THE NEURO-SYMBOLIC CONCEPT LEARNER: INTERPRETING SCENES, WORDS, AND SENTENCES FROM NATURAL SUPERVISION

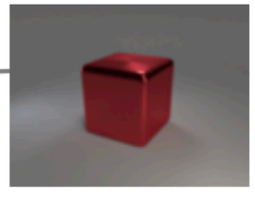
Jiayuan Mao, Chuang Gan, Pushmeet Kohli,
Joshua B. Tenenbaum, Jiajun Wu

ICLR 2019

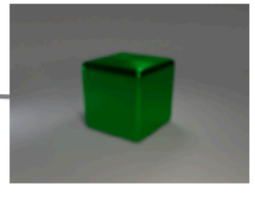
Motivation

- Humans learn visual concepts, words, and semantic parsing jointly and incrementally.
- With no prior knowledge, paired with the questions and answers. People can easily identify the difference in objects' visual appearance, and align it to the corresponding words in the questions and answers.
- Humans are able to inductively learn the correspondence between visual concepts and word semantics, and unravel compositional logic from complex questions assisted by the learned visual concepts

I. Learning basic, object-based concepts.

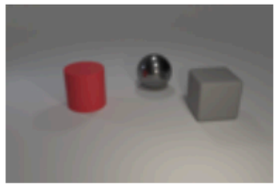


Q: What's the color of the object?
A: Red.
Q: Is there any cube?
A: Yes.



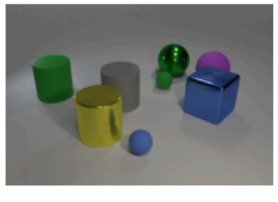
Q: What's the color of the object?
A: Green.
Q: Is there any cube?
A: Yes.

II. Learning relational concepts based on referential expressions.



Q: How many objects are right of the red object?
A: 2.
Q: How many objects have the same material as the cube?
A: 2

III. Interpret complex questions from visual cues.



Q: How many objects are both right of the green cylinder and have the same material as the small blue ball?
A: 3

Idea

- The neuro-symbolic concept learner (NS-CL)

- jointly learns visual perception, words, and semantic language parsing from images and question-answer pairs.

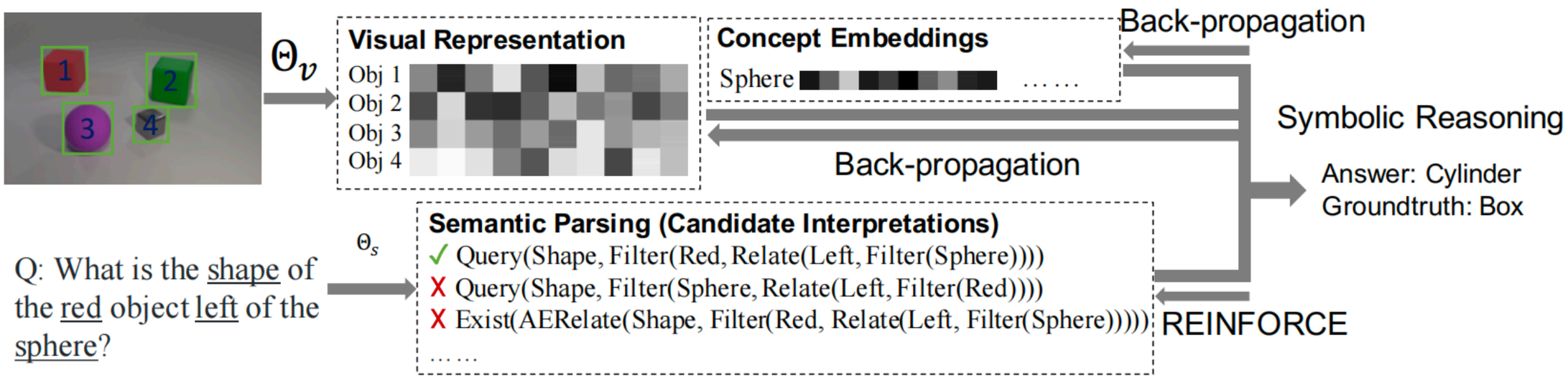
- NS-CL learns from natural supervision (i.e., images and QA pairs), requiring no annotations on images or semantic programs for sentences.

- Analogical to human concept learning, it learns via curriculum learning.

NEURO-SYMBOLIC CONCEPT LEARNER

Overview

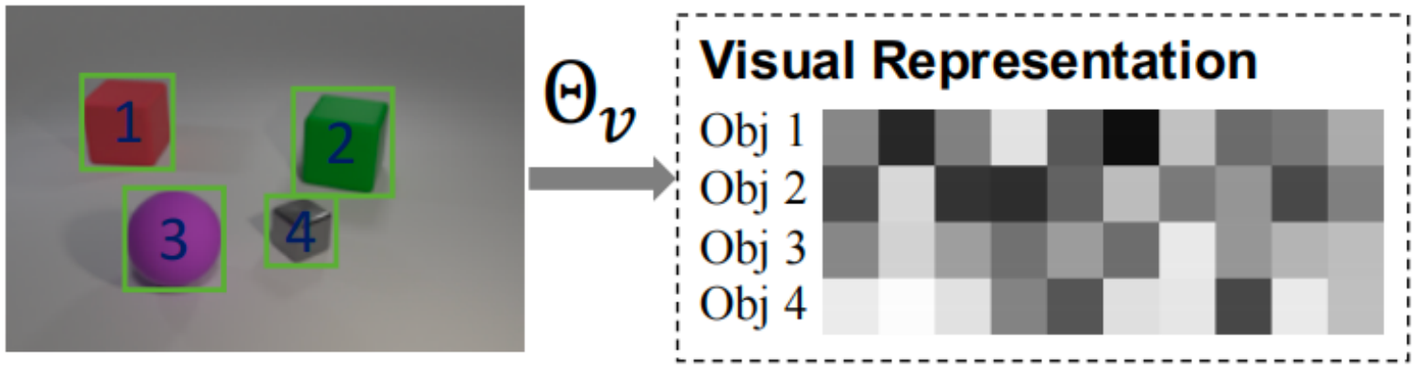
- a neural-based perception module that extracts object-level representations from the scene
- a visually-grounded semantic parser for translating questions into executable programs
- a symbolic program executor that reads out the perceptual representation of objects, classifies their attributes/relations, and executes the program to obtain an answer



NEURO-SYMBOLIC CONCEPT LEARNER

Model detail

- Visual perception:
  - use a pretrained Mask R-CNN to generate object proposals for all objects
  - sent to a ResNet-34 to extract the region-based and image-based features

