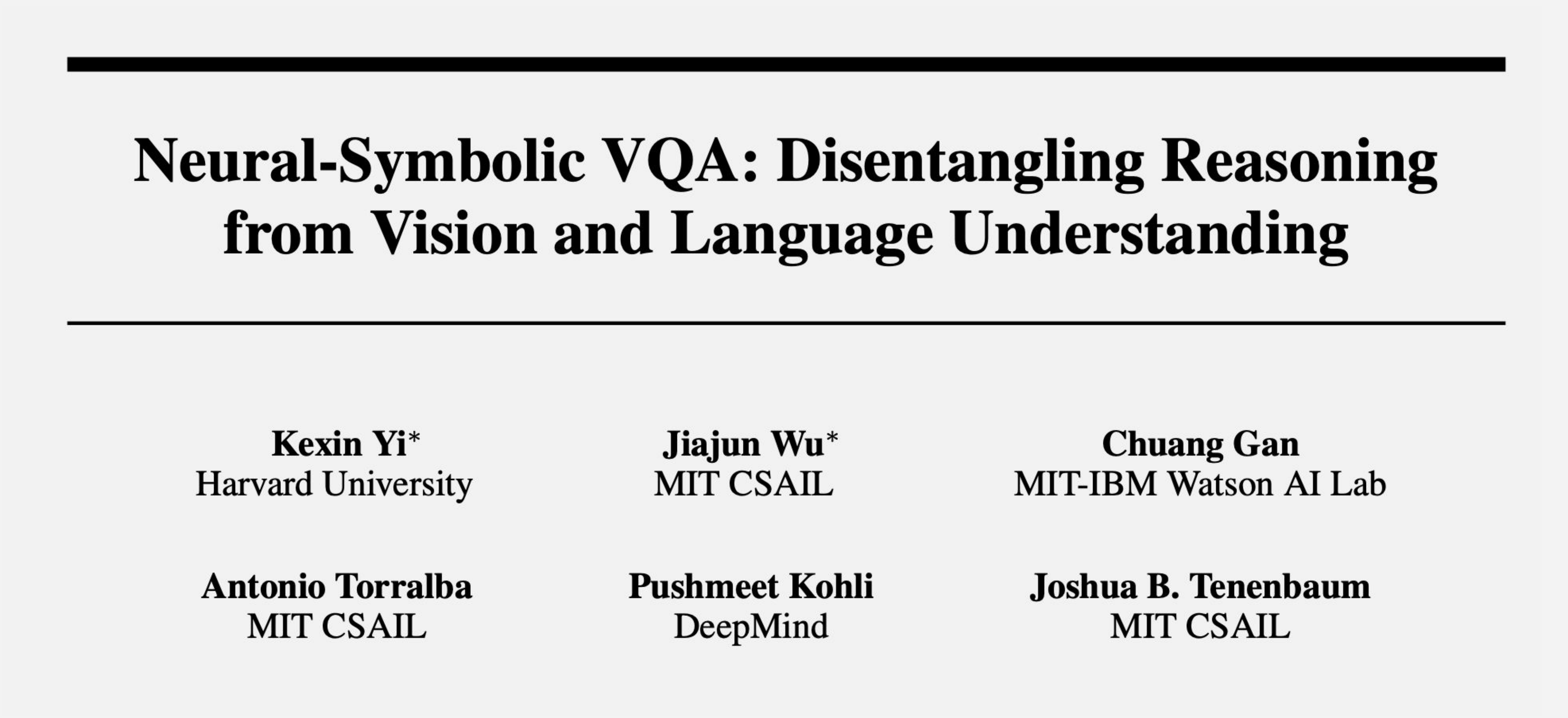
CS 957, System-2 AI
Neuro-Symbolic AI

Mohammad Hossein Rohban | Feb 2025

Sharif University of Technology



Neural-Symbolic VQA: Disentangling Reasoning from Vision and Language Understanding

Kexin Yi*
Harvard University

Jiajun Wu*
MIT CSAIL

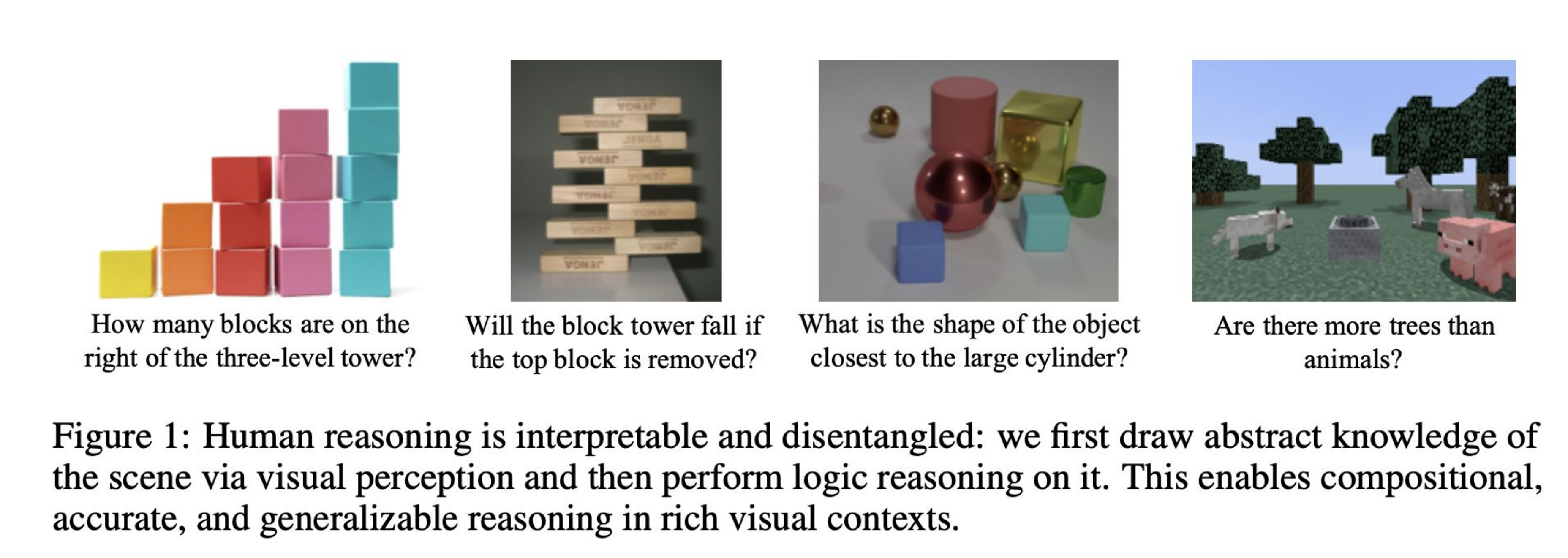
Chuang Gan
MIT-IBM Watson AI Lab

Antonio Torralba
MIT CSAIL

Pushmeet Kohli
DeepMind

Joshua B. Tenenbaum
MIT CSAIL

Visual Question Answering



How many blocks are on the right of the three-level tower?

