A Appendix

procedure RUN-MODULE($m, A, p, \boldsymbol{c_t}, \mathcal{I}$)
  $a_1 \leftarrow \sum_{i=1}^{L} A_i \cdot p_i$ // Read from stack
  $p \leftarrow$ 1D-conv$(p, [0, 0, 1])$ // decrement the stack pointer
if *no. of inputs==4* **then**
  $a_2 \leftarrow \sum_{i=1}^{L} A_i \cdot p_i$ // Read from stack
  $p \leftarrow$ 1D-conv$(p, [0, 0, 1])$ // decrement the stack pointer
  $o_m \leftarrow m(\mathcal{I}, \boldsymbol{c_t}, a_1, a_2)$
  end
  else
  $o_m \leftarrow m(\mathcal{I}, \boldsymbol{c_t}, a_1)$
  end
  $p \leftarrow$ 1D-conv$(p, [1, 0, 0])$ // increment the stack pointer
for $i = 1, ..., L$ **do**
  $A \leftarrow A \cdot (1 - p_i) + o_m \cdot p_i$ // Write to stack
  end
  return A, p

Algorithm 3: Operation of a module

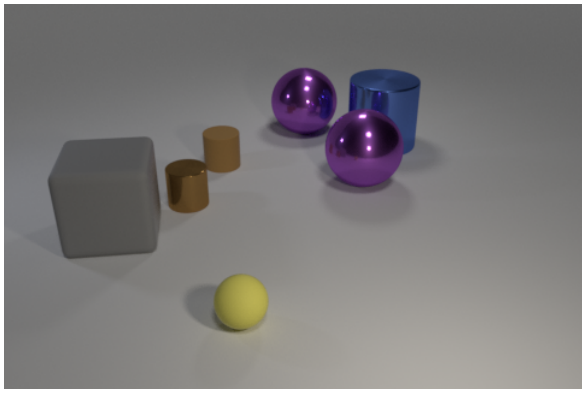

Figure 4: ***Q1****: What number of cylinders are gray objects or tiny brown matte objects?* ***A****: 1*
Q2*: Is the number of brown cylinders in front of the brown matte cylinder less than the number of brown rubber cylinders?* ***A****: no*

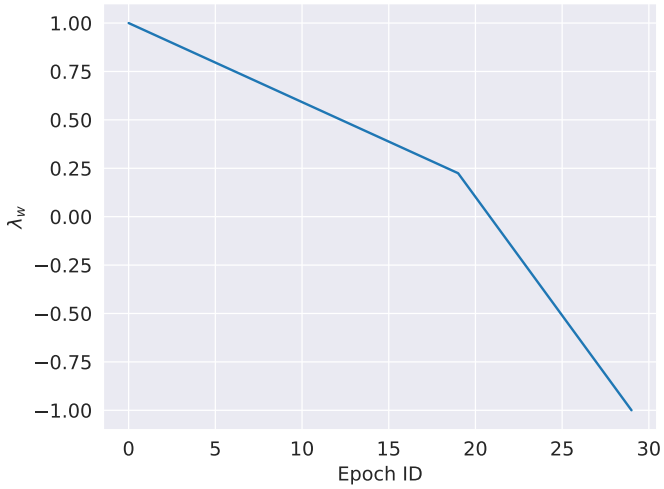

Figure 5: Plot of variation of λ_w with epochs.

A.1 Module schematic diagrams

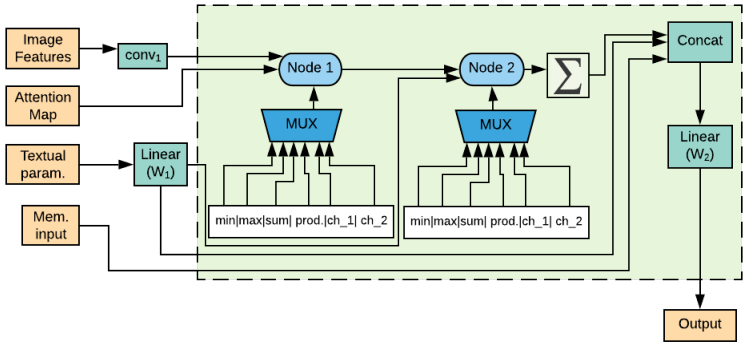

Figure 6: Answer Module schematic diagram (3 inputs)

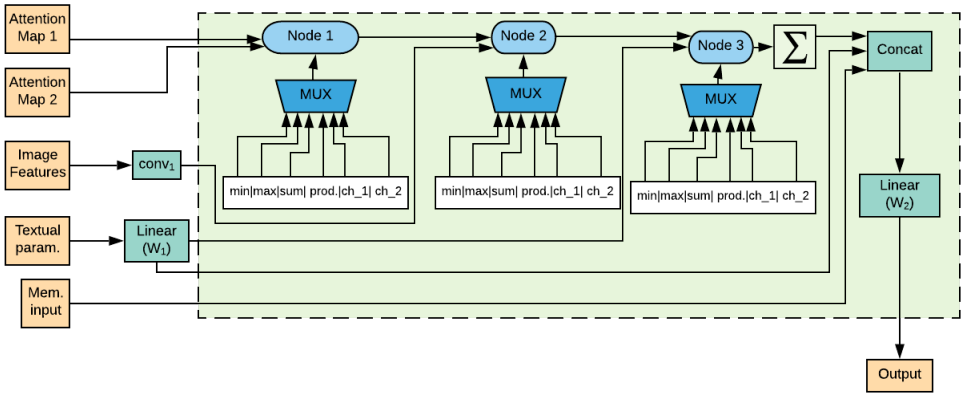

Figure 7: Answer Module schematic diagram (4 inputs)

