Appendix

A. Implementation Details

In both of the GT and Det experiments on the CLEVR dataset, we set the dimension of the label embedding (*i.e.*, d) and the dimension of $\mathbf{h}$ in all output modules to 128. The classifier consists of a simple multi-layer perceptron that maps the 128-dimentional features to the number of possible answers (*i.e.*, 28), and a softmax layer. We used Adam [1] optimizer with an initial learning rate 0.001 to train our module parameters. We trained for 5 epochs for the GT setting and 10 epochs for the Det setting, and we reduced the learning rate to 0.0001 after the first epoch. Each epoch takes about one hour with an Nvidia 1080Ti graphic card.

The mapping matrices of the `Describe` module are implemented differently between the GT and Det setting. In the GT setting, as each object has four attribute values corresponding to four attribute categories (*i.e.*, color, shape, size, material), and our node embedding is the concatenation of attribute label embeddings, the dimension of node embeddings is $4d$, where d is the dimension of label embeddings. We fix the order of label embedding concatenation as [color, shape, size, material], so we can extract node i's color feature by $[\mathbf{I}_d; \mathbf{0}_d; \mathbf{0}_d; \mathbf{0}_d] \cdot \mathbf{v}_i$, where $\mathbf{I}_d, \mathbf{0}_d$ are $d \times d$ identity matrix and zero matrix respectively. So in the GT setting, our four mapping matrixes are defines as:

$$\begin{aligned} \mathbf{M}_1 &= [\mathbf{I}_d; \mathbf{0}_d; \mathbf{0}_d; \mathbf{0}_d], \\ \mathbf{M}_2 &= [\mathbf{0}_d; \mathbf{I}_d; \mathbf{0}_d; \mathbf{0}_d], \\ \mathbf{M}_3 &= [\mathbf{0}_d; \mathbf{0}_d; \mathbf{I}_d; \mathbf{0}_d], \\ \mathbf{M}_4 &= [\mathbf{0}_d; \mathbf{0}_d; \mathbf{0}_d; \mathbf{I}_d]. \end{aligned} \tag{1}$$

In the Det setting, we regard $\mathbf{M}_k \in \mathbb{R}^{d \times d}$ as parameters and learn them automatically, which leads to an increase of the number of parameters (Det has 0.55M parameters while Gt only has 0.22M).

As for the attention functions in the CLEVR GT experiments, besides the label-space softmax which is mentioned in the main body, we have also tried the sigmoid activation. Specifically, we fused multiple label vectors into one vector via a fully connected layer, and then applied the sigmoid function to separately compute attention weights of each node and edge. Using this attention strategy, we can still obtain 100% accuracy in the GT setting, demonstrating that our model is robust and flexible.

In the VQAv2.0 dataset, we selected the most frequent 3000 answers from the training set, and predicted the target answer from these candidates. We used an LSTM as the question encoder and fused the 1024-dimensional question embedding with the output feature from our module network for the answer classification. For the training, we used Adam optimizer with an initial learning rate 0.0008 and we set batch size as 256. The learning rate was decayed by half every 50000 training steps and the training lasted for 100 epochs.

B. Failure Cases of the CLEVR Det Setting

We classify the failure cases in the CLEVR Det setting into three categories:

1. The coordinate detection is inaccurate (Figure 1).
2. Some objects are occluded (Figure 2).
3. The mask-based object division is inaccurate (Figure 3). Specifically, when we propose objects based on the generated mask from the "Attend" modules of [2], we may wrongly propose more or less objects due to the blurring boundaries.

In all of Figure 1, 2, and 3, **we mark the reasoning steps that cause mistakes in red box**. We can see that using our X reasoning over scene graphs, we can easily track the reasoning process and diagnose where and why the model makes a mistake, which is an inspiring step towards the explainable AI.