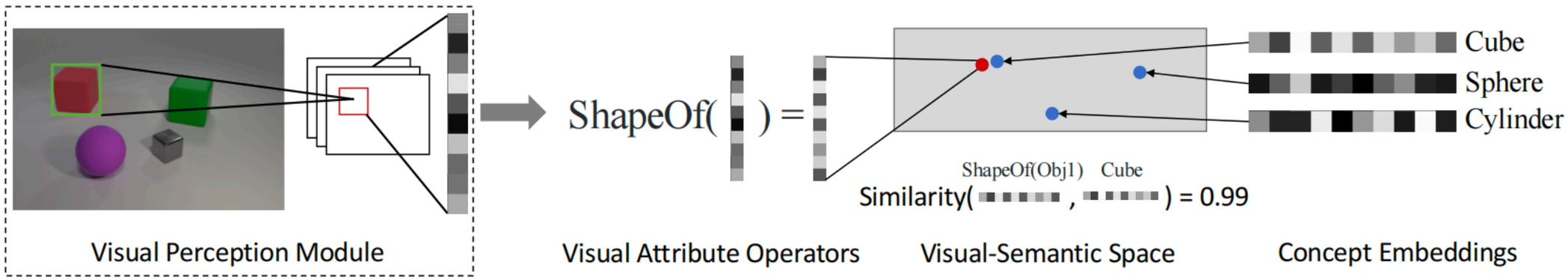


NEURO-SYMBOLIC CONCEPT LEARNER

Model detail

- Concept quantization:
  - assume each visual attribute (e.g., shape) contains a set of visual concept (e.g.,Cube)
  - visual attributes are implemented as neural operators , mapping the object representation into an attribute-specific embedding space
- These concept vectors are also learned along the process

$$\sigma\left(\langle \text{ShapeOf}(o_i), v^{\texttt{Cube}}\rangle - \gamma\right)/\tau$$



NEURO-SYMBOLIC CONCEPT LEARNER

Model detail

- DSL(domain specific language) and semantic parsing:
  - The DSL covers a set of fundamental operations
  - semantic parser generates the hierarchies of latent programs in a sequence to tree manner
  - use a bidirectional GRU to encode an input question
  - a decoder based on GRU cells is applied to the embedding, and recovers the hierarchy of operations as the latent program

Type	Example	Semantics
ObjConcept	`Red`, `Cube`, etc.	Object-level concepts.
Attribute	`Color`, `Shape`, etc.	Object-level attributes.
RelConcept	`Left`, `Front`, etc.	Relational concepts.
Object	●	A single object in the scene.
ObjectSet	{●, ■}	A set of objects in the scene.
Integer	$0, 1, 2, \cdots$	A single integer.
Bool	`True`, `False`	A single boolean value.

Operation	Signature	Semantics
`Scene`	() $\longrightarrow$ ObjectSet	Return all objects in the scene.
`Filter`	(ObjectSet, ObjConcept) $\longrightarrow$ ObjectSet	Filter out a set of objects having the object-level concept (e.g., red) from the input object set.
`Relate`	(Object, RelConcept) $\longrightarrow$ ObjectSet	Filter out a set of objects that have the relational concept (e.g., left) with the input object.
`AERelate`	(Object, Attribute) $\longrightarrow$ ObjectSet	(Attribute-Equality Relate) Filter out a set of objects that have the same attribute value (e.g., same color) as the input object.
`Intersection`	(ObjectSet, ObjectSet) $\longrightarrow$ ObjectSet	Return the intersection of two object sets.
`Union`	(ObjectSet, ObjectSet) $\longrightarrow$ ObjectSet	Return the union of two object sets.
`Query`	(Object, Attribute) $\longrightarrow$ ObjConcept	Query the attribute (e.g., color) of the input object.
`AEQuery`	(Object, Object, Attribute) $\longrightarrow$ Bool	(Attribute-Equality Query) Query if two input objects have the same attribute value (e.g., same color).
`Exist`	(ObjectSet) $\longrightarrow$ Bool	Query if the set is empty.
`Count`	(ObjectSet) $\longrightarrow$ Integer	Query the number of objects in the input set.
`CLessThan`	(ObjectSet, ObjectSet) $\longrightarrow$ Bool	(Counting LessThan) Query if the number of objects in the first input set is less than the one of the second set.
`CGreaterThan`	(ObjectSet, ObjectSet) $\longrightarrow$ Bool	(Counting GreaterThan) Query if the number of objects in the first input set is greater than the one of the second set.
`CEqual`	(ObjectSet, ObjectSet) $\longrightarrow$ Bool	(Counting Equal) Query if the number of objects in the first input set is the same as the one of the second set.

semantic parser

(1) a bidirectional GRU encoder **IEncoder** to encode an input question into a fixed-length embedding

(2) a operation decoder **OpDecoder** that determines the operation tokens(FFB)

(3) a concept decoder **ConceptDecoder** that selects concepts appeared in the input question as the parameters for certain operations (FFN)

(4) a set of output encoders **{OEncoder}** which encode the decoded operations by OpDecoder and output the latent embedding for decoding the next operation(GRU cells)

Algorithm 1: The String-to-Tree Semantic Parser.

Function `parse` $(f, \{c_i\})$ **:**

$program \leftarrow$ `EmptyProgram`$()$;

$program.op \leftarrow$ `OpDecoder`(f);

if $program.op$ requires a concept parameter **then**

$program.concept \leftarrow$ `ConceptDecoder`$(f, \{c_i\})$;

for $i = 0, 1, \cdots$ number of non-concept inputs of $program.op$ **do**

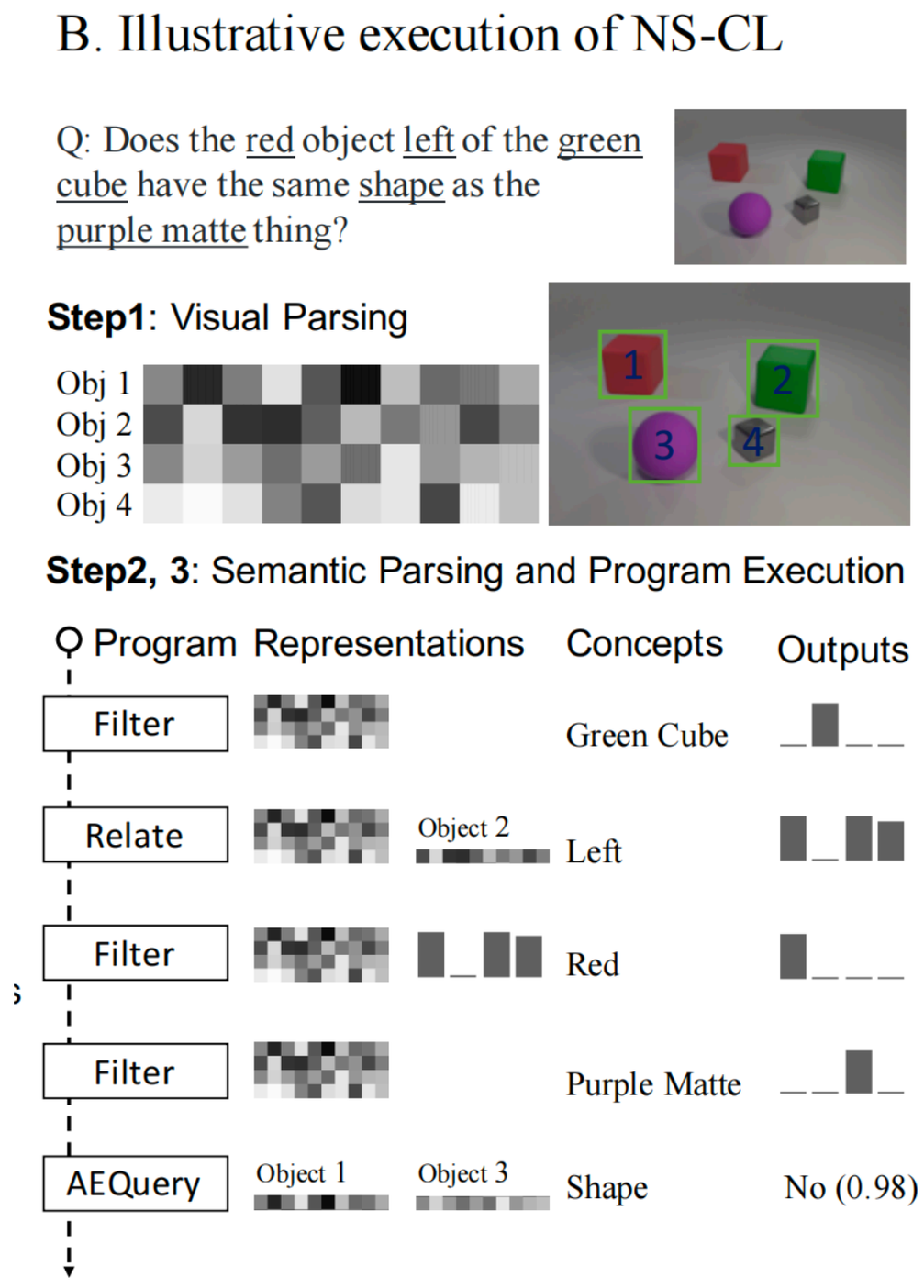
$program.input[i] \leftarrow$ `parse` (`OEncoder`$_i(f, program.op)$,$\{c_i\}$);