A.2 Hand-crafted modules of Stack-NMN

module name	input attention	output type	implementation details (x: image feature map, c: textual parameter)
`Find`	(none)	attention	$a_{out} = \text{conv}_2\left(\text{conv}_1(x) \odot Wc\right)$
`Transform`	a	attention	$a_{out} = \text{conv}_2\left(\text{conv}_1(x) \odot W_1 \sum(a \odot x) \odot W_2 c\right)$
`And`	a_1, a_2	attention	$a_{out} = \text{minimum}(a_1, a_2)$
`Or`	a_1, a_2	attention	$a_{out} = \text{maximum}(a_1, a_2)$
`Filter`	a	attention	$a_{out} = \texttt{And}(a, \texttt{Find}())$, i.e. reusing `Find` and `And`
`Scene`	(none)	attention	$a_{out} = \text{conv}_1(x)$
`Answer`	a	answer	$y = W_1^T \left(W_2 \sum(a \odot x) \odot W_3 c\right)$
`Compare`	a_1, a_2	answer	$y = W_1^T \left(W_2 \sum(a_1 \odot x) \odot W_3 \sum(a_2 \odot x) \odot W_4 c\right)$
`NoOp`	(none)	(none)	(does nothing)

Table 8: Neural modules used in (Hu et al., 2018). The modules take image attention maps as inputs, and output either a new image attention a_{out} or a score vector y over all possible answers ($\odot$ is elementwise multiplication; $\sum$ is sum over spatial dimensions).

A.3 Visualization of module structure parameters

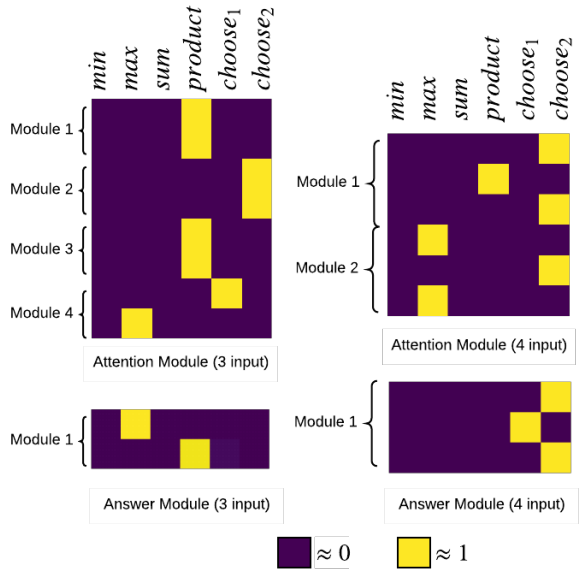

Figure 8: Visualization of module structure parameters (LNMN (9 modules)). For each module, each row denotes the $\boldsymbol{\alpha}' = \boldsymbol{\sigma}(\boldsymbol{\alpha})$ parameters of the corresponding node.

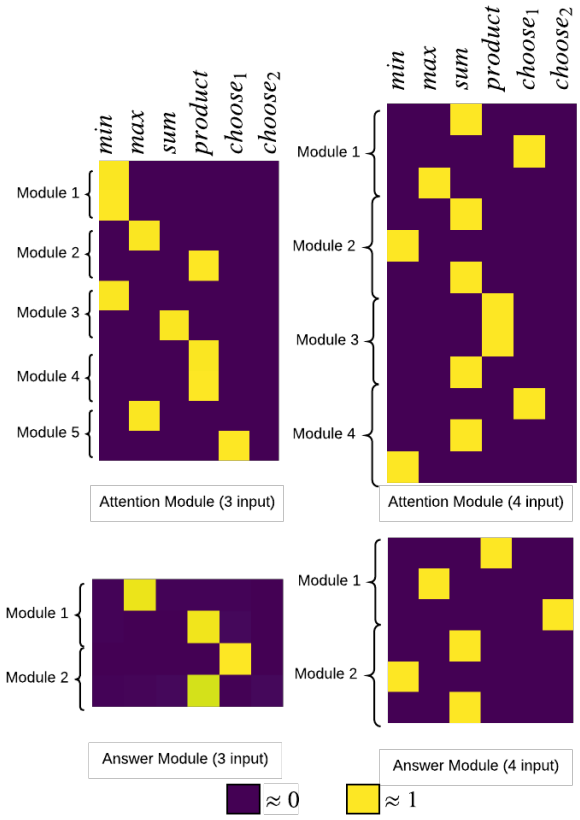

Figure 9: Visualization of module structure parameters (LNMN (14 modules)). For each module, each row denotes the $\alpha' = \sigma(\alpha)$ parameters of the corresponding node.

A.4 Results on Natural Image VQA datasets

Model	Overall	Yes/No	Number	Other
Stack-NMN	59.84	80.75	37.49	46.83
LNMN (9 modules)	57.67	80.41	36.65	42.82

Table 9: Test Accuracy on VQA v1 (Antol et al., 2015)

Model	Overall	Yes/No	Number	Other
Stack-NMN	58.23	77.06	37.48	46.59
LNMN (9 modules)	54.85	73.78	35.05	42.92

Table 10: Test Accuracy on VQA v2 (Goyal et al., 2017)