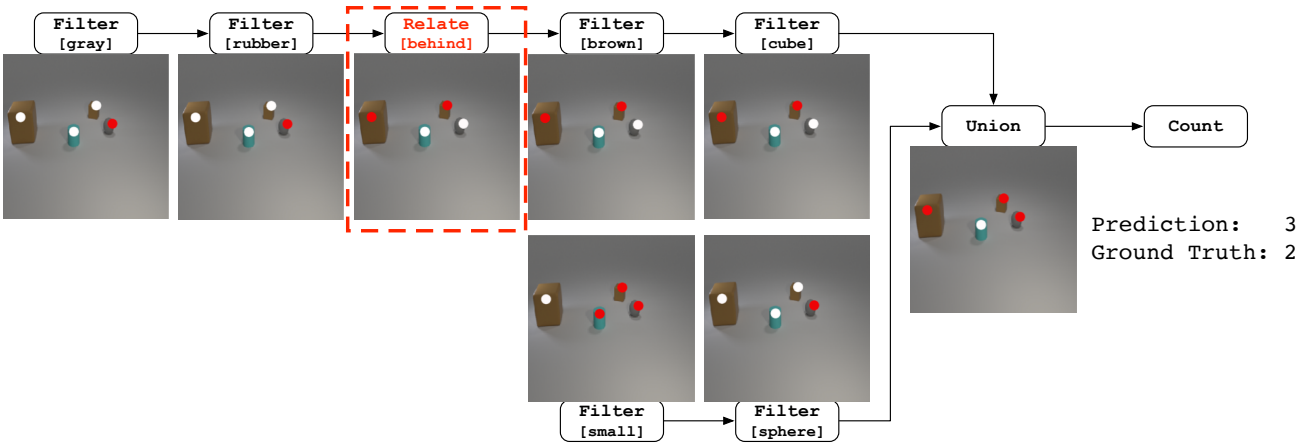
C. Case Study on the VQAv2.0

In the VQAv2.0 experiments, we predict a probability distribution over our modules at each step, and then feed the soft fusion of their outputs into next step. We set the reasoning length to 3 and force the last module to be `Describe`. We show a typical sample of the VQAv2.0 dataset in Figure 4. The top row is the modules with the most probability at each step, while the bottom row shows the results of `Relate` at Step 2. We can see that even though the question "what is the man wearing on his head" explicitly requires the relationship reasoning (*i.e.*, find the "man" first, and then move the focus to his head), our model scarcely relies on the results of the `Relate` module. Instead, it can directly focus on the "head" and give the correct prediction. We think this is due to the simplicity of questions, which is a shortcoming of the VQAv2.0 dataset.

