Inferring and Executing Programs for Visual Reasoning

Justin Johnson, Bharath Hariharan, Lauren van der Maaten,
Judy Hoffman, Li Fei-Fei, C. Lawrence Zitnick, Ross Girshick
Stanford University, Facebook AI Research

Presented by Hannes Perrot, 12.06.2018

Problem definition

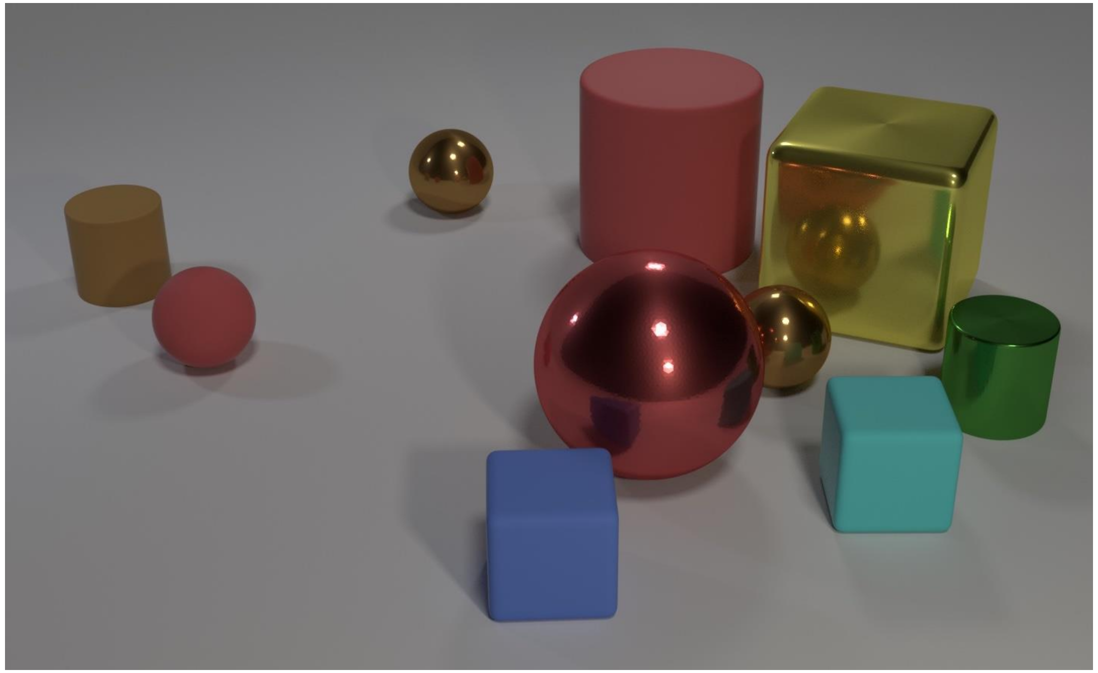
Is there a pedestrian in my lane?

- Inferring and Executing Programs
- Visual Reasoning:
  - the process of thinking about something in order to make a decision [Cambridge dictionary]
- Given: Image and question
- Come up with decision

Agenda

- Problem definition
  - CLEVER Dataset
- Method
  - Programs
  - Functions
  - Program generator
  - Execution engine
  - Training
- Experiments
  - Comparison training procedures
  - What do the modules learn?
  - Generalizing to new attribute combinations
  - Generalzing to new question types
  - CLEVER-Humans
- Conclusion