Will the block tower fall if the top block is removed?

What is the shape of the object closest to the large cylinder?

Are there more trees than animals?

Figure 1: Human reasoning is interpretable and disentangled: we first draw abstract knowledge of the scene via visual perception and then perform logic reasoning on it. This enables compositional, accurate, and generalizable reasoning in rich visual contexts.

CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning

Abstract

When building artificial intelligence systems that can reason and answer questions about visual data, we need diagnostic tests to analyze our progress and discover shortcomings. Existing benchmarks for visual question answering can help, but have strong biases that models can exploit to correctly answer questions without reasoning. They also conflate multiple sources of error, making it hard to pinpoint model weaknesses. We present a diagnostic dataset that tests a range of visual reasoning abilities. It contains minimal biases and has detailed annotations describing the kind of reasoning each question requires. We use this dataset to analyze a variety of modern visual reasoning systems, providing novel insights into their abilities and limitations.

Justin Johnson

Bharath Hariharan

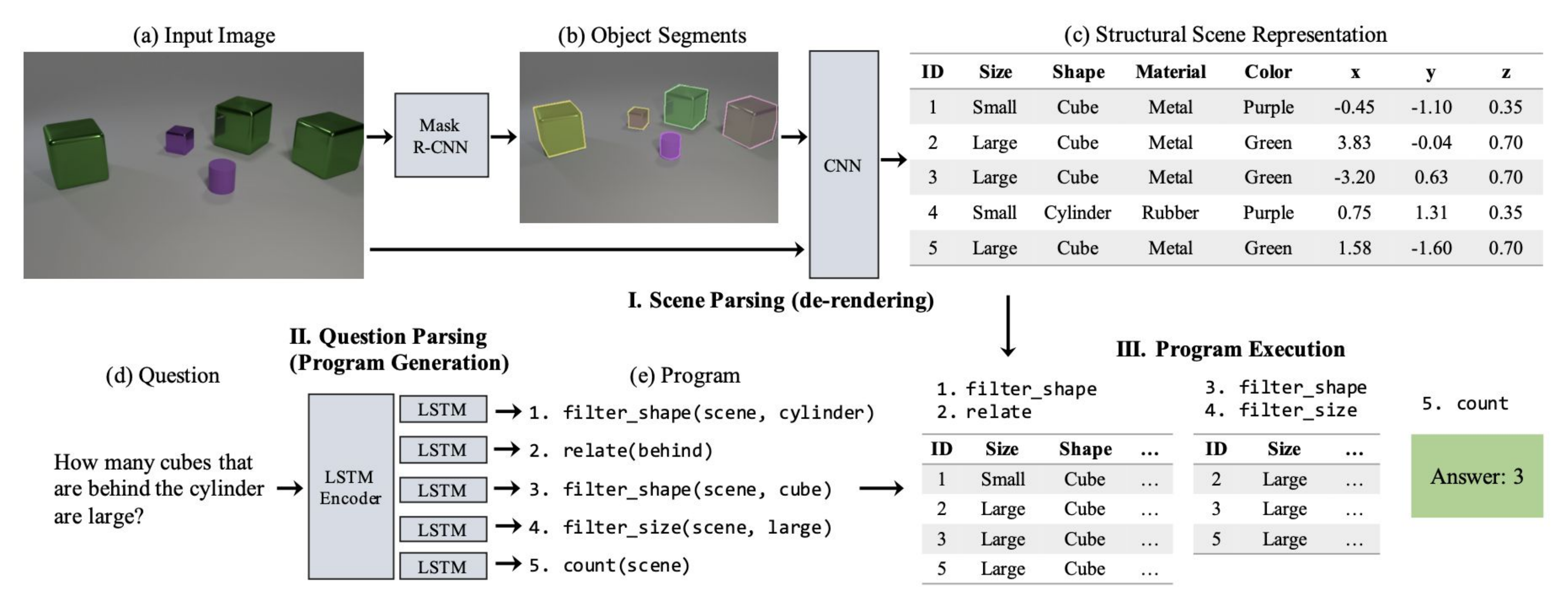
Laurens van der Maaten

Fei-Fei Li

Larry Zitnick

Ross Girshick

Disentangling Perception from Reasoning


(a) Input Image
Mask R-CNN
(b) Object Segments
CNN
(c) Structural Scene Representation

ID	Size	Shape	Material	Color	x	y	z
1	Small	Cube	Metal	Purple	-0.45	-1.10	0.35
2	Large	Cube	Metal	Green	3.83	-0.04	0.70
3	Large	Cube	Metal	Green	-3.20	0.63	0.70
4	Small	Cylinder	Rubber	Purple	0.75	1.31	0.35
5	Large	Cube	Metal	Green	1.58	-1.60	0.70

I. Scene Parsing (de-rendering)
II. Question Parsing (Program Generation)
(d) Question

How many cubes that are behind the cylinder are large?