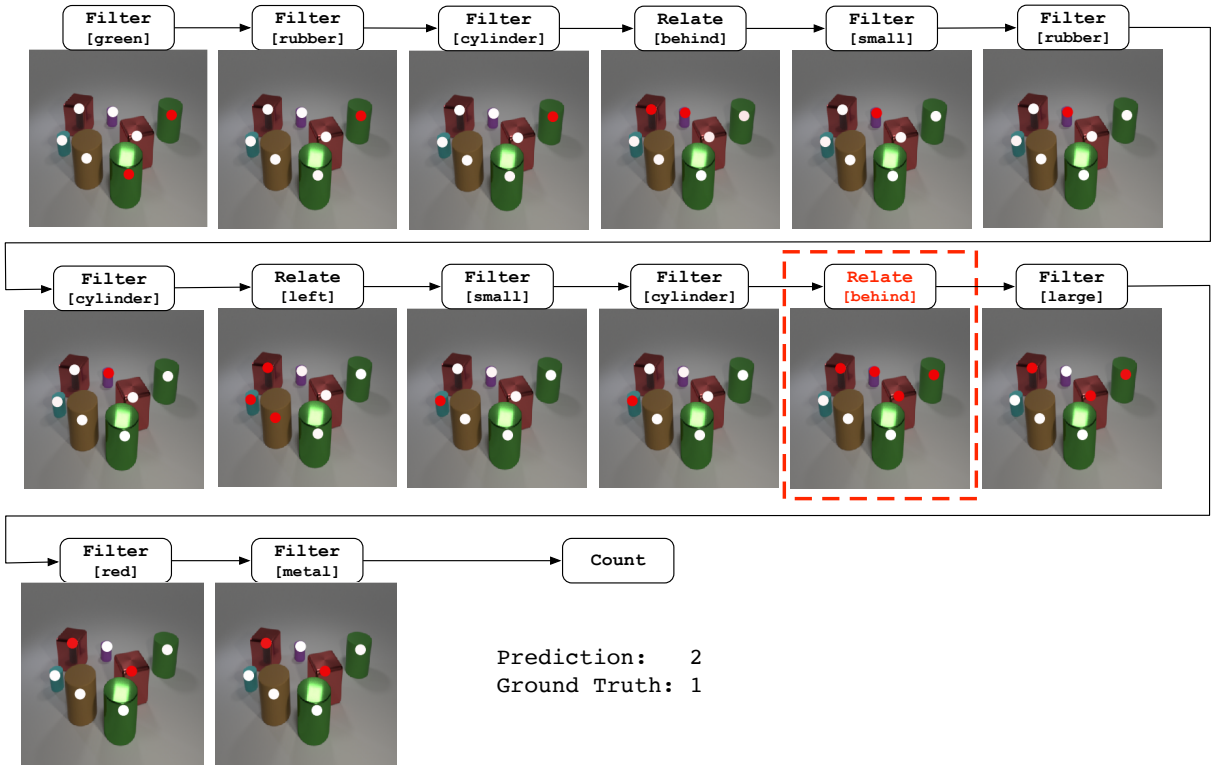
References

[1] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[2] D. Mascharka, P. Tran, R. Soklaski, and A. Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *CVPR*, 2018. 1

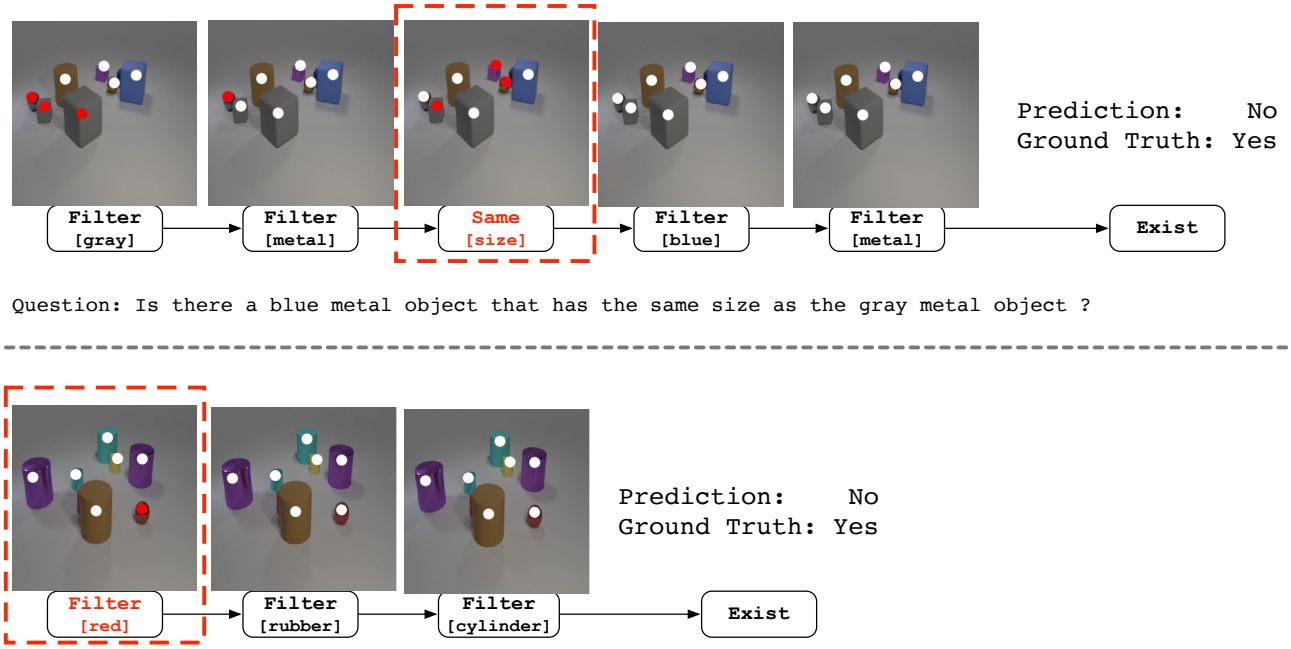


Question: What number of objects are tiny spheres or brown blocks behind the gray matte object ?