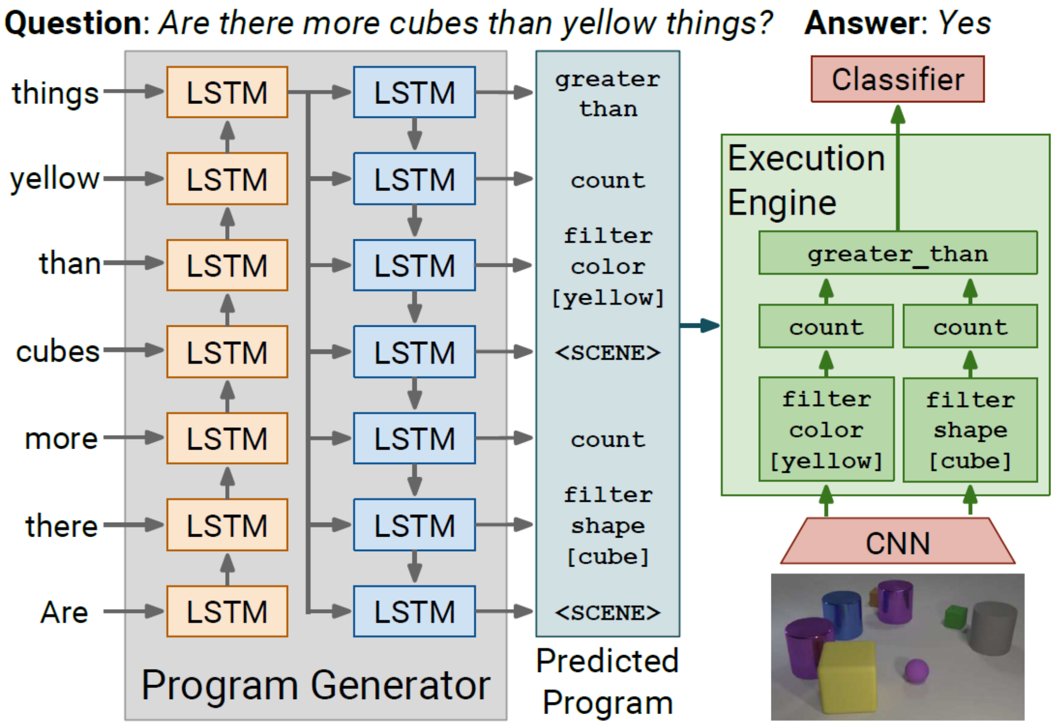
Clever Dataset



- Attribute identification, counting, comparison, spatial relationships, logical operations
- Are there an equal number of large things and metal spheres?
- What size is the cylinder that is left of the brown metal thing that is left of the big sphere?
- There is a sphere with the same size as the metal cube; is it made of the same material as the small red sphere?

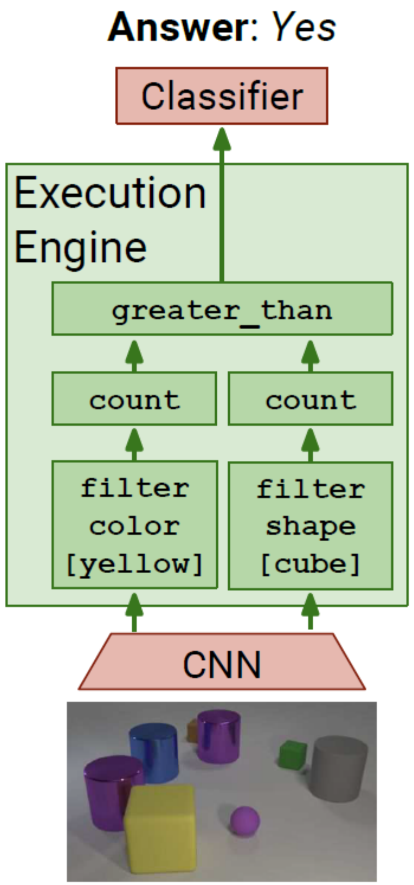
Method Overview

- Separate program generator and execution engine
- Program generator and execution engine are neural networks
- Trained by backpropagation and REINFORCE



