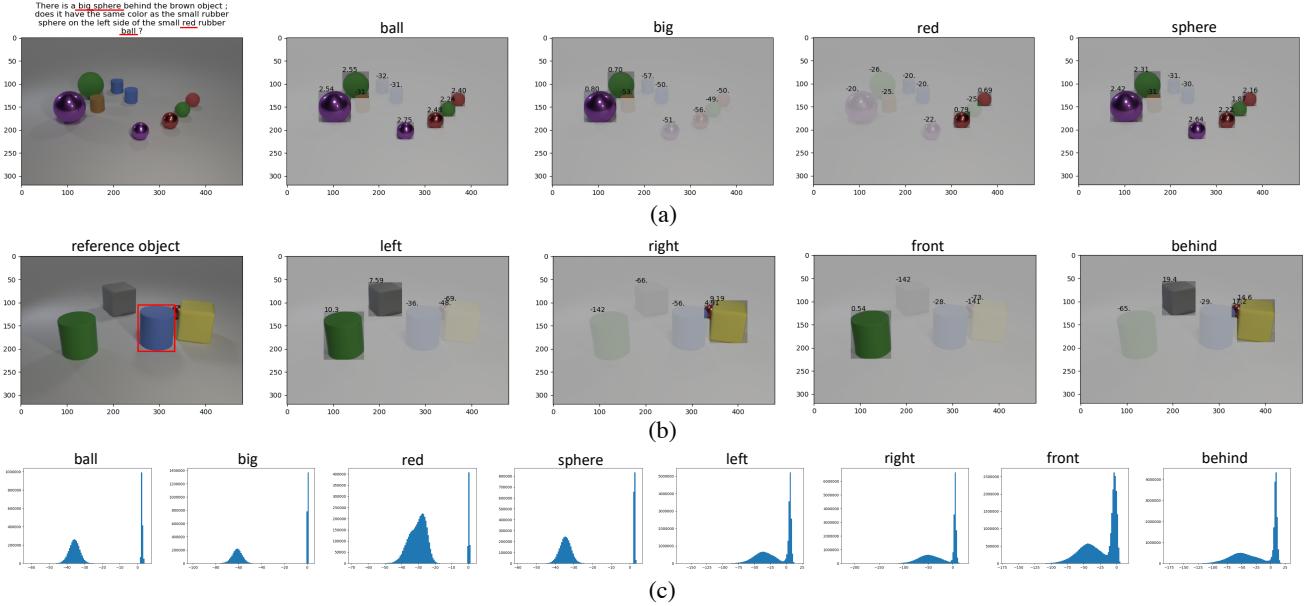


Figure 5. Attention visualization and attention logit distributions. (a) The attention visualization corresponding to the words describing the unary concepts by performing  $\mathcal{R}(vs, c_w, m_0)$ . Each of the words above the latter 4 images corresponds to a unique  $c_w$  and the value on each object is the attention logit (the same applies to (b)). (b) The attention visualization corresponding to the words describing the binary concepts by performing  $\mathcal{R}(vs, c_w, \mathcal{W}(m_0, v_2))$ .  $v_2$  represents the object bounded by a red rectangle in the first image. (c) the attention logit distribution corresponding to each word describing a concept.

We generate training data points  $P^u = \{(v_1^u, c_{w_i}^u, y_i^u)\}$  and  $P^b = \{(v_1^b, v_2^b, c_{w_i}^b, y_i^b)\}$  for  $\mathcal{B}^u$  and  $\mathcal{B}^b$  by utilizing the Read/Write unit (Figure 2(right)) in the reasoning module after phase 1 training. The whole generation process is described in Algorithm 1. We denote  $\mathcal{R}(vs, c_i, m_{i-1}) \in \mathbb{R}^{|\mathcal{O}|}$  for the sequence of functions before the softmax operation in the Read unit and  $\mathcal{W}(m_{i-1}, r_i) \in \mathbb{R}^D$  for the function of the Write unit, where  $O$  is the set of objects in an image and  $D$  is the vector dimension.

Specifically, our algorithm first uses  $\mathcal{R}(\cdot, \cdot, \cdot)$  and  $\mathcal{W}(\cdot, \cdot)$  to find the attention logits on the objects corresponding to words describing the unary and binary concepts in a question as shown in Figure 5(a&b). We then use the values of logits to determine if the object possesses the concept of the word (positive) or not (negative). Noticing the attention logit distribution of the sampled objects for each word is a two-peak distribution (Figure 5(c)), we use a GMM [33] with two Gaussian components to model the distribution and find the decision boundary for each word's attention logit distribution. It is worth mentioning that the attention logit distribution for a unary concept word has two waves that do not interfere each other; while for a binary concept word the two waves interfere. This is because in some cases it is hard to tell if two objects have the relation described by a word. For example, it is hard to describe the “*in front of*” relation between two objects on the same horizontal axis (e.g., the blue and yellow objects in Figure 5(b)). Finally,  $P^u$  and  $P^b$  are generated by classifying the data points to

---

**Algorithm 1:** Classifier data points generation. ST( $\cdot$ ) splits a vector  $\alpha \in \mathbb{R}^\beta$  to a set of  $\beta$  values. GMM( $\cdot$ ) uses Gaussian Mixture Model to cluster a set of data points. FB( $\cdot$ ) finds the decision boundary for the 2 Gaussian components.  $\mathbb{1}$  is the indicator function.