LSTM Encoder
LSTM
LSTM
LSTM
LSTM
LSTM
(e) Program

```
1. filter_shape(scene, cylinder)
2. relate(behind)
3. filter_shape(scene, cube)
4. filter_size(scene, large)
5. count(scene)
```

```
1. filter_shape
2. relate
```

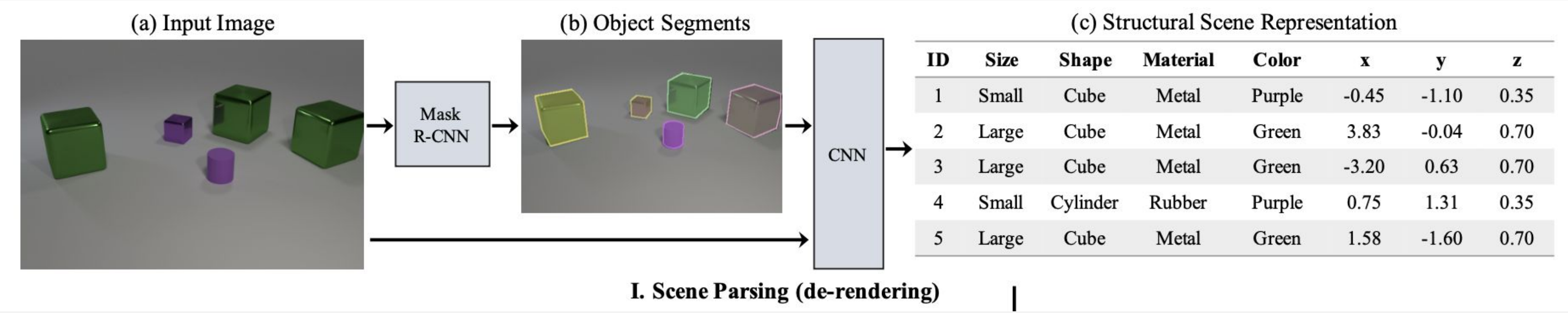
ID	Size	Shape	…
1	Small	Cube	…
2	Large	Cube	…
3	Large	Cube	…
5	Large	Cube	…

```
3. filter_shape
4. filter_size
```

ID	Size	…
2	Large	…
3	Large	…
5	Large	…

```
5. count
```

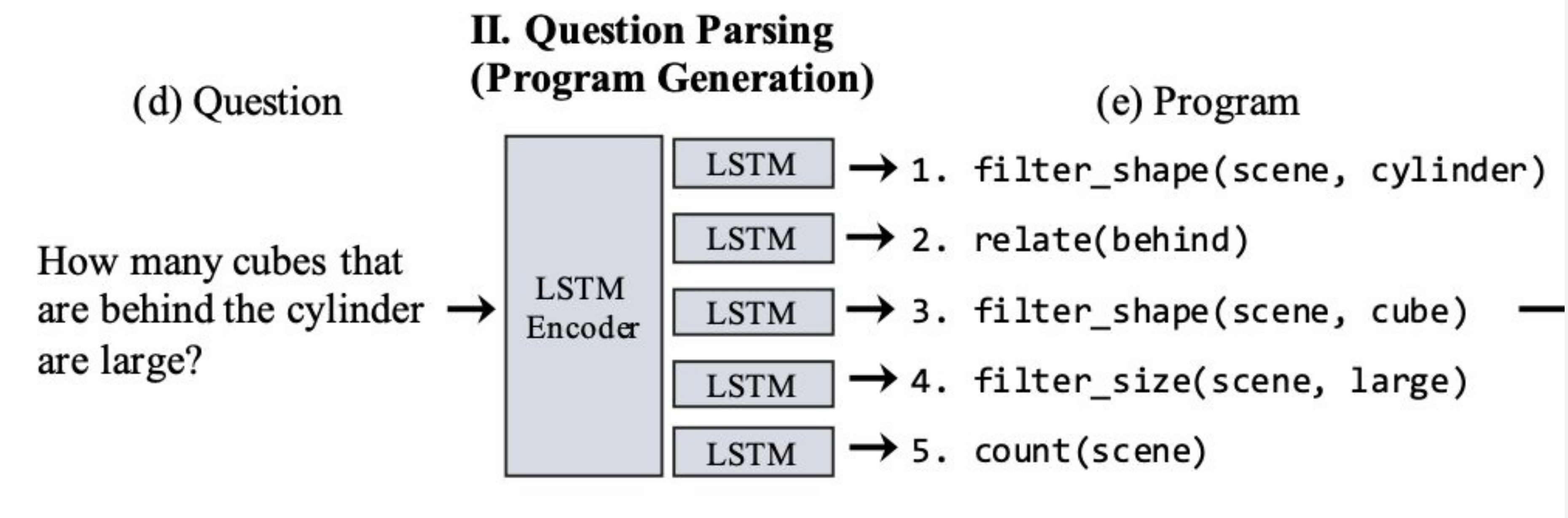
Scene Parsing



ID	Size	Shape	Material	Color	x	y	z
1	Small	Cube	Metal	Purple	-0.45	-1.10	0.35
2	Large	Cube	Metal	Green	3.83	-0.04	0.70
3	Large	Cube	Metal	Green	-3.20	0.63	0.70
4	Small	Cylinder	Rubber	Purple	0.75	1.31	0.35
5	Large	Cube	Metal	Green	1.58	-1.60	0.70