return $program$

NEURO-SYMBOLIC CONCEPT LEARNER

Model detail

- Quasi-symbolic program execution:
  - executes the program and derives the answer based on the object-based visual representation
  - program executor is a collection of deterministic functional modules designed to realize all logic operations specified in the DSL



PROGRAM EXECUTION

A Running Example:

"What is the color of the cube right of the red matte object?"

What is the <Attribute 1 (color)> of the <(ObjConcept 1 (cube)> <RelConcept 1 (right)> of the <ObjConcept 2 (red matte object)>?"

```
Query(<Attribute 1>,
      Filter(<ObjConcept 1>,
             Relate(<RelConcept 1>,
                    Filter(<ObjConcept 2>, Scene)
             )
      )
).
```

Step	Inputs	Outputs	Recursive Invocation
1	f_0	$\texttt{OpDecoder}(f_0) \rightarrow \texttt{Query}$; $\texttt{ConceptDecoder}(f_0) \rightarrow <$ Attribute 1 $>$; $\texttt{OEncoder}_0(f_0, \texttt{Query}) \rightarrow f_1$	$parse(f_1)$
2	f_1	$\texttt{OpDecoder}(f_1) \rightarrow \texttt{Filter}$; $\texttt{ConceptDecoder}(f_1) \rightarrow <$ ObjConcept 1 $>$; $\texttt{OEncoder}_0(f_1, \texttt{Filter}) \rightarrow f_2$	$parse(f_2)$
3	f_2	$\texttt{OpDecoder}(f_2) \rightarrow \texttt{Relate}$; $\texttt{ConceptDecoder}(f_2) \rightarrow <$ RelConcept 1 $>$; $\texttt{OEncoder}_0(f_2, \texttt{Relate}) \rightarrow f_3$	$parse(f_3)$
4	f_3	$\texttt{OpDecoder}(f_3) \rightarrow \texttt{Filter}$; $\texttt{ConceptDecoder}(f_3) \rightarrow <$ ObjConcept 2 $>$; $\texttt{OEncoder}_0(f_3, \texttt{Filter}) \rightarrow f_4$	$parse(f_4)$
5	f_4	$\texttt{OpDecoder}(f_3) \rightarrow \texttt{Scene}$;	(End of branch.)

PROGRAM EXECUTION

- Object-typed and ObjectSet-typed variables

  - consider a scene with **n** objects, an Object-typed(ObjectSet-typed) variable can be represented as a vector Object(ObjectSet) of length **n**, where Object_i ∈ [0, 1] (Objectset_i ∈ [0, 1])and $\sum_i \mathrm{Object}_i = 1$ $\mathrm{ObjectSet}_i \in [0, 1]$
  - If want to cast objects to object: $\mathrm{Object} = \mathrm{softmax}(\sigma^{-1}(\mathrm{ObjectSet}))$.

PROGRAM EXECUTION

- Concept quantization
  - Denote ***o_i*** as the visual representation of the i-th object, ***OC*** the set of all object-level concepts (red..), and ***A*** the set of all object-level attributes (color..)
  - Each object-level concept ***oc*** is associated with a vector embedding ***v^oc*** and a L1-normalized vector ***b^oc*** of length |A|. ***b^oc*** represents which attribute does this object-level concept belong to.
  - All attributes a ∈ A are implemented as neural operators, denoted as ***u^a***
  - 例如： *To classify the objects as being Red or not*

$$\Pr[\text{object } i \text{ is } \texttt{Red}] = \sigma\left(\sum_{a\in A}\left(b_a^{\texttt{Red}} \cdot \frac{\langle u^a(o_i), v_{\texttt{Red}}\rangle - \gamma}{\tau}\right)\right)$$

会得到一个长度为*n*的代表 *ObjClassify(Red)* 的向量表示，如果计算关系概念，则得到一个*NxN*的矩阵

例如： *To classify whether two objects have the same attribute*

$$\Pr[\text{object } i \text{ has the same } \texttt{Color} \text{ as object } j] = \sigma\left(\frac{\langle u^{\texttt{Color}}(o_i), u^{\texttt{Color}}(o_j)\rangle - \gamma}{\tau}\right)$$

会得到一个*NxN*的代表 *AEClassify(Color)* 的矩阵

PROGRAM EXECUTION

Signature	Implementation
`Scene`() → *out*: ObjectSet	$out_i := 1$
`Filter`(*in*: ObjectSet, *oc*: ObjConcept) → *out*: ObjectSet	$out_i := \min(in_i, \text{ObjClassify}(oc)_i)$
`Relate`(*in*: Object, *rc*: RelConcept) → *out*: ObjectSet	$out_i := \sum_j (in_j \cdot \text{RelClassify}(rc)_{j,i}))$
`AERelate`(*in*: Object, *a*: Attribute) → *out*: ObjectSet	$out_i := \sum_j (in_j \cdot \text{AEClassify}(a)_{j,i}))$
`Intersection`($in^{(1)}$: ObjectSet, $in^{(2)}$: ObjectSet) → *out*: ObjectSet	$out_i := \min(in_i^{(1)}, in_i^{(2)})$
`Union`($in^{(1)}$: ObjectSet, $in^{(2)}$: ObjectSet) → *out*: ObjectSet	$out_i := \max(in_i^{(1)}, in_i^{(2)})$
`Query`(*in*: Object, *a*: Attribute) → *out*: ObjConcept	$\Pr[out = oc] := \sum_i in_i \cdot \frac{\text{ObjClassify}(oc)_i \cdot b_a^{oc}}{\sum_{oc'} \text{ObjClassify}(oc')_i \cdot b_a^{oc'}}$
`AEQuery`($in^{(1)}$: Object, $in^{(2)}$: Object, *a*: Attribute) → *b*: Bool	$b := \sum_i \sum_j (in_i^{(1)} \cdot in_j^{(2)} \cdot \text{AEClassify}(a)_{j,i}))$
`Exist`(*in*: ObjectSet) → *b*: Bool	$b := \max_i in_i$
`Count`(*in*: ObjectSet) → *i*: Integer	$i := \sum_i in_i$
`CLessThan`($in^{(1)}$: ObjectSet, $in^{(2)}$: ObjectSet) → *b*: Bool	$b := \sigma\big((\sum_i in_i^{(2)} - \sum_i in_i^{(1)} - 1 + \gamma_c)/\tau_c\big)$
`CGreaterThan`($in^{(1)}$: ObjectSet, $in^{(2)}$: ObjectSet) → *b*: Bool	$b := \sigma\big((\sum_i in_i^{(1)} - \sum_i in_i^{(2)} - 1 + \gamma_c)/\tau_c\big)$
`CEqual`($in^{(1)}$: ObjectSet, $in^{(2)}$: ObjectSet) → *b*: Bool	$b := \sigma\big((-\lvert \sum_i in_i^{(1)} - \sum_i in_i^{(2)} \rvert + \gamma_c)/(\gamma_c \cdot \tau_c)\big)$