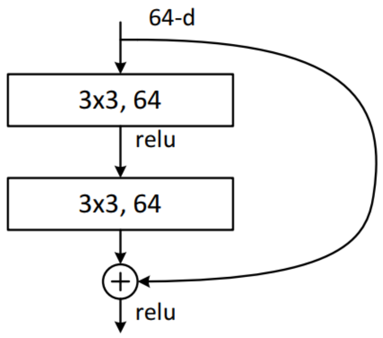
Programs

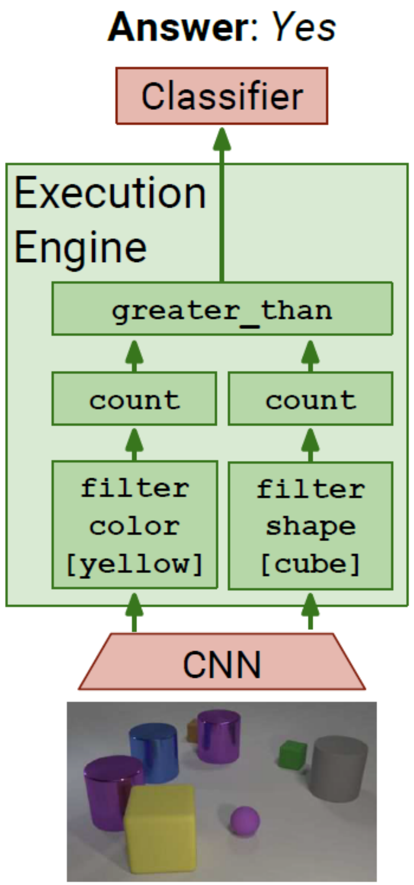
- The programs are composed of functions
- -> like in a normal programming language
- Fixed set of functions
- Functions have a predefined arity -> 1 or 2 inputs



Functions

- Output CxHxW
- SCENE
  - Visual features (output of conv4 from ResNet-101) as input
  - 4 convolutional layers
- Unary functions
  - Residual block
  - e.g. count

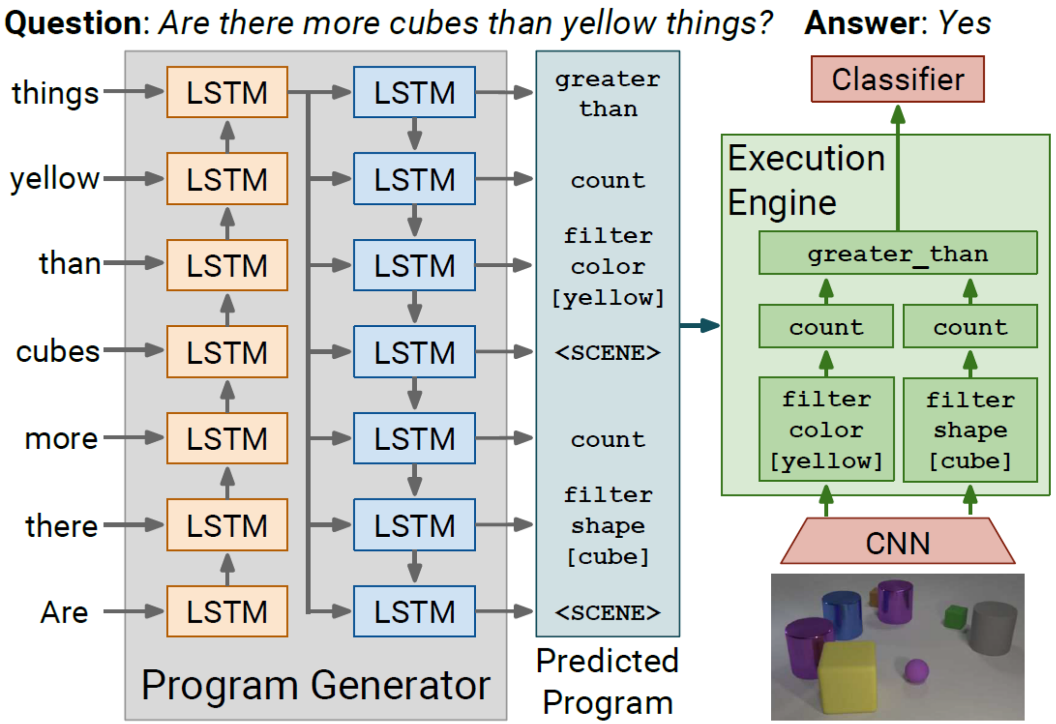




- Binary functions
  - Concatenate inputs along channel dim
  - Reduce channels using 1x1 convolution
  - Residual block
  - e.g. greater_than
- Classifier
  - Final output flattened
  - multilayer perceptron classifier