---

```

Result:  $P^u, P^b$ 
 $P^u = \{\}, P^b = \{\}$ 
for  $x \in M^u \cup M^b$  do
|  $S_x = \{\}, bd_x = 0$ 
for  $vs, ws \in \text{DATASET}$  do
| for  $c_w \in ws \cap M^u$  do
| |  $S_{c_w} = S_{c_w} \cup \text{ST}(\mathcal{R}(vs, c_w, m_0))$ 
| for  $c_w \in ws \cap M^b$  do
| | for  $v \in vs$  do
| | |  $S_{c_w} = S_{c_w} \cup \text{ST}(\mathcal{R}(vs, c_w, \mathcal{W}(m_0, v)))$ 
| for  $x \in M^u \cup M^b$  do
| |  $bd_x = \text{FB}(\text{GMM}(S_x))$ 
for  $vs, ws \in \text{DATASET}$  do
| | for  $v_1 \in vs$  do
| | | for  $c_w \in ws \cap M^u$  do
| | | |  $y = \mathbb{1}(\mathcal{R}(v_1, c_w, m_0) > bd_{c_w})$ 
| | | |  $P^u = P^u \cup \{(v_1, c_w, y)\}$ 
| | | for  $c_w \in ws \cap M^b$  do
| | | | for  $v_2 \in \{vs - v_1\}$  do
| | | | |  $y = \mathbb{1}(\mathcal{R}(v_1, c_w, \mathcal{W}(m_0, v_2)) > bd_{c_w})$ 
| | | | |  $P^b = P^b \cup \{(v_1, v_2, c_w, y)\}$ 
```

---

positives and negatives with the decision boundaries.

With the data points  $P^u$  and  $P^b$  generated, we can train

	small	tiny	big	cube	ball	...
1	1	0	1	0	...	
0	0	1	1	0	...	
1	1	0	0	1	...	
0	0	1	0	1	...	
⋮	⋮	⋮	⋮	⋮	⋮	⋮

	left	right	front	behind
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
⋮	⋮	⋮	⋮	⋮

Figure 6. The zero-one matrices indicating word descriptions of objects and object relations across the dataset. (a) The matrix  $\Gamma^u$  indicates what words can describe objects. (b) The matrix  $\Gamma^b$  indicates what words can describe the relations object  $v_1$ 's (bounded by green rectangles) are to object  $v_2$ 's (bounded by red rectangles).

$\mathcal{B}^u$  and  $\mathcal{B}^b$  by minimizing the binary cross entropy loss.

**Binary coding of objects** With trained  $\mathcal{B}^u$  and  $\mathcal{B}^b$ , we represent an object  $o_1$  with a binary code vector. Each dimension corresponds a word. A dimension has value 1 if the corresponding word can describe  $o_1$  and 0 otherwise. The binary vectors of object properties and of the relations between two objects,  $o_1$  and  $o_2$  can be computed with the functions  $\gamma^u \in \mathbb{R}^{|M^u|}$  and  $\gamma^b \in \mathbb{R}^{|M^b|}$  respectively:

$$\begin{aligned}\gamma^u &= \mathbb{1}_{i>0.5}(\mathcal{B}^u(v_1, C^u)) \\ \gamma^b &= \mathbb{1}_{i>0.5}(\mathcal{B}^b(v_1, v_2, C^b)),\end{aligned}\quad (2)$$

where  $v_1$  and  $v_2$  are the object-level visual vectors of  $o_1$  and  $o_2$ ,  $C^u \in \mathbb{R}^{|M^u| \times D}$  and  $C^b \in \mathbb{R}^{|M^b| \times D}$  are the stacks of word embeddings in vocabulary  $M^u$  and  $M^b$ .  $\mathbb{1}_\alpha(\beta)$  performs elementwise on  $\beta$ : return 1 if the element satisfies condition  $\alpha$  or 0 otherwise.

By applying  $\gamma^u$  and  $\gamma^b$  to all the objects and relations in the dataset, we can attain a matrix  $\Gamma^u \in \{0, 1\}^{M^u, N^u}$  and a matrix  $\Gamma^b \in \{0, 1\}^{M^b, N^b}$  as shown in Figure 6, where  $N^u$  and  $N^b$  are the total numbers of objects and co-occurred object pairs. The two matrices summarize each word's corresponding visual objects in the whole dataset.