Table 9: All operations in the domain-specific language for CLEVR VQA. $\gamma_c = 0.5$ and $\tau_c = 0.25$ are constants for scaling and shift the probability. During inference, one can quantify all operations as Yi et al. (2018).

NEURO-SYMBOLIC CONCEPT LEARNER

Training Paradigm

- Optimization objective:
  - concept learning (perception module Perception) and language understanding (semantic parsing module SemanticParse)
  - Goal is to maximize the likelihood of answering the question Q correctly

$$\Theta_v, \Theta_s \leftarrow \arg\max_{\Theta_v, \Theta_s} \mathbb{E}_P[\Pr[A = \text{Executor}(\text{Perception}(S; \Theta_v), P)]],$$

P denotes the program, A the answer, S the scene, and Executor the quasi-symbolic executor

- compute the gradient

Θ_v as $\nabla_{\Theta_v} \mathbb{E}_P[D_{\text{KL}}(\text{Executor}(\text{Perception}(S; \Theta_v), P) \| A)]$

$$\Theta_s\text{:}\ \ \nabla_{\Theta_s} = \mathbb{E}_P[r \cdot \log \Pr[P = \text{SemanticParse}(Q; \Theta_s)]]$$

$$\Theta_s\colon\ \nabla_{\Theta_s} = \mathbb{E}_P[r \cdot \log \Pr[P = \text{SemanticParse}(Q; \Theta_s)]]$$

- apply an off-policy program search process
  - approximate this gradient via Monte Carlo sampling
  - exactly compute the gradient (In the third stage of the curriculum learning, search for the set Q offline based on the quantified results of concept classification and compute the exact gradient)
  - [直观解释是，枚举所有可能的程序，执行它们，并找到那些导致正确答案的程序。 使用 Q(s) 作为问题的“groundtruth”程序注释，以监督学习，而不是运行基于 Monte Carlo 采样的 REINFORCE]

- Problem: $\ell = \sum_{p \in \mathbb{Q}(S)} - \log \Pr(p)$
  - intrinsic ambiguity: two programs are different but equivalent.
  - extrinsic spuriousness: one of the program is incorrect, but also leads to the correct answer

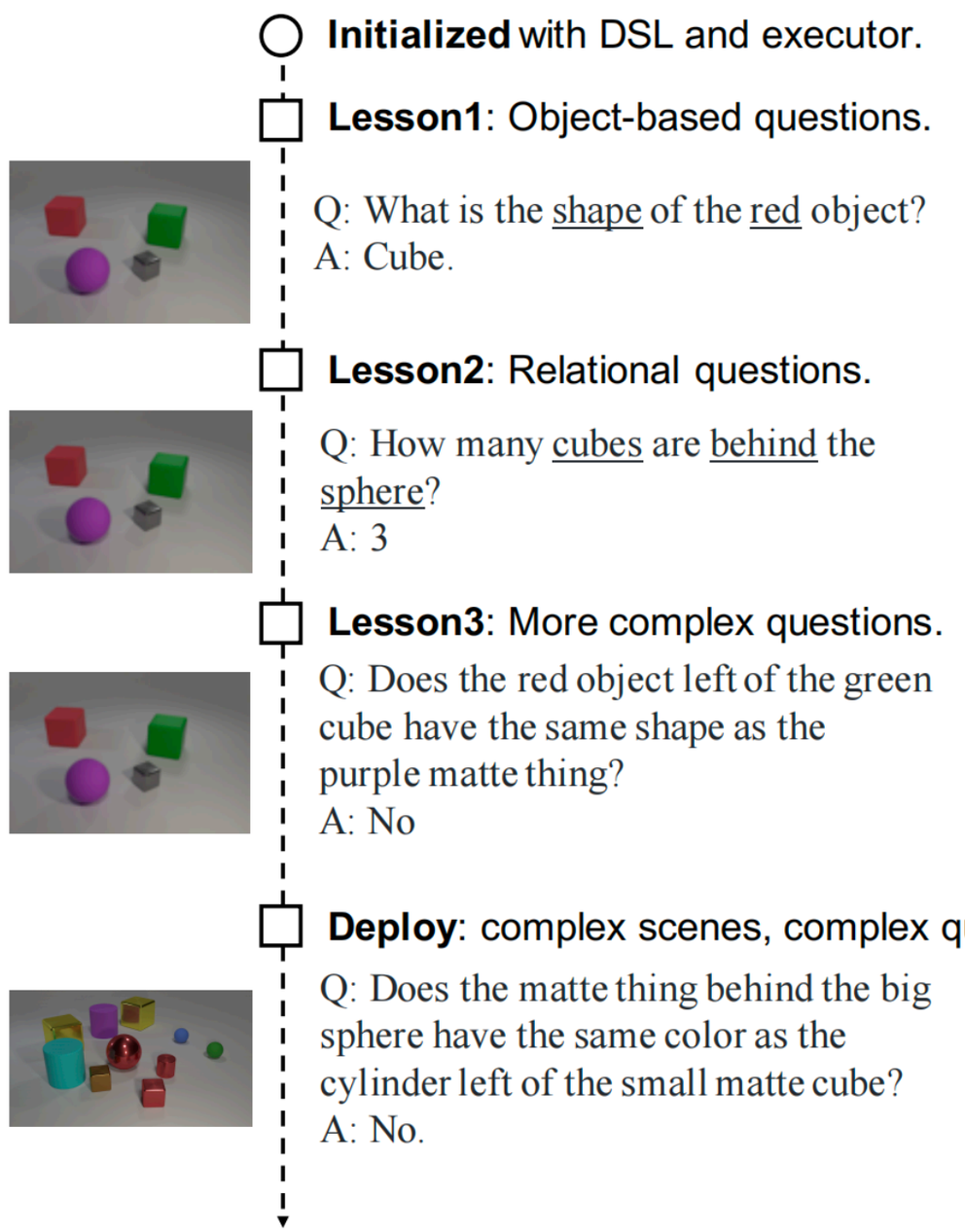
$$\ell = \sum_{p \in \mathbb{Q}} \text{stop\_gradient}(\Pr[p]) \cdot (-\log \Pr[p])$$

$$\nabla_{\Theta_s} = \sum_{p \in \mathbb{Q}} \Pr[p] \cdot \nabla_{\Theta_s} (r \cdot \log \Pr[P]) = \nabla_{\Theta_s} \left(\sum_{p \in \mathbb{Q}} r \cdot \Pr[p] \right)$$