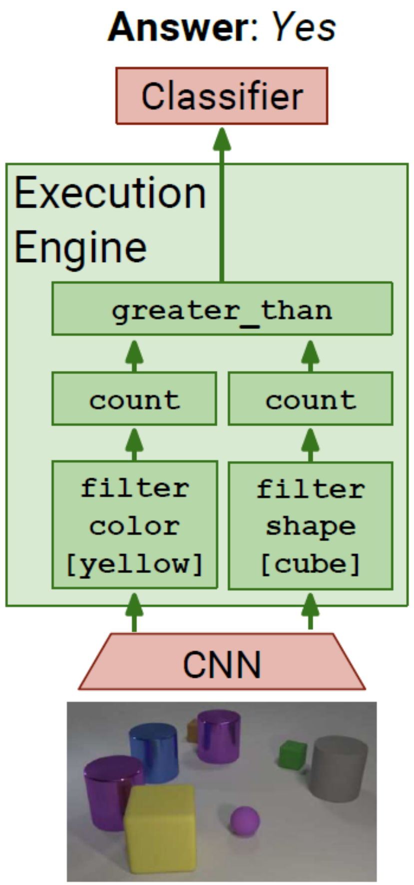
Program generator

- Predicts program from natural language question
- Programs are traversed to receive sequence of functions
- Use standard LSTM sequence-to-sequence model for program prediction



Execution engine

- Predicts answer given program and input image
- Implemented using neural networks
- Every syntactically correct program is executable



Training

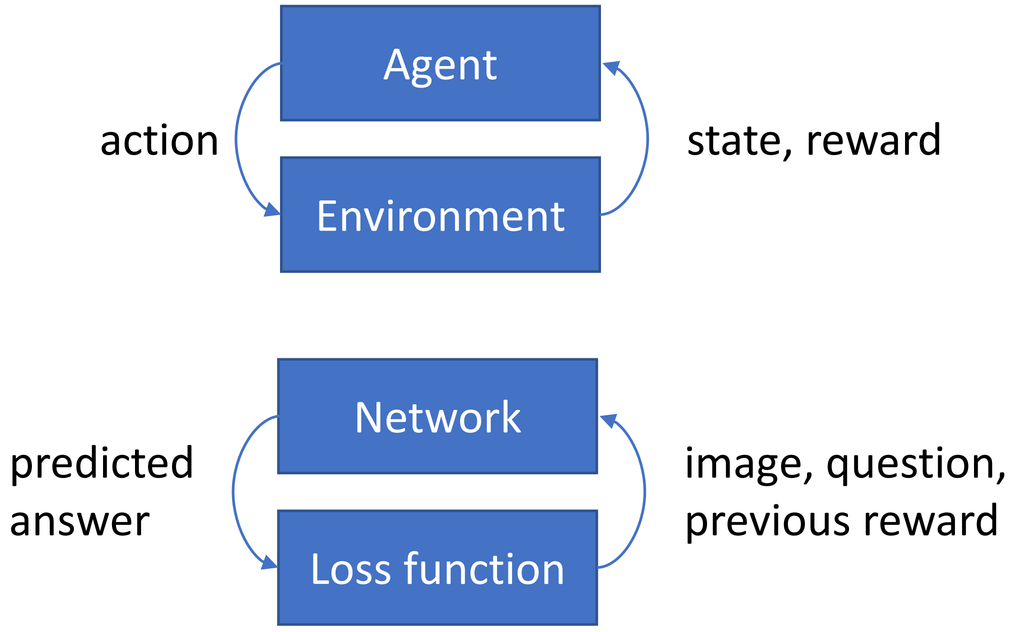
- Supervised
  - Program generator:
    with question and program
    - Standard LSTM training
  - Execution engine (functions):
    with image, program and answer
    - Standard classification training
- Benefits
  - Best performance achievable
- Limitations
  - Ground truth program for all questions needed
  - Not possible if no ground-truth program is available (CLEVER Humans)