**Concept/super-concept induction** Finally, we group synonym words to unary and binary concepts and generate the super concepts. These two tasks are achieved via exploring the word inclusiveness and the concept exclusiveness captured by  $\Gamma^u$  and  $\Gamma^b$ : (1) words describing the same concept correspond to similar column vectors, e.g.,  $\Gamma_{small}^u$  and  $\Gamma_{tiny}^u$ ; (2) words describing exclusive concepts have column vectors that usually do not have 1 values on same objects simultaneously, e.g.,  $\Gamma_{cube}^u$  and  $\Gamma_{ball}^u$ . Based on the aforementioned ideas, we define the correlation metric between two words  $c_{w_1}$  and  $c_{w_2}$  as below:

$$\begin{aligned}\theta_{c_{w_1}, c_{w_2}} &= P(\gamma_{c_{w_1}} = 1 \mid \gamma_{c_{w_2}} = 1) + \\ &\quad P(\gamma_{c_{w_2}} = 1 \mid \gamma_{c_{w_1}} = 1) \\ &= \frac{|\Gamma_{c_{w_1}} \odot \Gamma_{c_{w_2}}|_1^1}{|\Gamma_{c_{w_2}}|_1^1} + \frac{|\Gamma_{c_{w_1}} \odot \Gamma_{c_{w_2}}|_1^1}{|\Gamma_{c_{w_1}}|_1^1}.\end{aligned}\quad (3)$$

---

**Algorithm 2:** Concept vector generalization.  $\text{MAX}(\alpha)$  returns the greatest value in the vector  $\alpha$  and  $\text{HARDMAX}(\alpha)$  returns a zero-one vector indicating the position of the greatest value in the vector  $\alpha$ .

---

**Result:**  $K^u, K^b$   
 $K^u = \mathbf{0}^{|O| \times |E^u|}, K^b = \mathbf{0}^{|O| \times |O| \times |E^b|}$   
**for**  $i \in O$  **do**  
  **for**  $e^u \in E^u$  **do**  
     $|K^u[i][e^u] = \text{MAX}(\mathcal{B}^u(v_i, \rho_{e^u}))$   
  **for**  $l^u \in L^u$  **do**  
     $|K^u[i][l^u] = \text{HARDMAX}(K^u[i][l^u])$   
  **for**  $j \in O - \{i\}$  **do**  
    **for**  $e^b \in E^b$  **do**  
       $|K^b[i][j][e^b] = \text{MAX}(\mathcal{B}^b(v_i, v_j, \rho_{e^b}))$   
    **for**  $l^b \in L^b$  **do**  
       $|K^b[i][j][l^b] = \text{HARDMAX}(K^u[i][j][l^b])$

---

This guarantees that  $\theta \rightarrow 0^+$  for two synonym words,  $\theta \rightarrow 2^-$  for two words corresponding to exclusive concepts and  $\theta \in (0, 2)$  for words corresponding to different nonexclusive concepts. We can produce the correlation sets for the words describing the unary concepts and the binary concepts respectively with Eqn (4).

$$\begin{aligned}\Theta^u &= \{\theta_{c_{w_1}, c_{w_2}}\}; c_{w_1}, c_{w_2} \in M^u \\ \Theta^b &= \{\theta_{c_{w_1}, c_{w_2}}\}; c_{w_1}, c_{w_2} \in M^b.\end{aligned}\quad (4)$$

Our final step fits two GMM on  $\Theta^u$  and  $\Theta^b$  respectively. Each GMM has three components  $\mathcal{N}_0$ ,  $\mathcal{N}_1$  and  $\mathcal{N}_2$ , with their mean values initialized with 0, 1 and 2. We then induce the unary and binary concepts, where each concept consists of synonym words whose mutual correlation is clustered to the Gaussian component  $\mathcal{N}_0$ . Similarly, we induce the super concepts, where each super concept contains multiple concepts and any two words from different concepts have correlation clustered to the Gaussian component of  $\mathcal{N}_2$ .

We denote the set of words corresponding to a concept  $e$  as  $\rho_e$ , the set of the super concept sets as  $L$ , the set of all concepts as  $E$ . Then, we can represent all the objects in an image with a unary concept matrix  $K^u$  and represent all the relations between any two objects in an image with a binary concept matrix  $K^b$  with Algorithm 2.

### 4.3. Concept compositional reasoning

Our ultimate goal is to perform compositional reasoning to answer a question with the generated concept representations  $K^u$  and  $K^b$  for an image; so as to confirm that our induced concepts are accurate and sufficient. We achieve this with the phase 2 training process in Figure 2. The key idea is to transplant the learned compositional reasoning module from manipulating the visual features to manipulating  $K^u$  and  $K^b$ , for attaining the answer to a question.

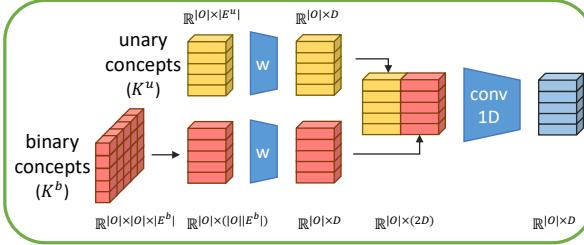


Figure 7. The structure of the **concept projection module**. We label the dimensions of matrices near them in the graph.