NEURO-SYMBOLIC CONCEPT LEARNER

Training Paradigm

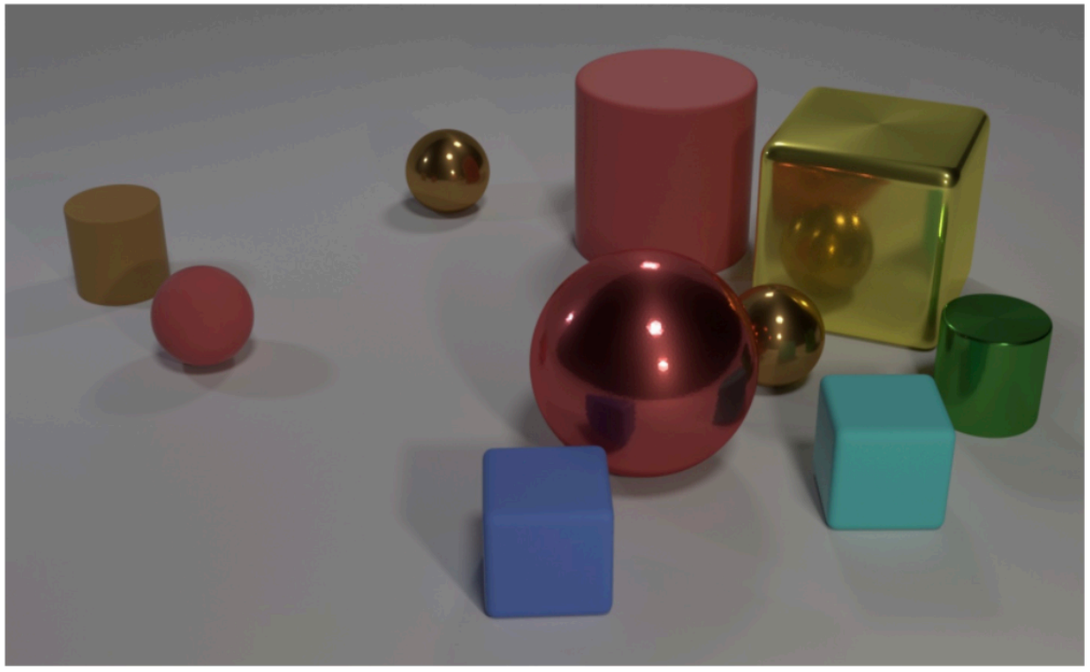
- Curriculum visual concept learning:
  - split the training samples into four stages:
    - first, learning object-level visual concepts;
    - second, learning relational questions;
    - third, learning more complex questions with perception modules fixed;
    - fourth, joint fine-tuning of all modules.



EXPERIMENTS

Datasets

- CLEVR
  - 由一些简单的几何形状组成的视觉图像。数据集中的问题多涉及到复杂推理，问题类别包括：属性查询（querying attribute），属性比较（comparing attributes），存在（existence），计数（counting），整数比较（integer comparison）等



Q: Are there an equal number of large things and metal spheres?
Q: What size is the cylinder that is left of the brown metal thing that is left of the big sphere? **Q:** There is a sphere with the same size as the metal cube; is it made of the same material as the small red sphere?
Q: How many objects are either small cylinders or metal things?

EXPERIMENTS

VISUAL CONCEPT LEARNING

- Classification-based concept evaluation
  - achieve near perfect classification accuracy (~99%) for all object properties on CLEVR
  - The result for spatial relations is relatively lower, it can only be learned indirectly
- Count-based concept evaluation
  - SOTA methods do not provide interpretable representation on individual objects
  - generate a synthetic question set (How many red objects are there?)

	Visual	Mean	Color	Mat.	Shape	Size
IEP	Conv.	90.6	91.0	90.0	89.9	90.6
MAC	Attn.	95.9	98.0	91.4	94.4	94.2
TbD (hres.)	Attn.	96.5	96.6	92.2	95.4	92.6
NS-CL	Obj.	**98.7**	**99.0**	**98.7**	**98.1**	**99.1**

EXPERIMENTS

DATA-EFFICIENT AND INTERPRETABLE VISUAL REASONING

Model	Prog. Anno.	Overall	Count	Cmp. Num.	Exist	Query Attr.	Cmp. Attr.
Human	N/A	92.6	86.7	86.4	96.6	95.0	96.0
NMN	700K	72.1	52.5	72.7	79.3	79.0	78.0
N2NMN	700K	88.8	68.5	84.9	85.7	90.0	88.8
IEP	700K	96.9	92.7	98.7	97.1	98.1	98.9
DDRprog	700K	98.3	96.5	98.4	98.8	99.1	99.0
TbD	700K	**99.1**	**97.6**	**99.4**	**99.2**	**99.5**	**99.6**
RN	0	95.5	90.1	93.6	97.8	97.1	97.9
FiLM	0	97.6	94.5	93.8	99.2	99.2	99.0
MAC	0	98.9	97.2	**99.4**	**99.5**	99.3	**99.5**
NS-CL	0	**98.9**	**98.2**	99.0	98.8	**99.3**	99.1

Table 4: Our model outperforms all baselines using no program annotations. It achieves comparable results with models trained by full program annotations such as TbD.

Model	Visual	Accuracy (100% Data)	Accuracy (10% Data)
TbD	Attn.	99.1	54.2
TbD-Object	Obj.	84.1	52.6
TbD-Mask	Attn.	99.0	55.0
MAC	Attn.	98.9	67.3
MAC-Object	Obj.	79.5	51.2
MAC-Mask	Attn.	98.7	68.4
NS-CL	Obj.	**99.2**	**98.9**

Table 3: We compare different variants of baselines for a systematic study on visual features and data efficiency. Using only 10% of the training images, our model is able to achieve a comparable results with the baselines trained on the full dataset. See the text for details.