Training

- REINFORCE
  - Training program generator and execution engine jointly end to end

- Benefits
  - Needs only images, questions and answers for training, no programs
- Limitations
  - Training without ground truth programs is hard:
    - Generator needs to produce programs without understanding what functions mean
    - Execution engine has to produce the right answer from programs, which may not implement the question correctly
  - Only for fine tuning applicable

REINFORCE



- Reward: Negative zero-one loss of the execution engine
  - 0 if correct, -1 if wrong
- Moving-average baseline
  - Subtracts moving average of rewards
  - Reduces variance of gradient directions
- Correct answering is reinforced

Training

Combined semi-supervised:

1. Train program generator on small subset of ground truth programs
2. Fix program generator and train execution engine using predicted programs on large dataset
3. Use REINFORCE to finetune program generator and execution engine

- Ground truth programs are only used to train program generator in the beginning
- Benefits
  - Possible to finetune on datasets without ground truth programs

Strongly and semi-supervised learning

Strongly supervised

1. Trained program generator and execution engine separately using all ground-truth programs

Semi-supervised

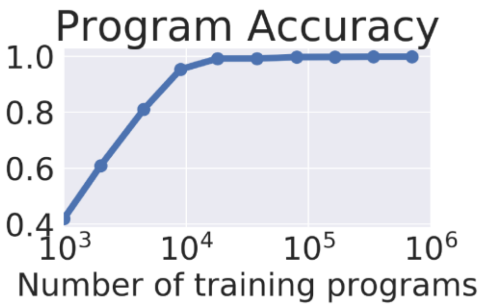
1. Train program generator on small set of ground-truth programs
2. Train execution engine with predicted programs
3. Finetune together without ground-truth programs

Results

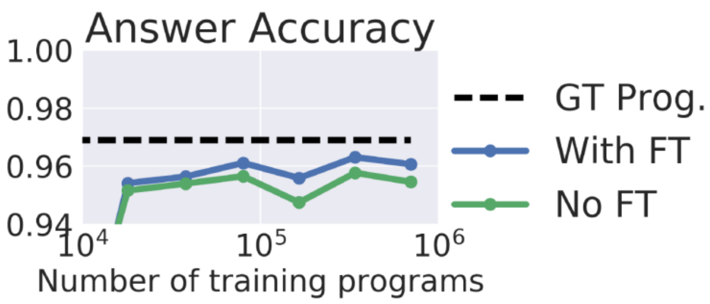
Method	Exist	Count	Compare Integer Equal	Compare Integer Less	Compare Integer More	Query Size	Query Color	Query Mat.	Query Shape	Compare Size	Compare Color	Compare Mat.	Compare Shape	Overall
Q-type mode	50.2	34.6	51.4	51.6	50.5	50.1	13.4	50.8	33.5	50.3	52.5	50.2	51.8	42.1
LSTM	61.8	42.5	63.0	73.2	71.7	49.9	12.2	50.8	33.2	50.5	52.5	49.7	51.8	47.0
CNN+LSTM	68.2	47.8	60.8	74.3	72.5	62.5	22.4	59.9	50.9	56.5	53.0	53.8	55.5	54.3
CNN+LSTM+SA [45]	68.4	57.5	56.8	74.9	68.2	90.1	83.3	89.8	87.6	52.1	55.5	49.7	50.9	69.8
CNN+LSTM+SA+MLP	77.9	59.7	60.3	83.7	76.7	85.4	73.1	84.5	80.7	72.3	71.2	70.1	69.7	73.2
Human† [19]	96.6	86.7	79.0	87.0	91.0	97.0	95.0	94.0	94.0	94.0	98.0	96.0	96.0	92.6
Ours-strong (700K prog.)	**97.1**	**92.7**	**98.0**	**99.0**	**98.9**	**98.8**	**98.4**	**98.1**	**97.3**	**99.8**	**98.5**	**98.9**	**98.4**	**96.9**
Ours-semi (18K prog.)	95.3	90.1	93.9	97.1	97.6	98.1	97.1	97.7	96.6	99.0	97.6	98.0	97.3	95.4
Ours-semi (9K prog.)	89.7	79.7	85.2	76.1	77.9	94.8	93.3	93.1	89.2	97.8	94.5	96.6	95.1	88.6

- Overall accuracy even better than humans on Mechanical Turk
- <4% of Questions sufficient to generalize to 450k unique questions

Results



- 20k ground-truth programs sufficient to have almost exact programs
- 3% better answer accuracy if trained on ground-truth programs
- Finetuning can eliminate some of the error



What do the modules learn?