To this end, first, we project  $K^u$  and  $K^b$  to the same vector space with  $vs$  with the concept projection module shown in Figure 7, so that the compositional module can perform the reasoning steps on the projected concept vectors. Specifically, we first reduce the dimension of  $K^b$  from  $\mathbb{R}^{|\mathcal{O}| \times |\mathcal{O}| \times |E^b|}$  to  $\mathbb{R}^{|\mathcal{O}| \times |\mathcal{O}| \times |E^b|}$ , resulted in  $\hat{K}^b$ , because  $K^b$  can be understood as the relations to other objects for each object in an image. Then, we use two separate fully connected networks to project  $K^u$  and  $\hat{K}^b$  respectively, concatenate and use a sequence of 1D convolution layers to project the results to the same dimension of  $vs$ ’s.

Second, to minimize the discrepancy between the distribution of our projected vectors and that of the original visual vectors  $vs$ , we fix the weights of other modules in the framework and only train the concept project module by optimizing the target function Eqn. (1). Then, we train the concept projection module and the compositional reasoning module with other modules’ weights fixed to better optimize Eqn. (1). The result is a compositional reasoning model that works on the induced concepts only.

## 5. Experiment

### 5.1. Settings

**Dataset** We use the CLEVR [18] dataset to evaluate our model. The dataset comprises images of synthetic objects of various shapes, colors, sizes and materials and question/answer pairs about these images. The questions require multi-hop reasoning, such as finding the transitive relations, counting numbers, comparing properties, to attain correct answers. Each question is provided a ground truth human-written programs. Because the programs rely on pre-defined concepts thus do not fit our problem, we let our framework learn the compositional reasoning by itself *without* using the program annotations. There are 70k images and  $\sim 700$ k questions in the training set; 15k images and  $\sim 150$ k questions in the validation set. We follow the previous works [36, 16, 25] to train our model on the whole training set and test our model on the validation set.

**Training details** We set the hidden dimension  $D$  to in all modules. We follow [16] to design the question embed-

Table 1. The comparison of our object-level compositional reasoning framework to the state-of-the-art methods. \* indicates the method uses external program annotations.

method	overall	count		exist	comp	query	comp
		numb	attr	attr	attr	attr	attr
Human [19]	92.6	86.7	96.6	86.5	95.0	96.0	
NMN* [3]	72.1	52.5	72.7	79.3	79.0	78.0	
N2NMN* [15]	83.7	68.5	85.7	84.9	90.0	88.7	
IEP* [19]	96.9	92.7	97.1	98.7	98.1	98.9	
TbD* [26]	99.1	97.6	99.4	99.2	99.5	99.6	
NS-VQA* [36]	<b>99.8</b>	<b>99.7</b>	<b>99.9</b>	<b>99.9</b>	<b>99.8</b>	<b>99.8</b>	
RN [30]	95.5	90.1	93.6	97.8	97.1	97.9	
FiLM [29]	97.6	94.5	93.8	99.2	99.2	99.0	
MAC [16]	98.9	97.2	99.4	<b>99.5</b>	99.3	99.5	
NS-CL [25]	98.9	<b>98.2</b>	99.0	98.8	99.3	99.1	
OCCAM (ours)	<b>99.4</b>	98.1	<b>99.8</b>	99.0	<b>99.9</b>	<b>99.9</b>	

Table 2. The ablation study on the choice of reasoning steps for the object-level compositional reasoning.

steps	4	8	12	16
accuracy	94.3	98.6	<b>99.4</b>	99.1

ding module, the compositional module and the classifier. For the object-level feature extractor, we make the backbone ResNet34 learnable and zero-pad the output  $vs$  to 12 vectors in total for any image. Notice that the maximum number of objects in an image is 11, so that the reasoning module is able to read nothing into the memory for some steps. For the concept projection module, to cover the full view of  $vs$ , the conv1D consists of five 1D convolution layers with kernel sizes (7,5,5,5,5), each followed by a Batch Norm layer [17] and an ELU activation layer [7].

We use Adam optimizer [21] with momentum 0.9 and 0.999. Phase 1 and phase 2 share a same training schedule: the learning rate is initiated with  $10^{-4}$  for the first 20 epochs and is halved every 5 epochs afterwards until stopped at the 40th epoch. We train the concept regression module separately with learning rate of  $10^{-4}$  for 6 epochs. Phase 1 uses 4 Nvidia V100 GPUs with batch size 256. The other training processes use a single GPU with a batch size of 192.

### 5.2. Object-level reasoning

We first perform the end-to-end phase 1 training shown in Figure 2. The performance comparison of our model to the state-of-the-art models is shown in Table 1. Under the condition that no external human-labeled programs are used, our model achieves 99.4% in accuracy, a new state-of-the-art performance on CLEVR validation set compared to other methods under the same condition. Our model also has an on-par performance with the best model [36] using external human-labeled programs. Compared to the original MAC [16] framework which uses image-level attentions

Table 3. The comparison of our object-level compositional reasoning framework and our concept compositional reasoning framework. The number of reasoning steps is set to 8.

method	overall	count	exist	comp	query	comp
				numb	attr	attr
object	98.6	95.9	99.8	96.2	99.8	99.7
concept	97.9	95.6	98.7	97.3	98.4	99.3

and achieves 98.9% in accuracy on CLEVR validation set, our model proves that the constraint of attentions on the objects are useful for improving the performance. We do not use the position embedding to explicitly encode the positions of objects for relational reasoning; however, we use the global features to enhance the model’s understanding of inter-object relations. This implicates that the relations among objects are totally learnable concepts without external knowledge for the deep network.