I. Scene Parsing (de-rendering)

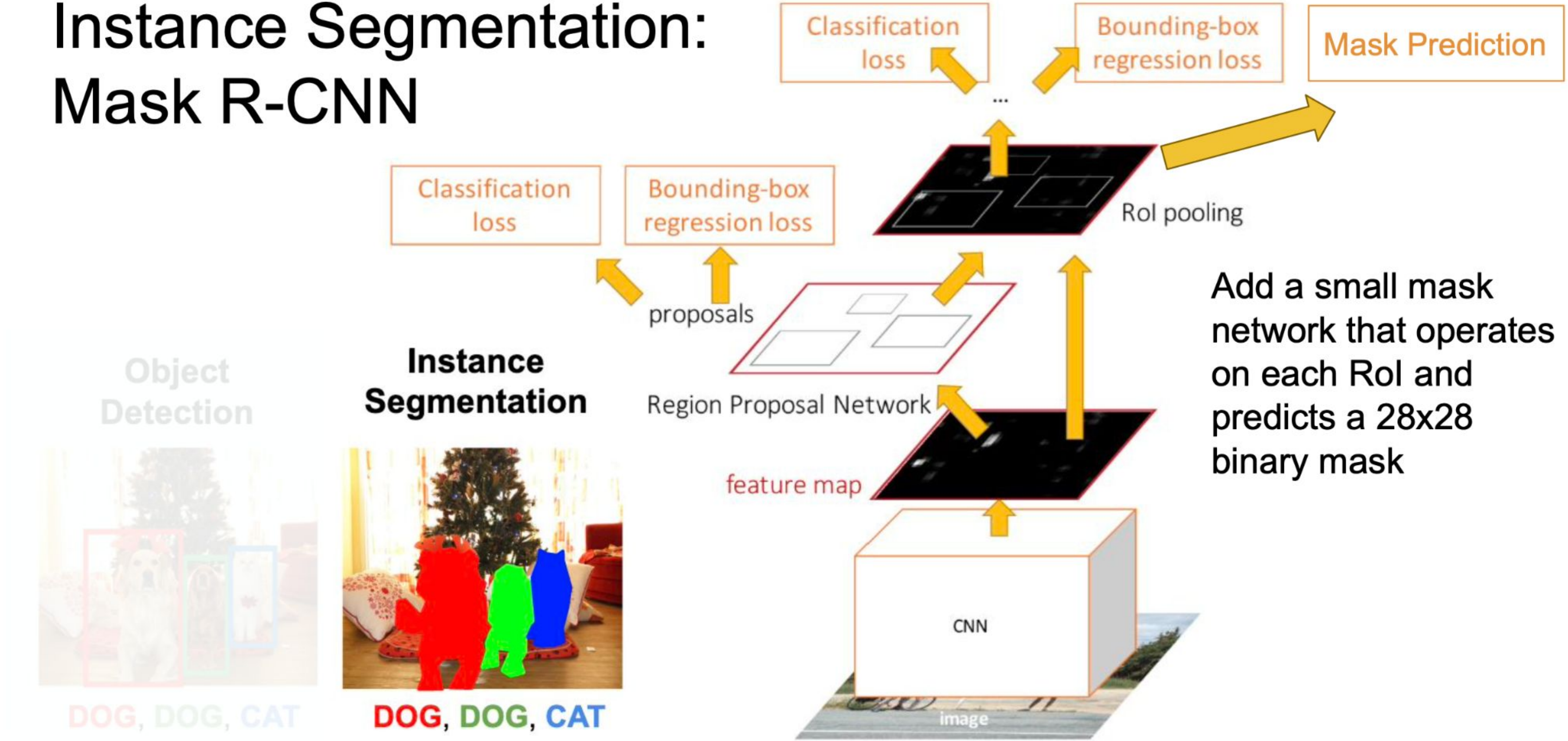
Question Parsing



How many cubes that are behind the cylinder are large?

```
1. filter_shape(scene, cylinder)
2. relate(behind)
3. filter_shape(scene, cube)
4. filter_size(scene, large)
5. count(scene)
```

Instance Segmentation: Mask R-CNN



Add a small mask network that operates on each RoI and predicts a 28x28 binary mask

He et al, “Mask R-CNN”, ICCV 2017

Courtesy: CS 231N, Stanford

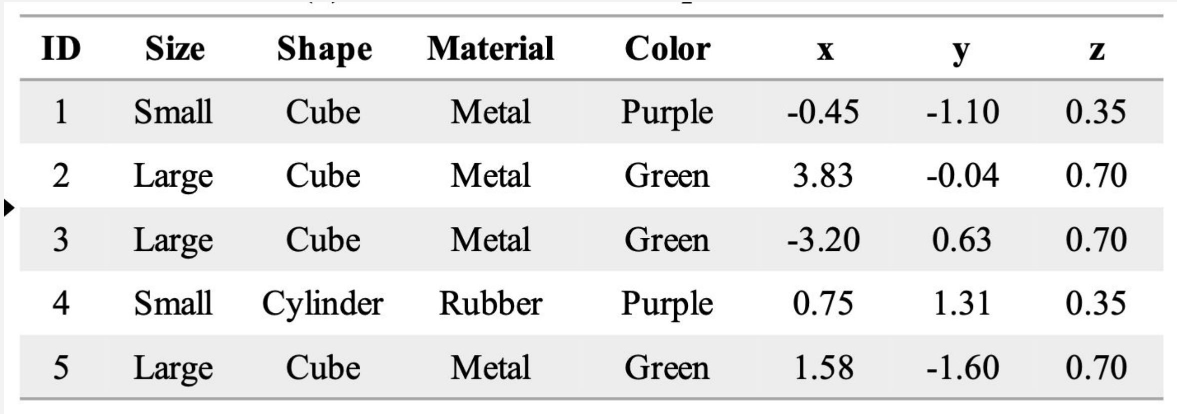
Scene Parsing

Separate heads to predict

- Color, material, size, and shape

Each object resized to 224*224 and together with original image fed into a ResNet-34

This determines the pose and 3D coordinates of each object



ID	Size	Shape	Material	Color	x	y	z
1	Small	Cube	Metal	Purple	-0.45	-1.10	0.35
2	Large	Cube	Metal	Green	3.83	-0.04	0.70
3	Large	Cube	Metal	Green	-3.20	0.63	0.70
4	Small	Cylinder	Rubber	Purple	0.75	1.31	0.35
5	Large	Cube	Metal	Green	1.58	-1.60	0.70

Scene Parsing Training

ResNet-50 FPN backbone

Trained on 4,000 CLEVR images

For 30,000 iterations with batch size = 8 (for FPN, 50 and continuous attribute net)

Question Parser

Attention-based Bidirectional LSTM

$$e_i = [e_i^F, e_i^B], \quad \text{where} \quad e_i^F, h_i^F = \mathrm{LSTM}(\Phi_E(x_i), h_{i-1}^F), \quad e_i^B, h_i^B = \mathrm{LSTM}(\Phi_E(x_i), h_{i+1}^B). \tag{1}$$

$$q_t = \mathrm{LSTM}(\Phi_D(y_{t-1})), \qquad \alpha_{ti} \propto \exp(q_t^\top W_A e_i), \qquad c_t = \sum_i \alpha_{ti} e_i.$$

$$y_t \sim \mathrm{softmax}(W_O[q_t, c_t]).$$

Question Parser Training

Pre-training with ~ 100-300 question-program pairs 20k iterations

Post-training with ~ 9K-700k questions-solution pairs 2M iterations with early stopping

- Reward: 1 if program execution leads to correct answer; 0 otherwise
- REINFORCE as the training algorithm.