EXPERIMENTS

GENERALIZATION TO NEW ATTRIBUTES AND COMPOSITIONS

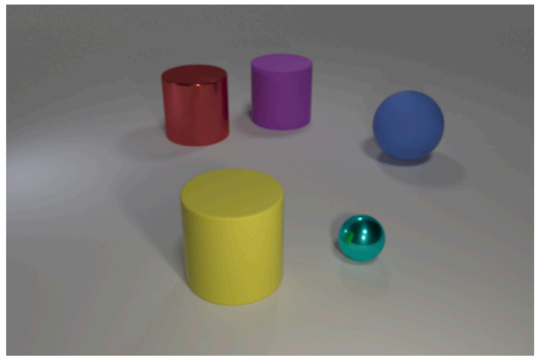
- Generalizing to new visual compositions:
  - CLEVR-CoGenT dataset: SplitA and SplitB has the opposite color on the same shape
  - achieves an accuracy of 98.8% on split A and 98.9% on split B
- Generalizing to new visual concepts:
  - build a synthetic split of the CLEVR dataset: SplitA has no purple objects and SplitB has at least 1
  - performs a 93.9% accuracy on the QA test in Split B

EXPERIMENTS

COMBINATORIAL GENERALIZATION TO NEW SCENES AND QUESTIONS

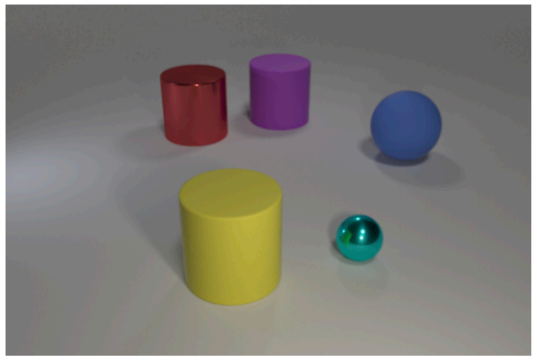
- Split A contains only scenes with less than 6 objects;
- Split B contains scenes with less than 6 objects, but arbitrary questions;
- Split C contains arbitrary scenes, but restricts the program depth being less than 5;
- Split D contains arbitrary scenes and questions.

Split A



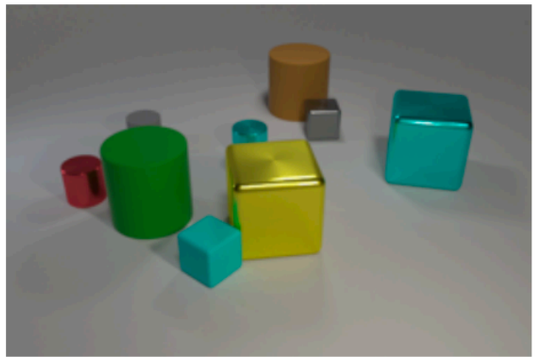
Q: What's the shape of the big yellow thing?

Split B



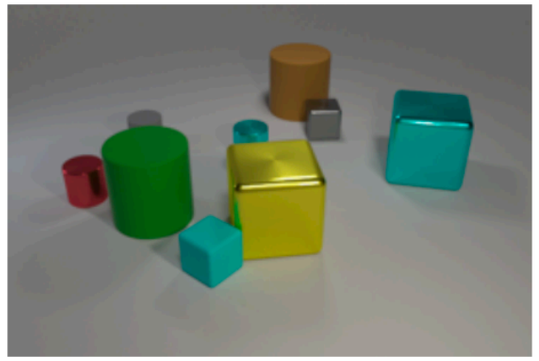
Q: What size is the cylinder that is left of the cyan thing that is in front of the big sphere?

Split C



Q: What's the shape of the big yellow thing?

Split D



Q: What size is the cylinder that is left of the cyan thing that is in front of the gray cube?

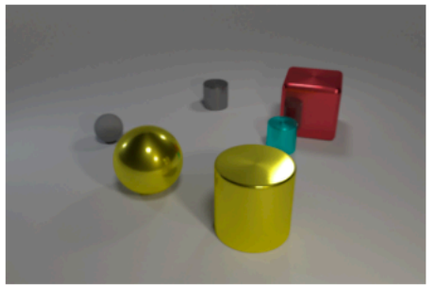
Model	Test			
	Split A	Split B	Split C	Split D
MAC	97.3	N/A	92.9	N/A
IEP	96.1	92.1	91.5	90.9
TbD	98.8	94.5	94.3	91.9
NS-CL	**98.9**	**98.9**	**98.7**	**98.8**

EXPERIMENTS

EXTENDING TO OTHER PROGRAM DOMAIN

- The learned visual concepts can also be used in other domains such as image retrieval
- A separate semantic parser is trained for the VQA baselines, which translates captions into a CLEVR QA-compatible program

```
Exist(Filter(Box, Relate(Right, Filter(Cylinder))))
```



Caption: There is a big yellow cylinder in front of a gray object.

(a) An illustrative pair of image and caption in our synthetic dataset.

Model	Retrieval Accuracy
IEP	95.5
TbD	**97.0**
NS-CL	96.9

(b) Image-caption retrieval accuracy on a subset of data. Our model archives comparable results with VQA baselines.

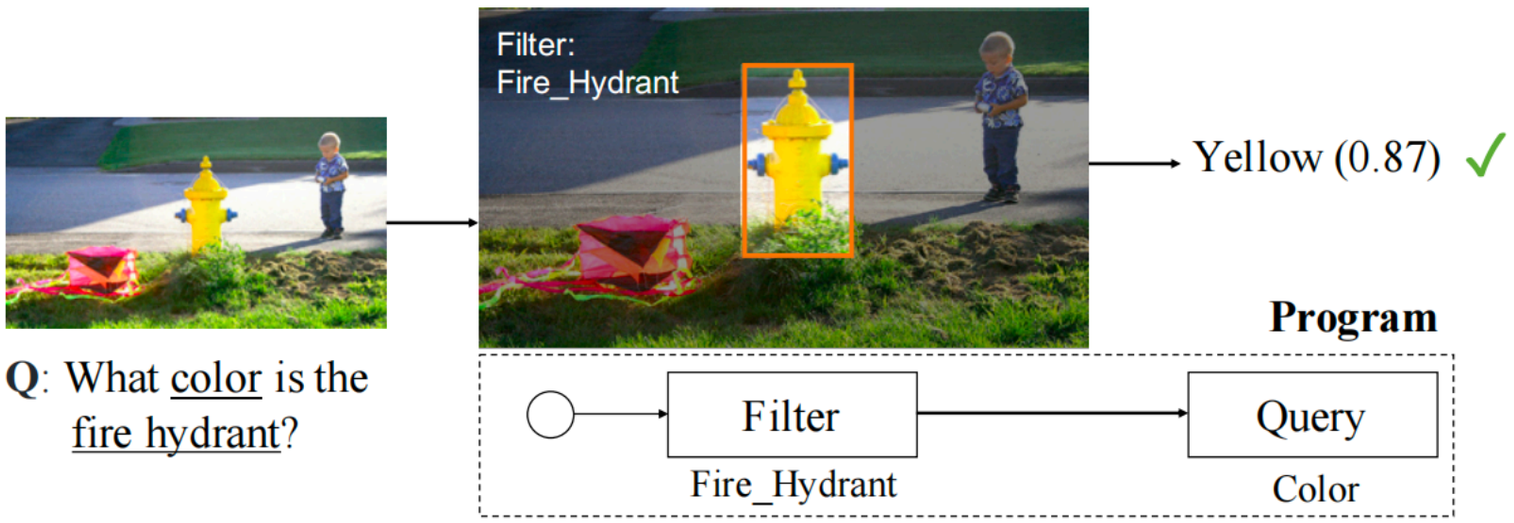
Model	Retrieval Accuracy
CNN-LSTM	68.9
NS-CL	**97.0**

(c) Image-caption retrieval accuracy on the full dataset. Our model outperforms baselines and requires no extra training or fine-tuning of the visual perception module.