Table 2 further gives an ablation study on the numbers of reasoning steps, i.e., the number of MAC modules, for our model. The reasoning model with 4 steps has a performance gap to the models with 8, 12 or 16 steps, while the latter three models have on-par performances. We conjecture that the model with low reasoning steps may not be able to capture multiple hops of a question and the model performance converges with an increasing number of reasoning steps.

### 5.3. Concept induction and reasoning

**Results of concept induction** To achieve the balance of the performance and the interpretability, we choose the object-level compositional reasoning model with 8 reasoning steps for the concept induction and reasoning. After visual mapping, binary coding and concept/super-concept induction described in section 4.2, the unary concepts and super concepts are induced as shown in Figure 8; the binary concepts are ‘left’, ‘right’, ‘front’ and ‘behind’, and {‘left’, ‘right’} and {‘front’, ‘behind’} form two super concept sets. Appendix A provides more detailed information on how these clusters are generated by presenting the concept correlations. The generated concept hierarchy perfectly recovers the definition in CLEVR data generator and matches human prior knowledge, showing the success of our approach.

**Concept-level reasoning** For each image in the dataset, we can now represent each object and the relations between any two objects with the induced unary and binary concepts. With the method described in section 4.3, we project the concepts back to the visual feature space, so that the compositional reasoning module can perform the reasoning steps on the projected vectors( $ps$ ). Our concept compositional reasoning model achieves an on-par performance with the one of the object-level compositional reasoning model as shown in Table 3. We also visualize the reasoning steps for the concept reasoning module as shown in Appendix B.

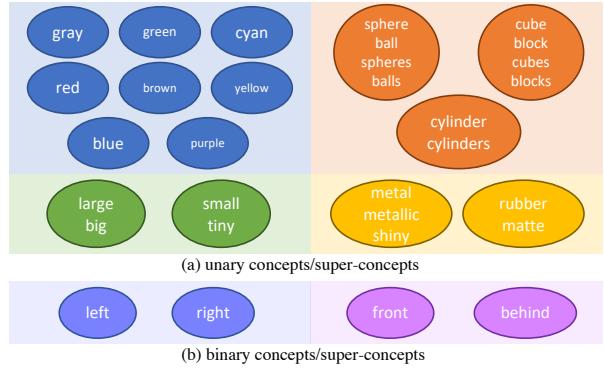


Figure 8. Concepts and super concept sets. Each circle represents a concept described by the words in that circle. A super concept set comprises the concepts represented by circles of the same color.

$$v_0 - v_1 + v_2 \approx v_3$$

$$k_0 = \{brown, sphere, large, matte\} \setminus k_1 = \{brown, sphere, small, metal\} \oplus k_2 = \{purple, cylinder, small, metal\} = k_3 = \{purple, cylinder, large, matte\}$$

Figure 9. An multi-modal analogy example enabled by our results.

**Discussion** Our concept induction results bridge the visual and symbolic spaces. The results enable to extend word analogy [27] (e.g., “Madrid” - “Spain” + “France” → “Paris”) into the multi-modality setting. Figure 9 gives an example, starting with the initial object  $v_0$  and its predicted concepts  $K_0$ , subtracting concepts  $K_1$  and adding new concepts  $K_2$  result in a new concept set  $K_3$  (Figure 9 (bottom)). Then if we retrieve visual object  $v_i$  with each concept set  $K_i$  along the path (Figure 9 (top)), we have  $v_0 - v_1 + v_2 \approx v_3$  in the original visual feature space. Similarly, with the exclusiveness inside a super concept, we can quantify the distance between two visual objects with the super concept space. More details are provided in Appendix B.

For future works, our method can be extended to more sophisticated induction tasks, such as inducing concepts from phrases, with more complicated hierarchy, with degrees of features (e.g., *dark blue*, *light blue*) and inducing complicated relations between objects (e.g. *a little bigger*).

## 6. Conclusions

Our proposed OCCAM framework performs pure object-level reasoning and achieves a new state-of-the-art without human-annotated functional programs on the CLEVR dataset. Our framework makes the object-word cooccurrence information available, which enables induction of the concepts and super concepts based on the inclusiveness and the mutual exclusiveness of words’ visual mappings. When working on concepts instead of visual features, OCCAM achieves comparable performance, proving the accuracy and sufficiency of the induced concepts.