REINFORCE

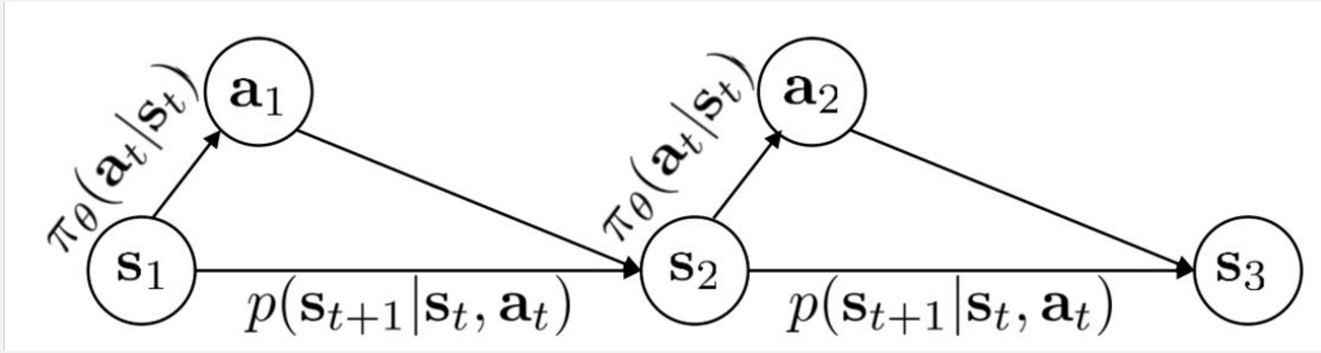
Optimizing the expected return J(θ):

$$\theta^{\star} = \arg\max_{\theta} \underbrace{E_{\tau\sim p_\theta(\tau)}\left[\sum_t r(\mathbf{s}_t, \mathbf{a}_t)\right]}_{J(\theta)}$$

$$J(\theta) = E_{\tau\sim p_\theta(\tau)}[\underbrace{r(\tau)}_{\sum_{t=1}^{T} r(\mathbf{s}_t, \mathbf{a}_t)}] = \int p_\theta(\tau) r(\tau) d\tau$$

$$\nabla_\theta J(\theta) = \int \underline{\nabla_\theta p_\theta(\tau)} r(\tau) d\tau = \int \underline{p_\theta(\tau) \nabla_\theta \log p_\theta(\tau)} r(\tau) d\tau = E_{\tau\sim p_\theta(\tau)}[\nabla_\theta \log p_\theta(\tau) r(\tau)]$$

REINFORCE (cont.)


$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$
$\mathbf{a}_1$
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$
$\mathbf{a}_2$
$\mathbf{s}_1$
$\mathbf{s}_2$
$\mathbf{s}_3$
$p(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{a}_t)$
$p(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{a}_t)$

$$\underbrace{p_\theta(\mathbf{s}_1,\mathbf{a}_1,\ldots,\mathbf{s}_T,\mathbf{a}_T)}_{p_\theta(\tau)} = p(\mathbf{s}_1)\prod_{t=1}^{T}\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)p(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{a}_t)$$

log of both sides

$$\log p_\theta(\tau) = \log p(\mathbf{s}_1) + \sum_{t=1}^{T}\log\pi_\theta(\mathbf{a}_t|\mathbf{s}_t) + \log p(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{a}_t)$$

REINFORCE (cont.)

$$\theta^\star = \arg\max_\theta J(\theta)$$

$$J(\theta) = E_{\tau\sim p_\theta(\tau)}[r(\tau)]$$

$$\nabla_\theta J(\theta) = E_{\tau\sim p_\theta(\tau)}[\nabla_\theta \underline{\log p_\theta(\tau)} r(\tau)]$$

$$\nabla_\theta \overbrace{\left[\log \cancel{p}(\mathbf{s}_1) + \sum_{t=1}^{T} \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) + \log \cancel{p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)}\right]}$$

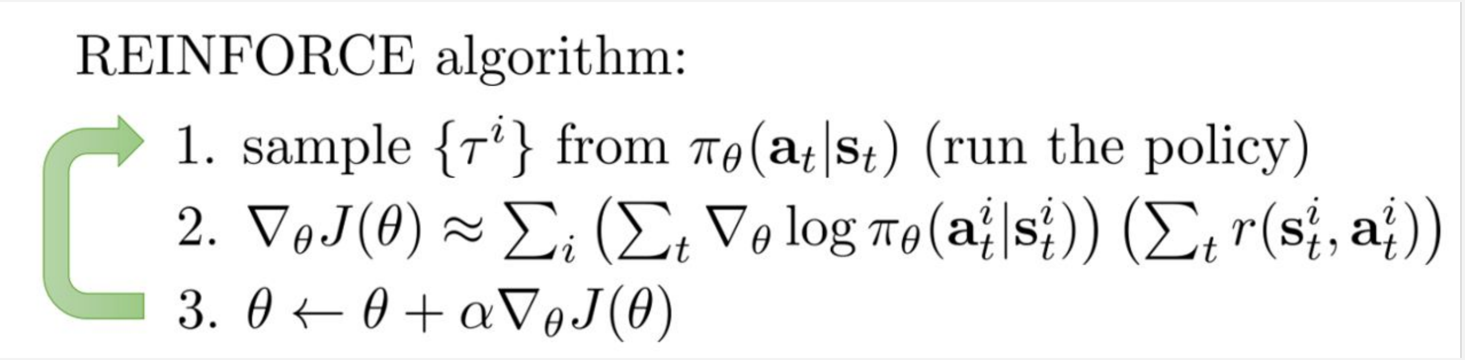
$$\nabla_\theta J(\theta) = E_{\tau\sim p_\theta(\tau)}\left[\left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)\right)\left(\sum_{t=1}^{T} r(\mathbf{s}_t, \mathbf{a}_t)\right)\right]$$

REINFORCE (cont.)

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \right) \left(\sum_{t=1}^{T} r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) \right)$$

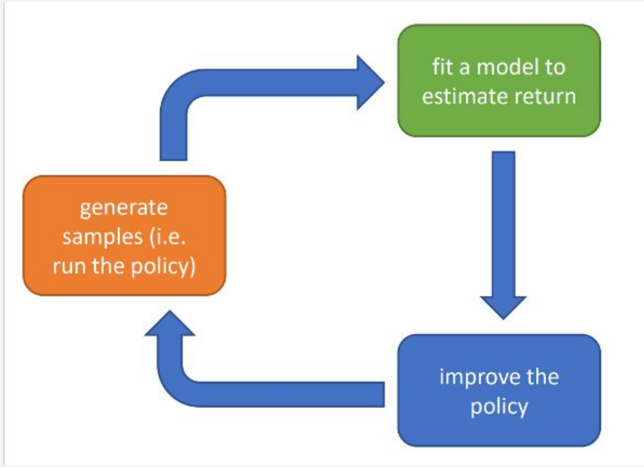
$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$$

REINFORCE (cont.)