- Attention is on correct objects
- Changing single module changes answer and module attention drastically

➢Learned meaningful functions

Generalizing to new attribute combinations

Method	Train A A	Train A B	Finetune B A	Finetune B B
LSTM	55.2	50.9	51.5	54.9
CNN+LSTM	63.7	57.0	58.3	61.1
CNN+LSTM+SA+MLP	80.3	68.7	75.7	75.8
Ours (18K prog.)	**96.6**	**73.7**	**76.1**	**92.7**

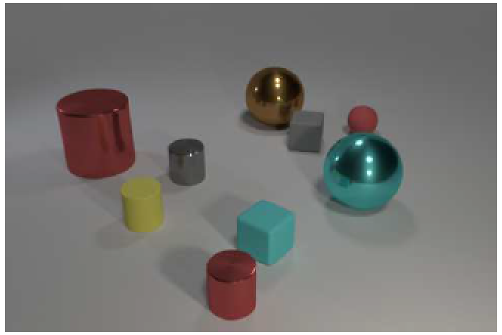
- Split dataset A:
  - Cubes: gray, blue, brown, or yellow
  - Cylinders: red, green, purple, or cyan
- Split B:
  - Colors exchanged

➢ No complete generalization possible if features not in training set

➢ Accuracy on A lost after finetuned on B

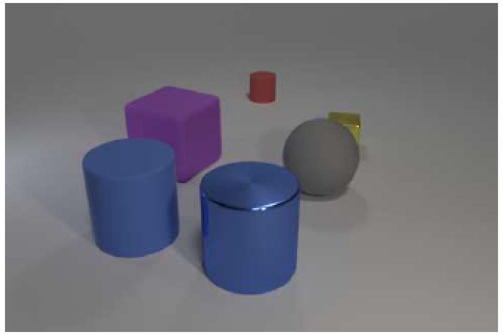
Generalizing to new question types

Ground-truth question:
Is the number of matte blocks in front of the small yellow cylinder greater than the number of red rubber spheres to the left of the large red shiny cylinder?
Program length: 20 **A:** *yes* ✓



Predicted program (translated):
Is the number of matte blocks in front of the small yellow cylinder greater than the number of large red shiny cylinders?
Program length: 15 **A:** *no* ✗

Ground-truth question:
How many objects are big rubber objects that are in front of the big gray thing or large rubber things that are in front of the large rubber sphere?
Program length: 16 **A:** *1* ✓