## References

- [1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6077–6086, 2018. [1](#), [2](#)
- [2] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NACACL)*, 2016. [2](#)
- [3] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 39–48, 2016. [1](#), [2](#), [7](#)
- [4] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2425–2433, 2015. [2](#)
- [5] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6541–6549, 2017. [2](#)
- [6] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning (ICML)*, pages 883–892, 2018. [2](#)
- [7] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *International Conference on Learning Representations (ICLR)*, 2016. [7](#)
- [8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. [3](#), [4](#)
- [9] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610, 2005. [3](#)
- [10] Chi Han, Jiayuan Mao, Chuang Gan, Josh Tenenbaum, and Jiajun Wu. Visual concept-metaconcept learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5001–5012, 2019. [2](#)
- [11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017. [4](#)
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. [2](#), [3](#), [4](#)
- [13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. [3](#)
- [14] Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. Explainable neural computation via stack neural module networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 53–69, 2018. [1](#), [2](#)
- [15] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 804–813, 2017. [1](#), [2](#), [7](#)
- [16] Drew A Hudson and Christopher D Manning. Compositional attention networks for machine reasoning. In *International Conference on Learning Representations (ICLR)*, 2018. [1](#), [2](#), [3](#), [7](#)
- [17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015. [7](#)
- [18] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2901–2910, 2017. [7](#)
- [19] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2989–2998, 2017. [1](#), [2](#), [7](#)
- [20] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137, 2015. [2](#)
- [21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. [7](#)
- [22] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models, 2014. [2](#)
- [23] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 107–117, 2016. [2](#)
- [24] Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1–9, 2015. [2](#)
- [25] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *International Conference on Learning Representations (ICLR)*, 2018. [1](#), [2](#), [4](#), [7](#)
- [26] David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning.

- In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 7
- [27] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3111–3119, 2013. 8
- [28] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Attentive explanations: Justifying decisions and pointing to the evidence. *arXiv preprint arXiv:1612.04757*, 2016. 2
- [29] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C Courville. Film: Visual reasoning with a general conditioning layer. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018. 1, 2, 7
- [30] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4967–4976, 2017. 7
- [31] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8376–8384, 2019. 2
- [32] Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision (ECCV)*, pages 451–466. Springer, 2016. 2
- [33] Guorong Xuan, Wei Zhang, and Peiqi Chai. Em algorithms of gaussian mixture model and hidden markov model. In *International Conference on Image Processing (ICIP)*, volume 1, pages 145–148. IEEE, 2001. 5
- [34] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21–29, 2016. 2
- [35] Raymond A Yeh, Minh N Do, and Alexander G Schwing. Unsupervised textual grounding: Linking words to image concepts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6125–6134, 2018. 2
- [36] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1031–1042, 2018. 1, 2, 7
- [37] Mo Yu, Shiyu Chang, Yang Zhang, and Tommi Jaakkola. Rethinking cooperative rationalization: Introspective extraction and complement control. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4085–4094, 2019. 2
- [38] Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. Interpreting cnns via decision trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6261–6270, 2019. 2

## A. Unary concept correlations $\Theta^u$

We present the unary concept correlations  $\Theta^u$  in Figure 10. The correlation between any pair of synonyms is close to 0, the correlation between words belonging to the same super concept set is close to 2, and the correlation between words belonging to two different super concept sets is in the middle of the range [0, 2]. Therefore, a GMM with three components whose mean values are initialized with 0, 1 and 2 can successfully generate the concepts and super concept sets.

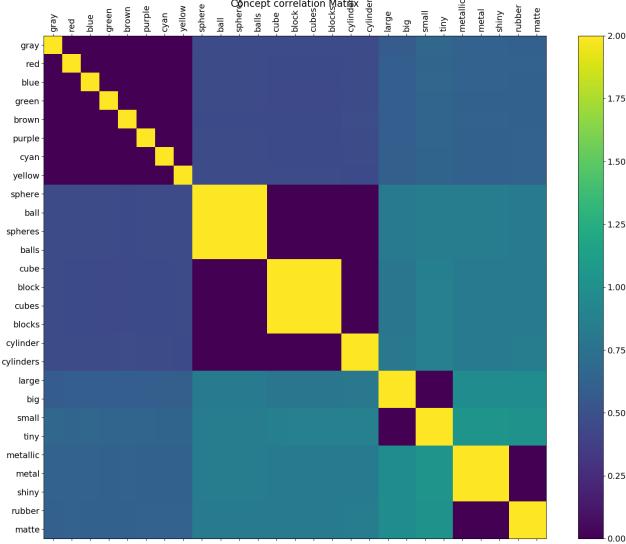


Figure 10. The unary concept correlations  $\Theta^u$ .