REINFORCE algorithm:

1. sample $\{\tau^i\}$ from $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ (run the policy)
2. $\nabla_\theta J(\theta) \approx \sum_i \left(\sum_t \nabla_\theta \log \pi_\theta(\mathbf{a}_t^i|\mathbf{s}_t^i)\right) \left(\sum_t r(\mathbf{s}_t^i, \mathbf{a}_t^i)\right)$
3. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

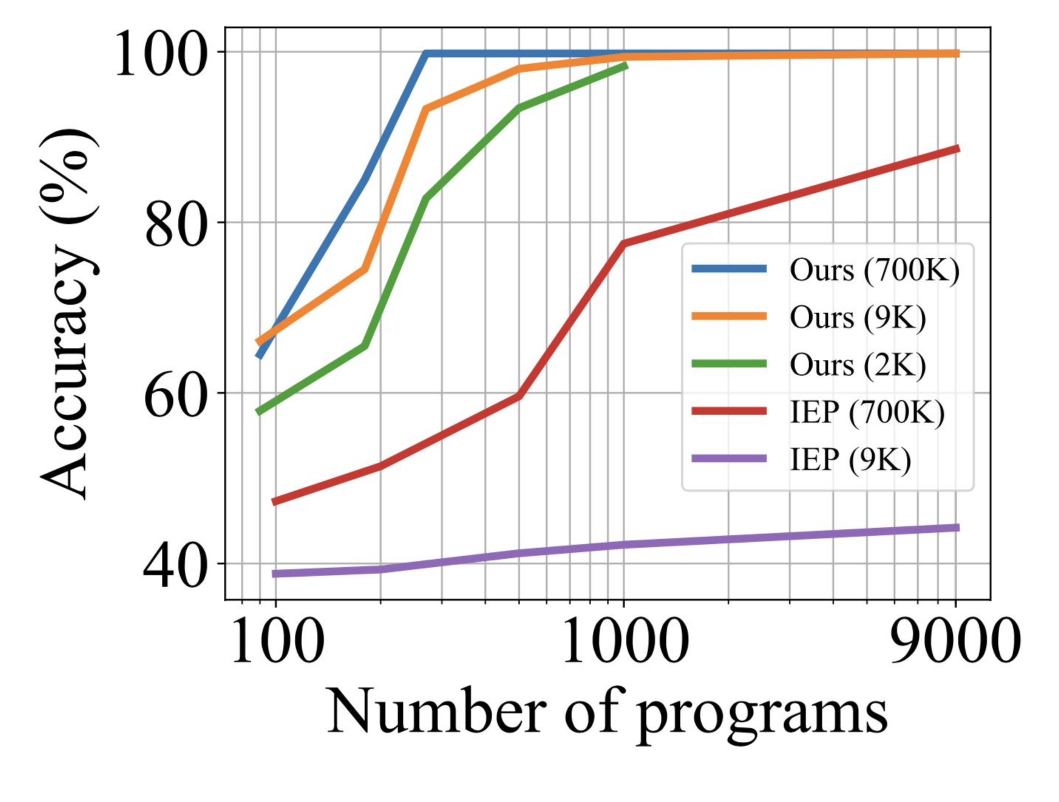


fit a model to estimate return

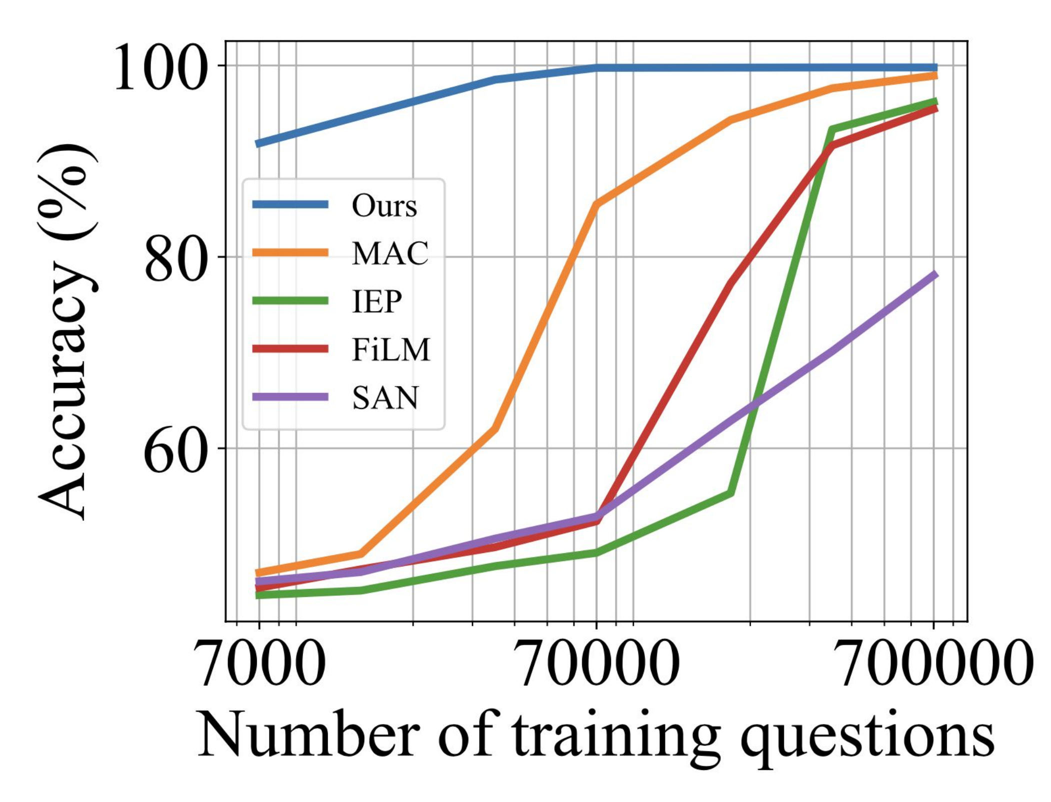
improve the policy

generate samples (i.e. run the policy)