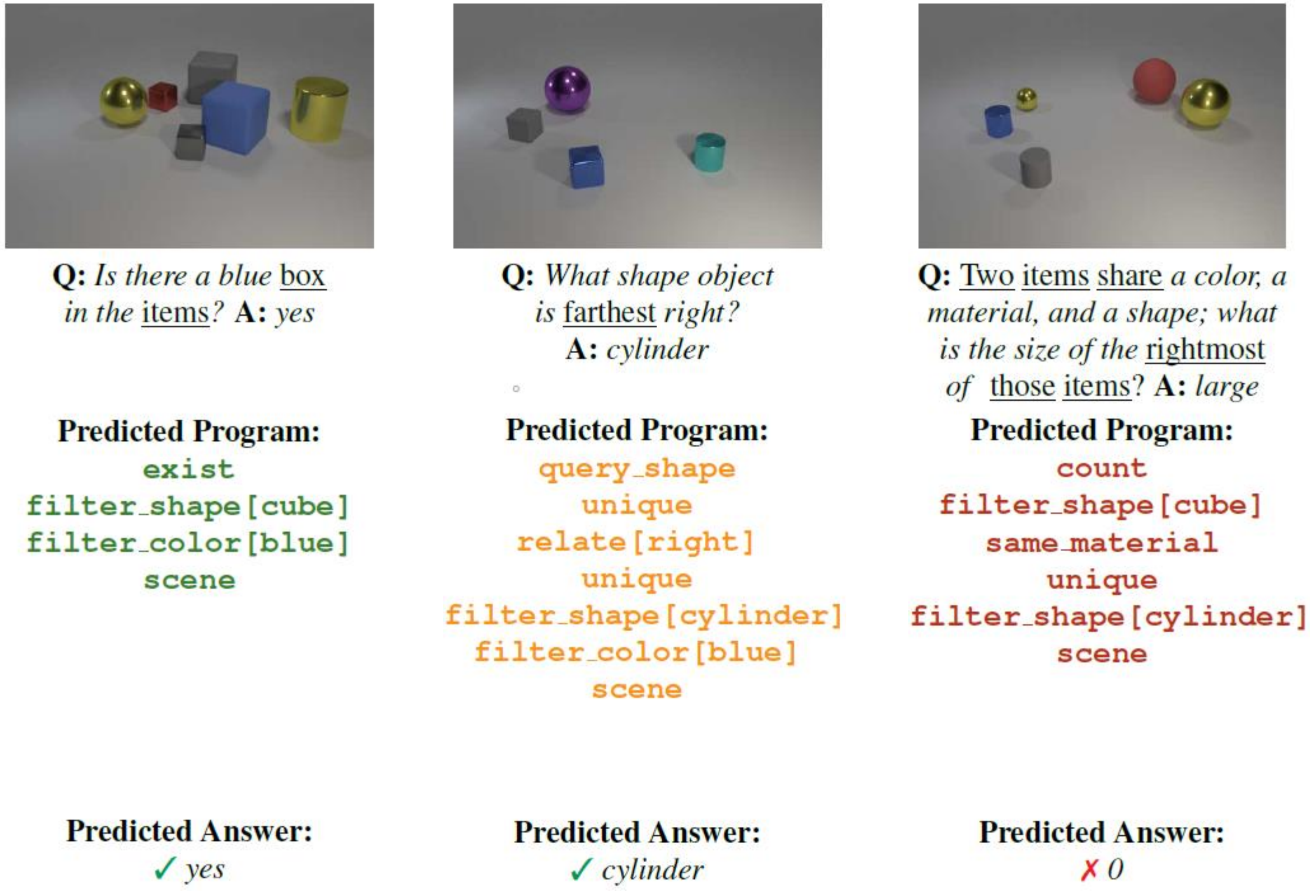


Predicted program (translated):
How many objects are big rubber objects in front of the big gray thing or large rubber spheres?
Program length: 12 **A:** *2* ✗

Method	Train Short		Finetune Both	
	Short	Long	Short	Long
LSTM	46.4	48.6	46.5	49.9
CNN+LSTM	54.0	52.8	54.3	54.2
CNN+LSTM+SA+MLP	74.2	**64.3**	74.2	67.8
Ours (25K prog.)	**95.9**	55.3	**95.6**	**77.8**

- Split long/short questions
- No good performance on long questions if not trained on them
- Generalization possible with finetuning program generator

CLEVER-Humans



- Training on CLEVER
- Random initialization of new word embeddings
- Finetune program generator on CLEVER-Humans
- Answer linguistically more diverse questions
- Reuses reasoning
- Fails if functions are not appropriate to answer question
- Outperforms Baselines

Conclusion

- Increased explainability by step through explainable functions
- Capability to adapt to new question types
- Model exceeds human performance

References

- Johnson et al. 2017; Inferring and Executing Programs for Visual Reasoning
- Johnson et al. 2016; CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning
- Sutskever et al. 2014; Sequence to Sequence Learning with Neural Networks
- He et al. 2015; Deep Residual Learning for Image Recognition
- Zhao et al. 2011; Analysis and Improvement of Policy Gradient Estimation