EXPERIMENTS

EXTENDING TO NATURAL IMAGES AND LANGUAGE

- Results are presented on the VQS dataset



Model	Accuracy
MLP	43.9
MAC	**46.2**
NS-CL	44.3

ABLATION STUDY

- Semantic parsing accuracy: archives > 99.9% QA accuracy on the validation split
- Impacts of the ImageNet Pre-Training: almost remain the same except for Shape drops from 98.7 to 97.5
- Data efficiency and object-based representations:

Model	Visual	Accuracy (100% Data)	Accuracy (10% Data)
TbD	Attn.	99.1	54.2
TbD-Object	Obj.	84.1	52.6
TbD-Mask	Attn.	99.0	55.0
MAC	Attn.	98.9	67.3
MAC-Object	Obj.	79.5	51.2
MAC-Mask	Attn.	98.7	68.4
NS-CL	Obj.	**99.2**	**98.9**

Table 3: We compare different variants of baselines for a systematic study on visual features and data efficiency. Using only 10% of the training images, our model is able to achieve a comparable results with the baselines trained on the full dataset. See the text for details.

Extending to other scene and language domains

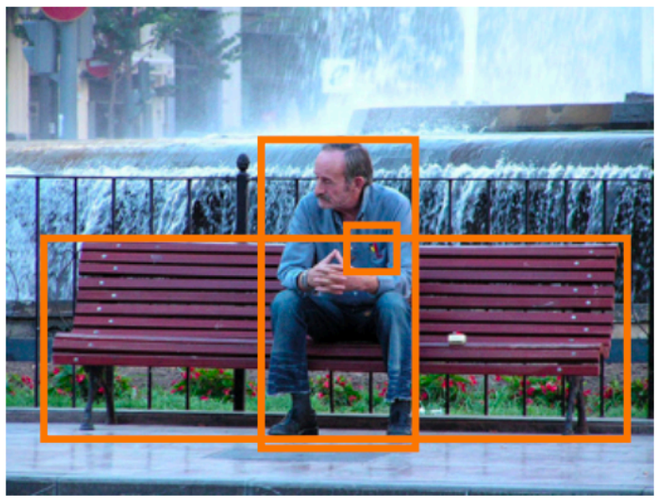
- Minecraft DATASET:

Q: What direction is the closest creature facing?
A: Left.
P: Query(Direction, FilterMost(Closest,
Filter(Creature)
))

Model	Overall	Count	Exist	Belong	Query
NS-VQA	87.7	83.3	91.5	91.1	86.4
NS-CL	**93.3**	**91.3**	**95.6**	**93.9**	**94.3**

Extending to other scene and language domains

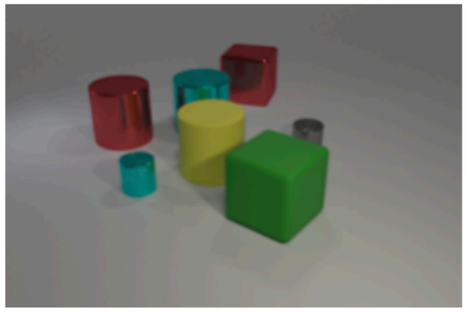
- VQS DATASET:



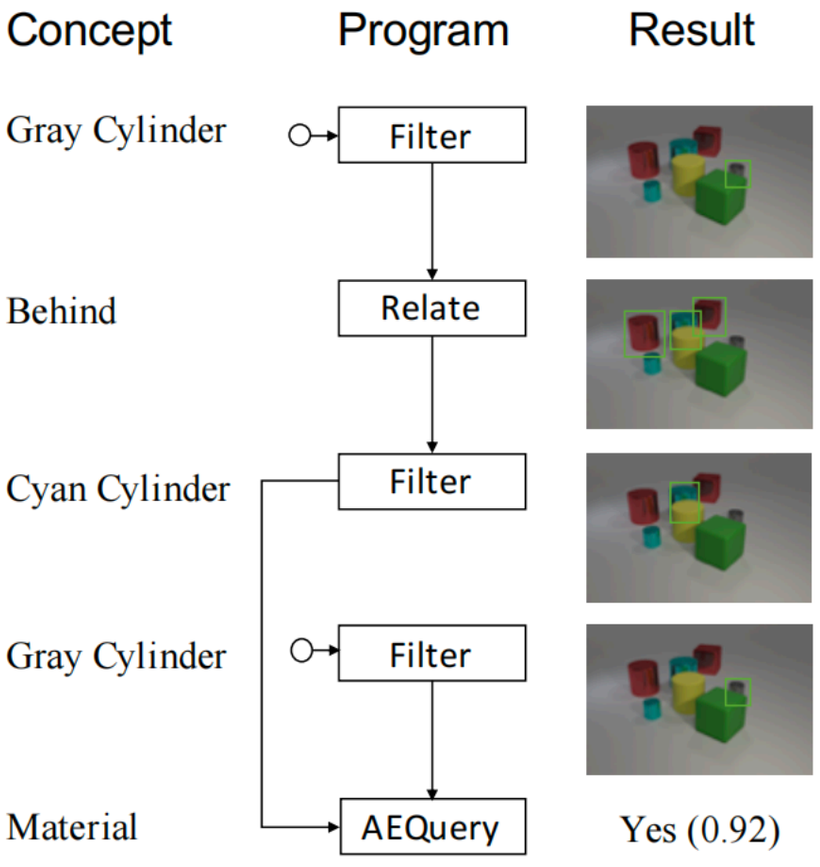
Q: Does this man have any pens on him?
A: Yes.
P: Exist(Filter(Man,
Relate(Have, Filter(Pen))
))

Model	Accuracy
MLP	43.9
MAC	**46.2**
NS-CL	44.3

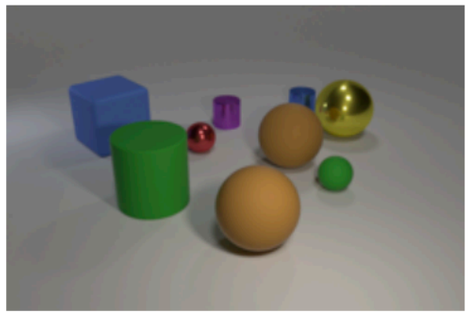
Example A.