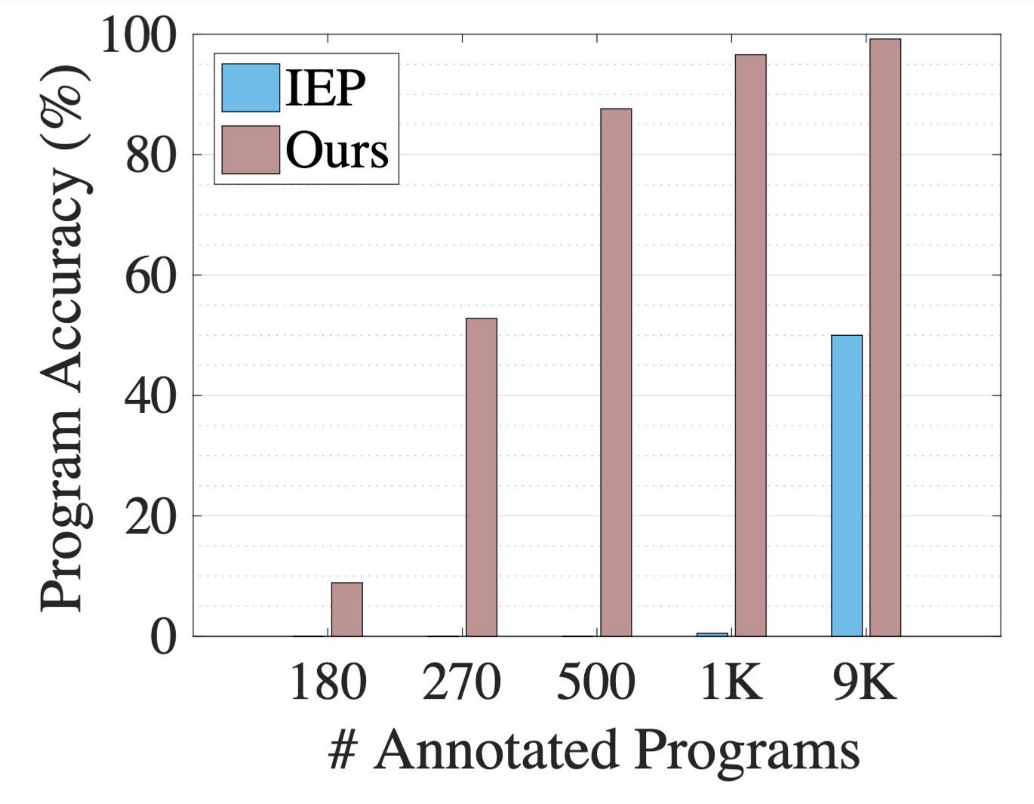
Sample Efficient in pre-training



Sample Efficient in post-training



Generated Programs are more Accurate



Generalization on Unseen Attribute Compositions

CLEVR-CoGenT dataset

Two splits A and B

split A only contains cubes that are either gray, blue, brown or yellow, and cylinders that are red, green, purple or cyan;

split B has the **opposite** color-shape pairs for cubes and cylinders.

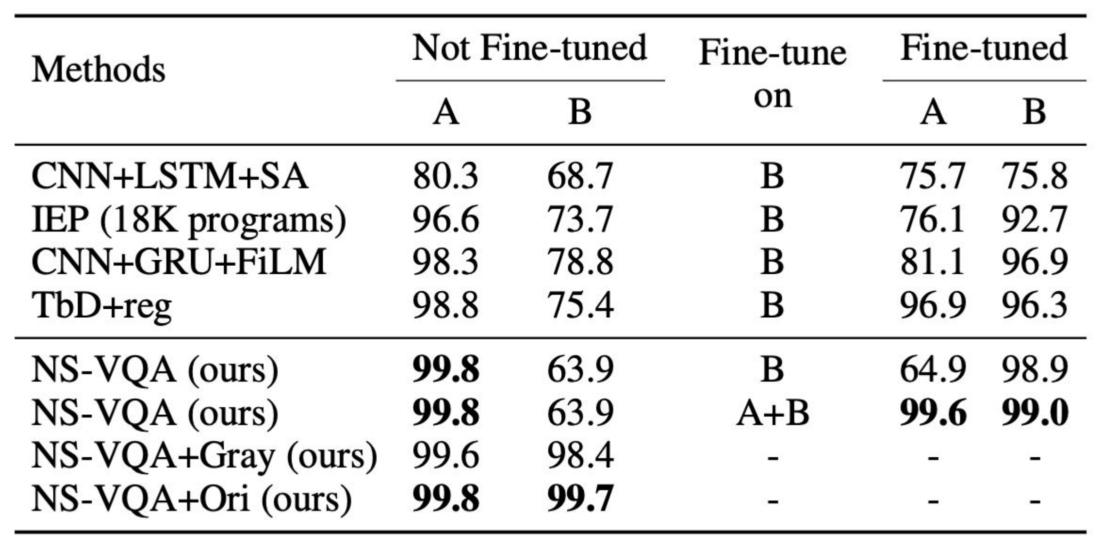
Both splits contain spheres of any color.

Split A has 70K images and 700K questions for training and both splits have 15K images and 150K questions for evaluation and testing.

Results

NS-VQA+Ori is when image parser trained on the original condition (i.e. the same as in CLEVR)

NS-VQA+Gray is when a gray-scale image is used as input to the scene parser.



Methods	Not Fine-tuned		Fine-tune on	Fine-tuned	
	A	B		A	B
CNN+LSTM+SA	80.3	68.7	B	75.7	75.8
IEP (18K programs)	96.6	73.7	B	76.1	92.7
CNN+GRU+FiLM	98.3	78.8	B	81.1	96.9
TbD+reg	98.8	75.4	B	96.9	96.3
NS-VQA (ours)	**99.8**	63.9	B	64.9	98.9
NS-VQA (ours)	**99.8**	63.9	A+B	**99.6**	**99.0**
NS-VQA+Gray (ours)	99.6	98.4	-	-	-
NS-VQA+Ori (ours)	**99.8**	**99.7**	-	-	-

Generalization to Human Questions

CLEVR-Humans: human-generated questions on CLEVR images

Setup:

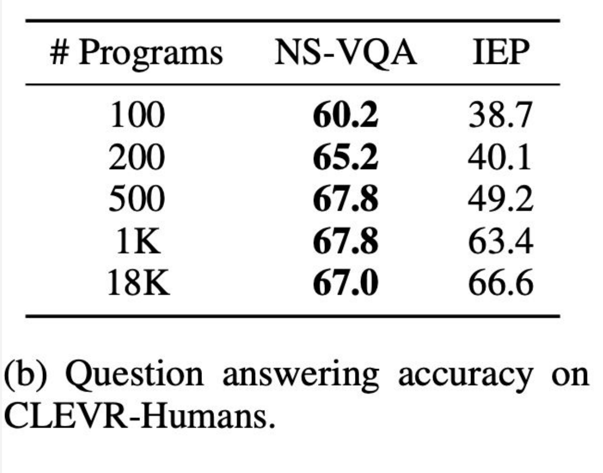
- pretrain the model with a limited number of programs from CLEVR,
- fine-tune it on CLEVR-Humans with REINFORCE.
- initialize the encoder word embedding by the GloVe word vectors and keep it fixed during pre-training.