## B. Derivation from the concept interpretation

With the induced concepts and super concept sets, each object can be represented with a zero-one vector,  $k$ , where the entry is 1 if that object possesses the corresponding concept or 0 otherwise. Notice that the super concept sets split the whole concept set; we thereby name the entries of  $k$  corresponding to one super concept set as a super concept. The super concept is thus a zero-one vector with exactly one entry to be 1. We name this pattern as the super concept constraint. Therefore, we can define the semantic distance between two visual objects by the number of different super concepts or by Eqn. (5).

$$\zeta_{k_1, k_2} = \frac{|k_1 \oplus k_2|_1}{2}, \quad (5)$$

where  $k_1$  and  $k_2$  are the concept vectors representing two objects and  $\oplus$  is the operation XOR. Studying the concepts and super concept sets induced, we acknowledge that the super concept sets correspond to color, shape, size and material in semantics. Thereby, we give an example of the semantic distances of multiple objects to one object as shown

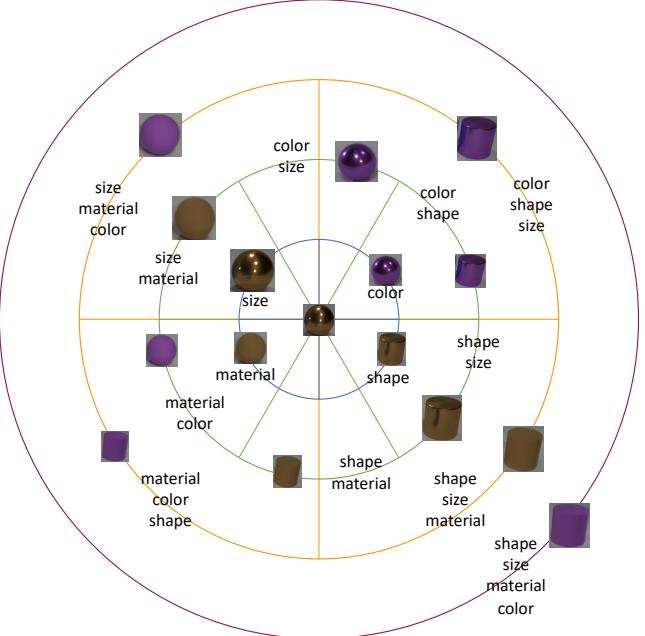


Figure 11. Illustration of the semantic distance.

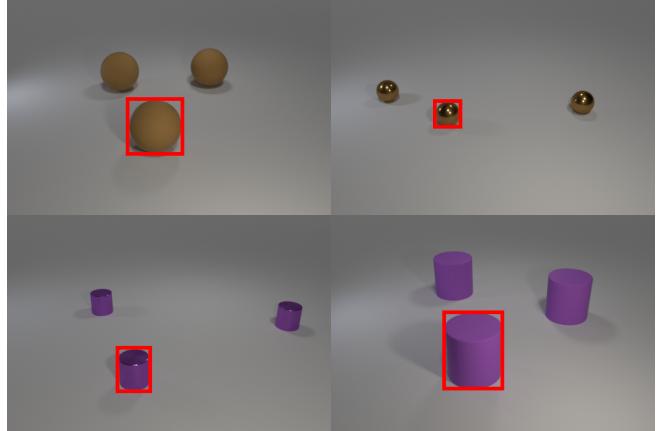


Figure 12. The original images for extracting visual features. The object-level features corresponding to the objects bounded by red rectangles are used for the illustration of semantic operations in the visual feature space.

in Figure 11. The circle radii indicate the semantic distances to the object at the centers of these circles. The inner three circles are segmented so that each segment represents what super concepts are different. The outer circle represents all the 4 super concepts are different between the object on that circle and the object at the center.

We can further interpret the semantic analogy in the visual feature space with the induced concept vectors. Shown in Figure 12, we first generate four images of different objects; then, we use our trained OCCAM structure to extract the object-level features corresponding to the objects bounded by red rectangles. Shown in Figure 13(a), we can

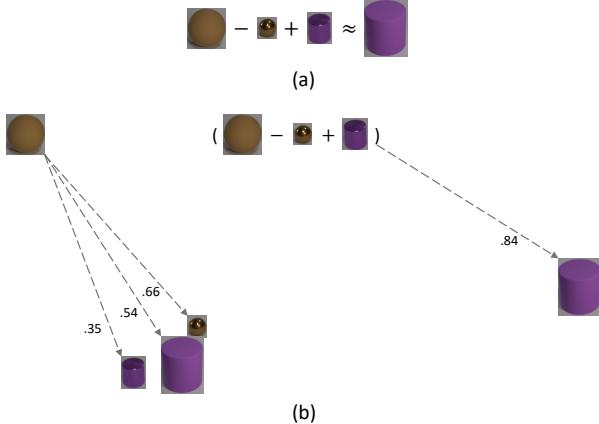


Figure 13. Illustration of the semantic analogy in the visual feature space. (a) The operations on the visual features. (b) The cosine similarities between pairs of visual feature vectors.

	$\backslash$	$\oplus$	=
gray	0	0	0
red	0	0	0
blue	0	0	0
green	0	0	0
brown	1	1	0
purple	0	0	1
cyan	0	0	0
yellow	0	0	0
sphere	1	1	0
cube	0	0	0
cylinder	0	0	1
large	1	0	0
small	0	1	1
metal	0	1	0
matte	1	0	1

Figure 14. Operations on the concept vectors.

move the visual feature vector of the leftmost object closer to that of the rightmost object by subtracting and adding visual feature vectors of two other objects. The proximity between pairs of visual feature vectors is measured with cosine similarity as shown in Figure 13(b). In the concept vector space, we can define a 'minus' operation,  $k_1 \setminus k_2$ , as eliminate the shared super concepts between  $k_1$  and  $k_2$  from  $k_1$ . We can also define a 'plus' operation,  $k'_1 \oplus k_2$ , between a concept vector template  $k'_1$  and a concept vector  $k_2$  as add the super concepts of  $k_2$  that  $k'_1$  misses to  $k'_1$ . Therefore, The operations in the visual feature space can be explained with the operations we defined in the concept vector space shown in Figure 14.