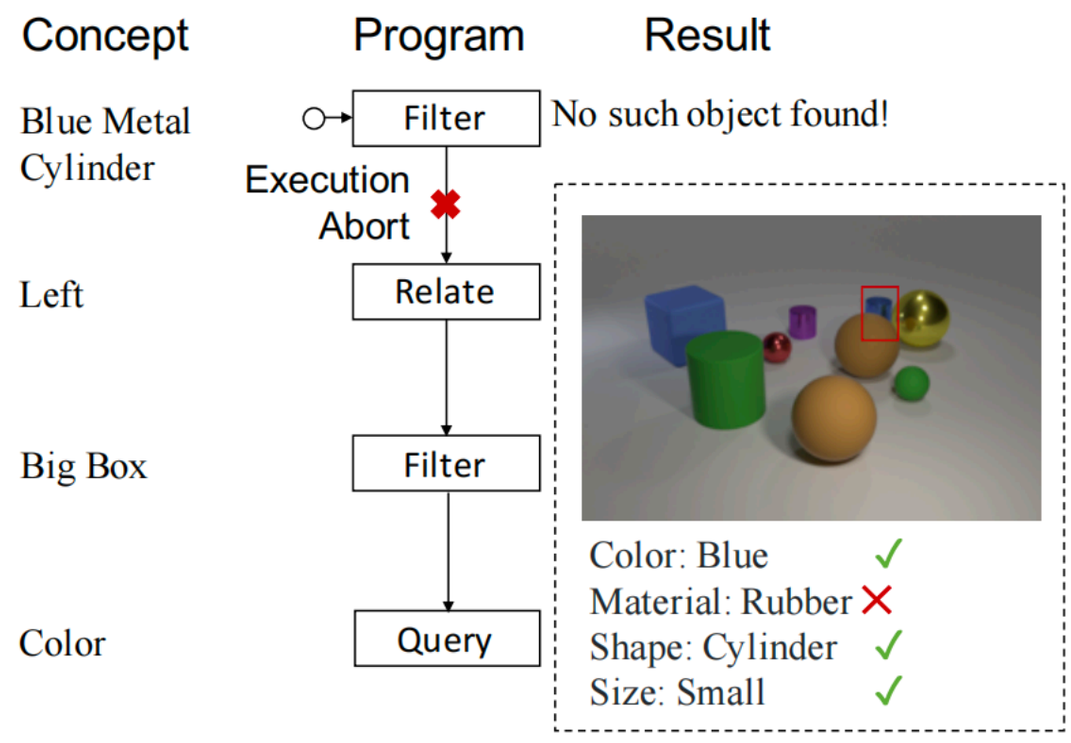
Q: Do the cyan cylinder that is behind the gray cylinder and the gray cylinder have the same material?



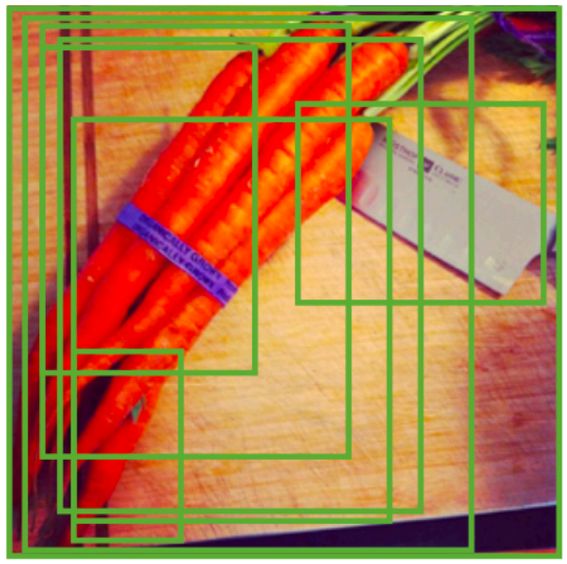
Example C. Failure Case



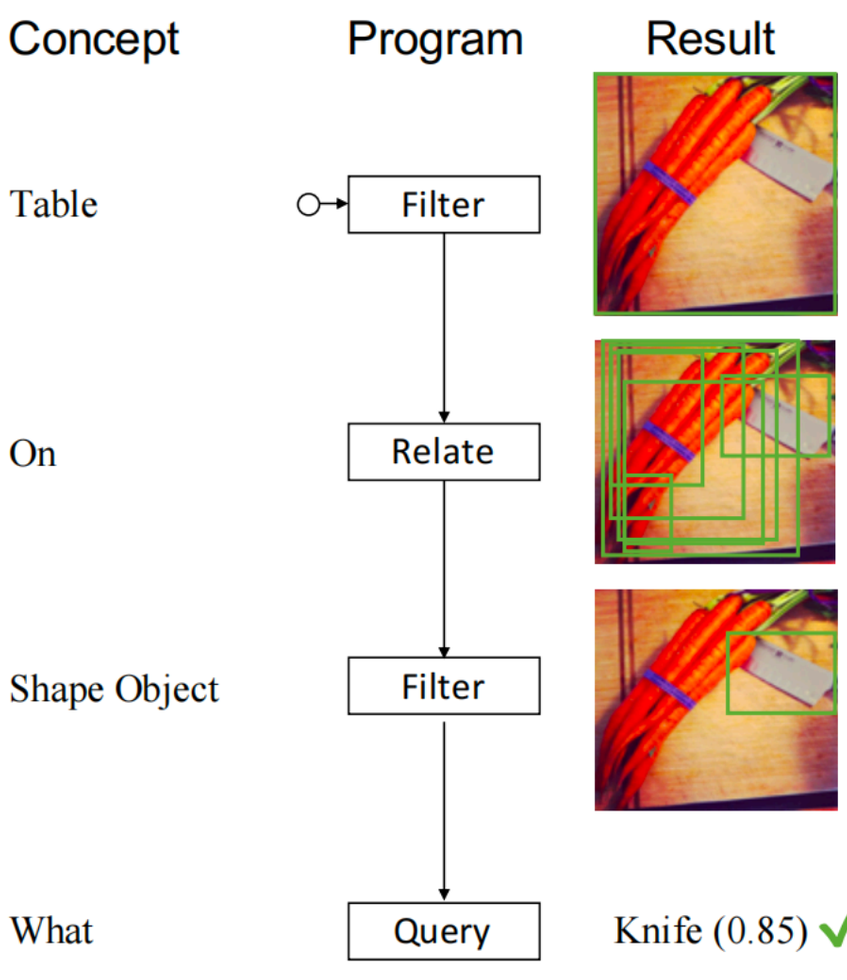
Q: What is the color of the big box left of the blue metal cylinder?



Example B.

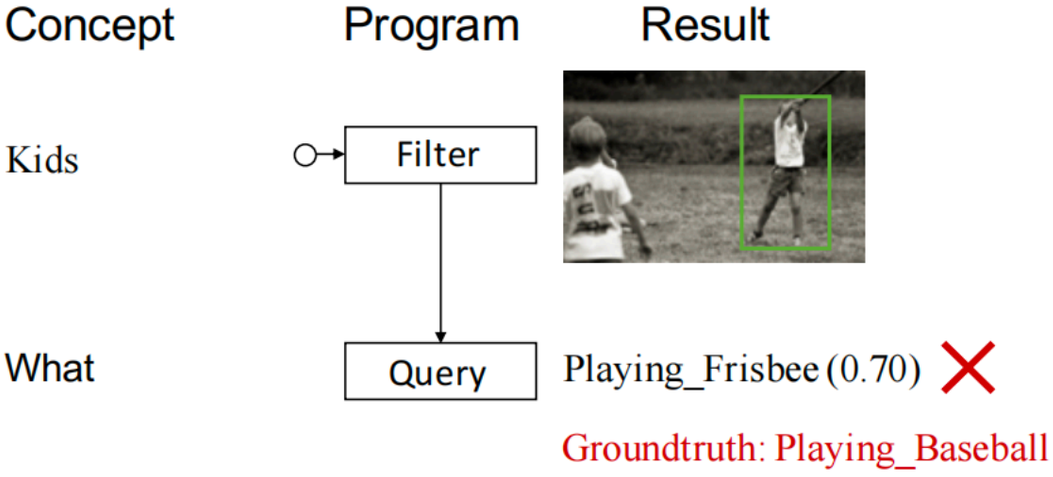


Q: What is the sharp object on the table?



Example D.

Q: What are the kids doing?



Thank You