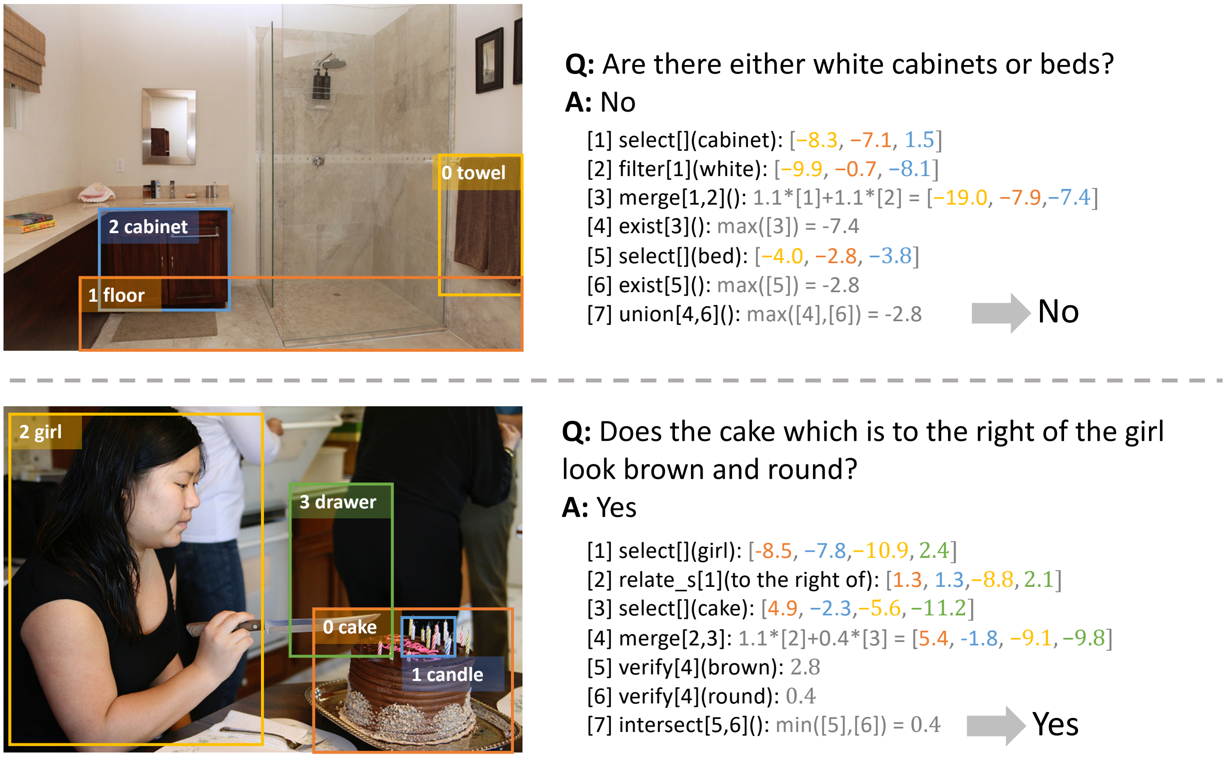
Calibrating Concepts and Operations:
Towards Symbolic Reasoning on Real Images
Supplementary Materials

1. Execution Examples



Q: Are there either white cabinets or beds?
A: No

```
[1] select[](cabinet): [−8.3, −7.1, 1.5]
[2] filter[1](white): [−9.9, −0.7, −8.1]
[3] merge[1,2](): 1.1*[1]+1.1*[2] = [−19.0, −7.9,−7.4]
[4] exist[3](): max([3]) = -7.4
[5] select[](bed): [−4.0, −2.8, −3.8]
[6] exist[5](): max([5]) = -2.8
[7] union[4,6](): max([4],[6]) = -2.8        No
```

Q: Does the cake which is to the right of the girl look brown and round?
A: Yes

```
[1] select[](girl): [-8.5, −7.8,−10.9, 2.4]
[2] relate_s[1](to the right of): [1.3, 1.3,−8.8, 2.1]
[3] select[](cake): [4.9, −2.3,−5.6, −11.2]
[4] merge[2,3]: 1.1*[2]+0.4*[3] = [5.4, -1.8, −9.1, −9.8]
[5] verify[4](brown): 2.8
[6] verify[4](round): 0.4
[7] intersect[5,6](): min([5],[6]) = 0.4        Yes
```

Figure 1: Examples of the execution process. Best view in color.

In Figure 1, two examples are shown to help better understand the reasoning process. For simplification, in each example, only a few representative object region proposal boxes are shown. The output of each execution step (shown in different colors corresponding to the image region colors) are the scores representing each region box being selected or not. By looking at the intermediate outputs, we can see how each execution step changes the selection scores. For example, in example (a), the $select(cabinet)$ step produces positive scores for the cabinet object region and negative scores for the others, but after $filter(white)$ and the $merge$ step, all scores become negative, which indicates that there is not white cabinet in this image. At last, the output modules does logistic operation on top of the final selection scores to produce answers. For example, $exist$ checks whether there is a positive score to find if there is some objects being selected.