The REINFORCE stage lasts for at most 1M iterations; early stop is applied.

Results

NS-VQA outperforms IEP on CLEVR-Humans by a considerable margin under **small amount of annotated programs**.

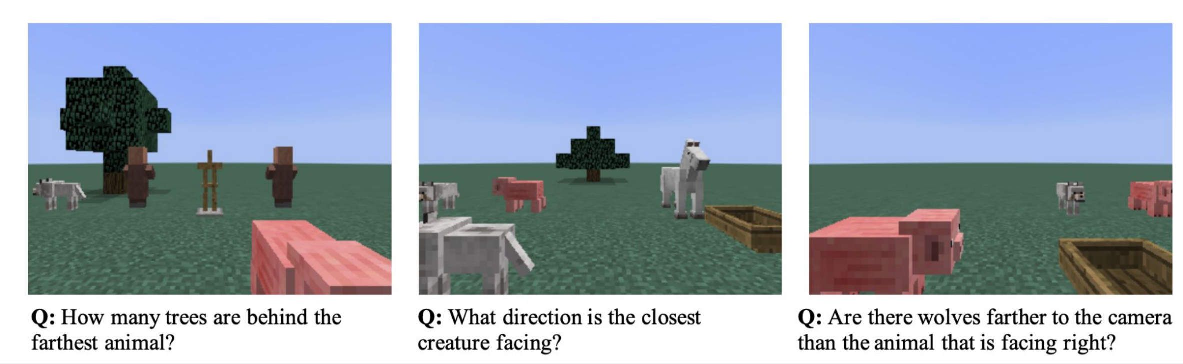
structural scene representation and symbolic program executor helps to exploit the **strong exploration power** of REINFORCE



# Programs	NS-VQA	IEP
100	**60.2**	38.7
200	**65.2**	40.1
500	**67.8**	49.2
1K	**67.8**	63.4
18K	**67.0**	66.6

(b) Question answering accuracy on CLEVR-Humans.

Extending to New Scene Context


Q: How many trees are behind the farthest animal?

Q: What direction is the closest creature facing?

Q: Are there wolves farther to the camera than the animal that is facing right?

Objects and scenes are taken from Minecraft.

Render 10,000 Minecraft scenes.

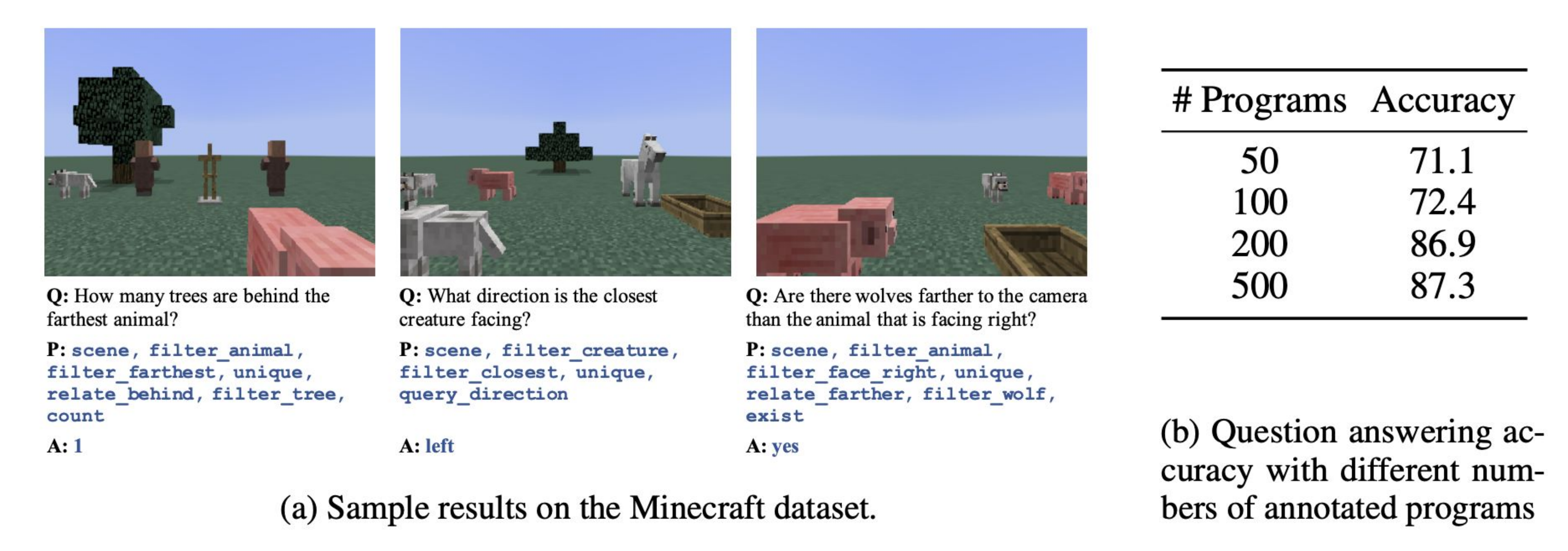
Each image consists of 3 to 6 objects, and each object is sampled from a set of 12 entities.

Minecraft hosts a larger set of 3D objects with richer image content and visual appearance;

Questions and programs involve hierarchical attributes. For example, a “wolf” and a “pig” are both “animals”, and an “animal” and a “tree” are both “creatures”.

Results

Setup: use the first 9,000 images with 88,109 questions for training and the remaining 1,000 images with 9,761 questions for testing.



Q: How many trees are behind the farthest animal?

P: `scene, filter_animal, filter_farthest, unique, relate_behind, filter_tree, count`

A: 1

Q: What direction is the closest creature facing?

P: `scene, filter_creature, filter_closest, unique, query_direction`

A: left

Q: Are there wolves farther to the camera than the animal that is facing right?

P: `scene, filter_animal, filter_face_right, unique, relate_farther, filter_wolf, exist`

A: yes

(a) Sample results on the Minecraft dataset.

# Programs	Accuracy
50	71.1
100	72.4
200	86.9
500	87.3

(b) Question answering accuracy with different numbers of annotated programs