### C. Visualization of reasoning steps

We give an example of the compositional reasoning steps on the induced concept space of OCCAM as shown in Figure 15. While the attention is directly imposed on the projected concept vectors in the read unit of the compositional

reasoning module, the attention can be equally mapped to the concept vectors and the visual objects as the projected concept vector to the concept vector or the projected concept vector to the visual object is a one-to-one mapping relationship.

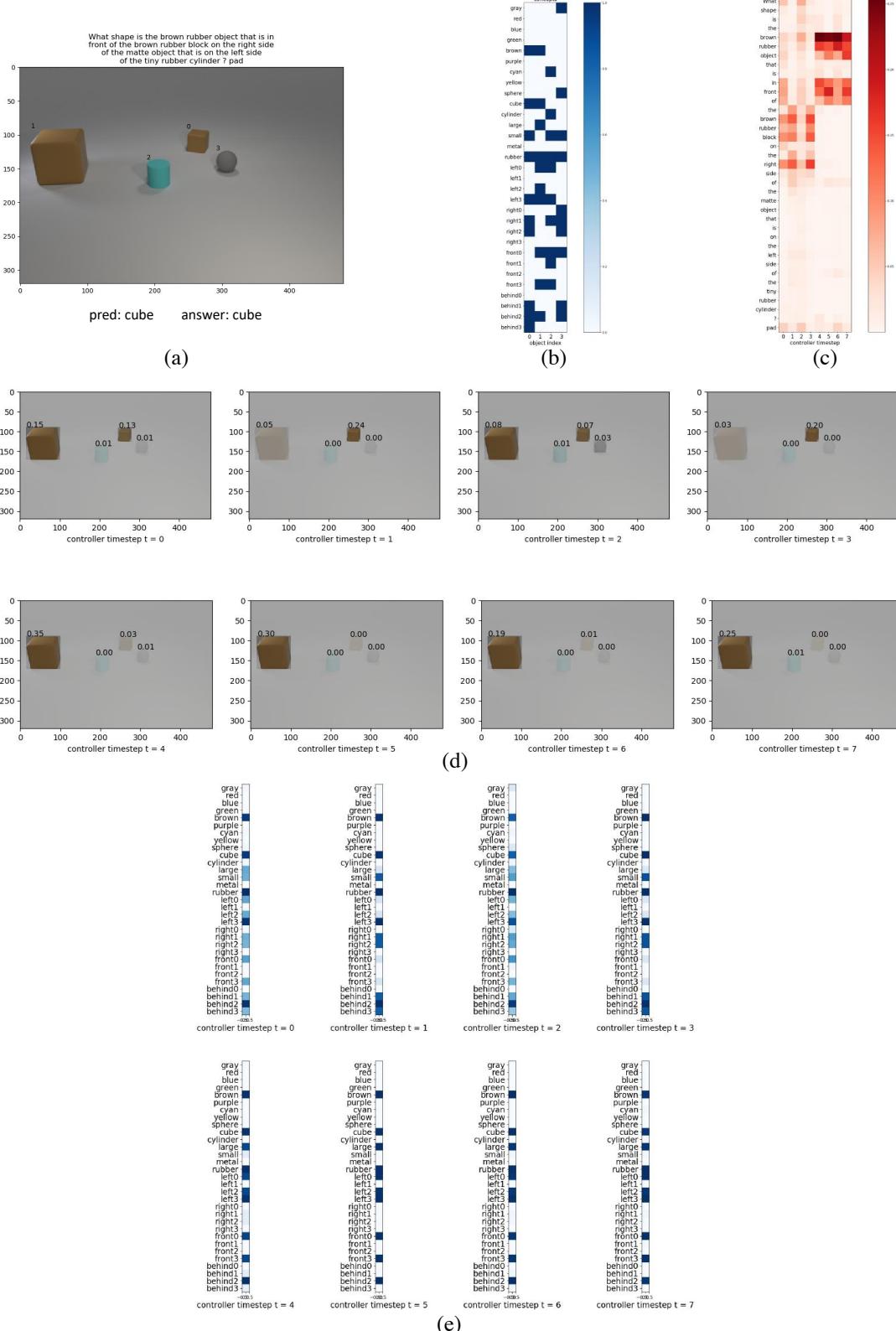


Figure 15. Visualization of reasoning steps. (a) The question, image, prediction and ground truth answer. The index of each object is shown on the upper left of the object. (b) The induced concepts of objects and relations. (c) The stepwise attentions on question words. (d) The stepwise attentions on objects. (e) The concept vector read into the memory of the reasoning module in each step.