2. Module Details

While Section 4.2 shows computations for $select$ and $query$ modules, here in Table 1, we show details of all modules, including their inputs, output, arguments, description and computation. In the inputs and output, bold symbols (*e.g.*, $\mathbf{d}, \mathbf{a}$)

represent vectors while plain symbols (*e.g.*, a) represent scalar scores. [1]

Table 1: List of all modules. Inputs, output, arguments, description and computation details are shown for each module.

Module	**Inputs**	**Output**	**Arguments**	**Description**	**Computation**
Intermediate Modules					
select	-	$\mathbf{d}^{out}$	$concept, attr$	find object named $concept$	$\mathbf{e}_i = \mathcal{M}_{attr}(\mathbf{v}_i)$, $\mathbf{d}_i^{out} = \text{sim}(\mathbf{e}_i, \mathbf{c}_{concept})$
filter	$\mathbf{d}^{in}$	$\mathbf{d}^{out}$	$concept, attr$	from the input $\mathbf{d}^{in}$, find object whose attribute $attr$ is $concept$	$\mathbf{e}_i = \mathcal{M}_{attr}(\mathbf{v}_i)$, $\mathbf{d}_i^{res} = \text{sim}(\mathbf{e}_i, \mathbf{c}_{concept})$, $\mathbf{d}^{out} = \text{Merge}(\mathbf{d}^{in}, \mathbf{d}^{res})$
relate_o	$\mathbf{d}_1^{in}, \mathbf{d}_2^{in}$	$\mathbf{d}^{out}$	$rel, rtype$ [2]	find the object of rel to $\mathbf{d}_2^{in}$ from $\mathbf{d}_1^{in}$	$\mathbf{e}_{ij} = \mathcal{M}_{rtype}^{r}(\mathbf{v}_i, \mathbf{v}_j)$, $\mathbf{mask}_{ij} = \text{sim}(\mathbf{e}_{ij}, \mathbf{c}_{rel})$, $\mathbf{d}_i^{res} = \sum_{j=1}^{N} \mathbf{d}_{2j}^{in} \mathbf{mask}_{ij}$, $\mathbf{d}^{out} = \text{Merge}(\mathbf{d}_1^{in}, \mathbf{d}^{res})$
relate_s	$\mathbf{d}_1^{in}, \mathbf{d}_2^{in}$	$\mathbf{d}^{out}$	$rel, rtype$	find the object of rel to $\mathbf{d}_2^{in}$ from $\mathbf{d}_1^{in}$	$\mathbf{e}_{ij} = \mathcal{M}_{rtype}^{r}(\mathbf{v}_j, \mathbf{v}_i)$, $\mathbf{mask}_{ij} = \text{sim}(\mathbf{e}_{ij}, \mathbf{c}_{rel})$, $\mathbf{d}_i^{res} = \sum_{j=1}^{N} \mathbf{d}_{2j}^{in} \mathbf{mask}_{ij}$, $\mathbf{d}^{out} = \text{Merge}(\mathbf{d}_1^{in}, \mathbf{d}^{res})$
relate_ae	$\mathbf{d}_1^{in}, \mathbf{d}_2^{in}$	$\mathbf{d}^{out}$	$attr$	find the object from $\mathbf{d}_1^{in}$ that has the same $attr$ with $\mathbf{d}_2^{in}$	$\mathbf{e}_i = \mathcal{M}_{attr}(\mathbf{v}_i)$, $\mathbf{mask}_{ij} = \text{sim}(\mathbf{e}_i, \mathbf{e}_j)$, $\mathbf{d}_i^{res} = \sum_{j=1}^{N} \mathbf{d}_{2j}^{in} \mathbf{mask}_{ij}$, $\mathbf{d}^{out} = \text{Merge}(\mathbf{d}_1^{in}, \mathbf{d}^{res})$
Output Modules					
query	$\mathbf{d}^{in}$	$\mathbf{a}$	$attr$	query the attribute $attr$ of the given input $\mathbf{d}^{in}$	$\mathbf{e}_i = \mathcal{M}_{attr}(\mathbf{v}_i)$, $\mathbf{e} = \mathbf{d}^{in} \cdot [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_N]$, $\mathbf{a}_{concept} = \text{sim}(\mathbf{e}, \mathbf{c}_{concept})$, $concept \in \mathcal{C}(attr)$ [3]
query_rel_s	$\mathbf{d}_1^{in}, \mathbf{d}_2^{in}$	$\mathbf{a}$	$rtype$	query the relationship between $\mathbf{d}_1^{in}$ (subject) and $\mathbf{d}_2^{in}$ (object)	$\mathbf{e}_{ij} = \mathcal{M}_{rtype}^{r}(\mathbf{v}_i, \mathbf{v}_j)$, $\mathbf{e} = \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbf{d}_{1i}^{in} \mathbf{e}_{ij} \mathbf{d}_{2j}^{in}$, $\mathbf{a}_{rel} = \text{sim}(\mathbf{e}, \mathbf{c}_{rel})$, $rel \in \mathcal{C}(rtype)$ [4]
query_rel_o	$\mathbf{d}_1^{in}, \mathbf{d}_2^{in}$	$\mathbf{a}$	$rtype$	query the relationship between $\mathbf{d}_1^{in}$ (object) and $\mathbf{d}_2^{in}$ (subject)	$\mathbf{a} = \text{query\_rel\_s}[rtype](\mathbf{d}_2^{in}, \mathbf{d}_1^{in})$
verify	$\mathbf{d}^{in}$	a	$concept, attr$	verify whether the attribute $attr$ of given input $\mathbf{d}^{in}$ is $concept$	$\mathbf{e}_i = \mathcal{M}_{attr}(\mathbf{v}_i)$, $\mathbf{e} = \mathbf{d}^{in} \cdot [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_N]$, $a = \text{sim}(\mathbf{e}, \mathbf{c}_{concept})$
choose	$\mathbf{d}_1^{in}, \mathbf{d}_2^{in}$	$\mathbf{a}$	$concept, attr$	choose whether $\mathbf{d}_1^{in}$ or $\mathbf{d}_2^{in}$ is of $concept$ in specified attribute $attr$	$\mathbf{a}_1 = \text{verify}[attr, concept](\mathbf{d}_1^{in})$, $\mathbf{a}_2 = \text{verify}[attr, concept](\mathbf{d}_2^{in})$
verify_rel_s	$\mathbf{d}_1^{in}, \mathbf{d}_2^{in}$	a	$rel, rtype$	verify whether $\mathbf{d}_1^{in}$ (subject) and $\mathbf{d}_2^{in}$ (object) are of relationship rel	$\mathbf{e}_{ij} = \mathcal{M}_{rtype}^{r}(\mathbf{v}_i, \mathbf{v}_j)$, $\mathbf{e} = \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbf{d}_{1i}^{in} \mathbf{e}_{ij} \mathbf{d}_{2j}^{in}$, $a = \text{sim}(\mathbf{e}, \mathbf{c}_{rel})$
verify_rel_o	$\mathbf{d}_1^{in}, \mathbf{d}_2^{in}$	a	$rel, rtype$	verify whether $\mathbf{d}_1^{in}$ (object) and $\mathbf{d}_2^{in}$ (subject) are of relationship rel	$a = \text{verify\_rel\_s}[rtype](\mathbf{d}_2^{in}, \mathbf{d}_1^{in})$
same	$\mathbf{d}^{in}$	a	$attr$	whether objects in $\mathbf{d}^{in}$ have the same $attr$	$\mathbf{e}_i = \mathcal{M}_{attr}(\mathbf{v}_i)$, $\mathbf{e} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{e}_i$, $a = \sum_{i=1}^{N} \mathbf{d}_i^{in} \text{sim}(\mathbf{e}_i, \mathbf{e})$
query_ae	$\mathbf{d}_1^{in}, \mathbf{d}_2^{in}$	a	$attr$	whether $\mathbf{d}_1^{in}$ and $\mathbf{d}_2^{in}$ have the same $attr$	$\mathbf{e}_i = \mathcal{M}_{attr}(\mathbf{v}_i)$, $\mathbf{e}^1 = \sum_{i=1}^{N} \mathbf{d}_{1i}^{in} \mathbf{e}_i$, $\mathbf{e}^2 = \sum_{i=1}^{N} \mathbf{d}_{2i}^{in} \mathbf{e}_i$, $a = \text{sim}(\mathbf{e}^1, \mathbf{e}^2)$
common	$\mathbf{d}_1^{in}, \mathbf{d}_2^{in}$	$\mathbf{a}$	-	what attribute do $\mathbf{d}_1^{in}$ and $\mathbf{d}_2^{in}$ share	For all possible $attr$s: $\mathbf{a}_{attr} = \text{query\_ae}[attr](\mathbf{d}_1^{in}, \mathbf{d}_2^{in})$
exist	$\mathbf{d}^{in}$	a	-	whether object $\mathbf{d}^{in}$ exists	$a = \max(\mathbf{d}^{in})$
intersect	a_1^{in}, a_2^{in}	a	-	whether both a_1^{in} AND a_2^{in} are true	$a = \min(a_1^{in}, a_2^{in})$
union	a_1^{in}, a_2^{in}	a	-	whether a_1^{in} OR a_2^{in} is true	$a = \max(a_1^{in}, a_2^{in})$

[1] In training, cross entropy loss is used for open questions (where the output module produces a vector $\mathbf{a}$), while binary cross entropy loss is used for binary questions (where the output module produces a scalar score a.

[2] We assign each relationship into one of the three predefined relationship types ($rtype$), which are `spatial` (*e.g.*, to the left of, on top of, etc.), `semantic` (*e.g.*, wearing, holding, etc.) and `spatial+semantic` (*e.g.*, sitting on, looking at, etc.).

[3] Here $\mathcal{C}(attr)$ represents the set of concepts of the given attribute $attr$.

[4] Here $\mathcal{C}(rtype)$ represents the set of relationships of the given relationship type $rtype$.