Question: There is a tiny cylinder left of the tiny matte cylinder that is behind the green rubber cylinder ; how many big red metal things are behind it ?

Figure 1: Failure cases caused by the inaccurate coordinate detection. Top case: the large brown cube is not behind the gray rubber object. Bottom case: a large red cube is wrongly recognized to be behind the tiny cylinder.



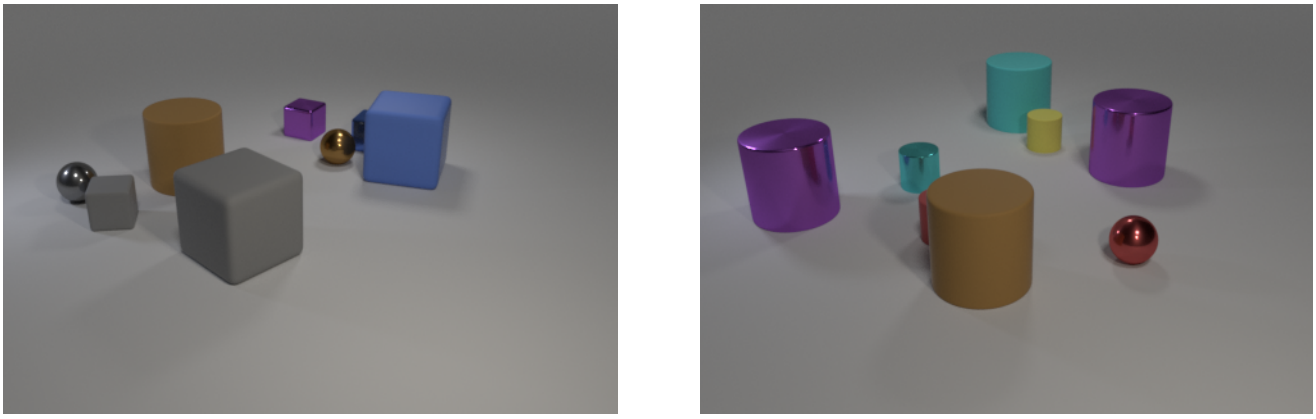
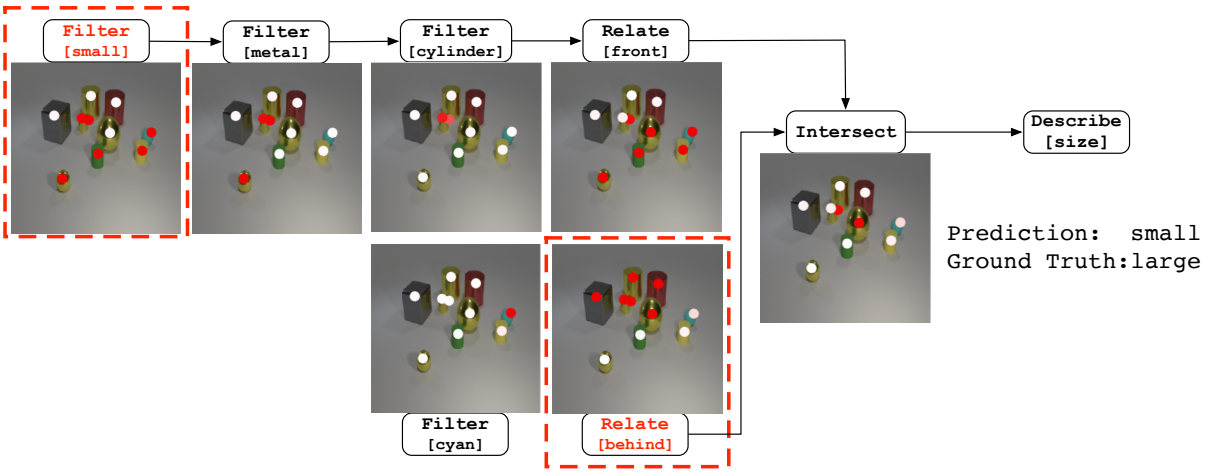
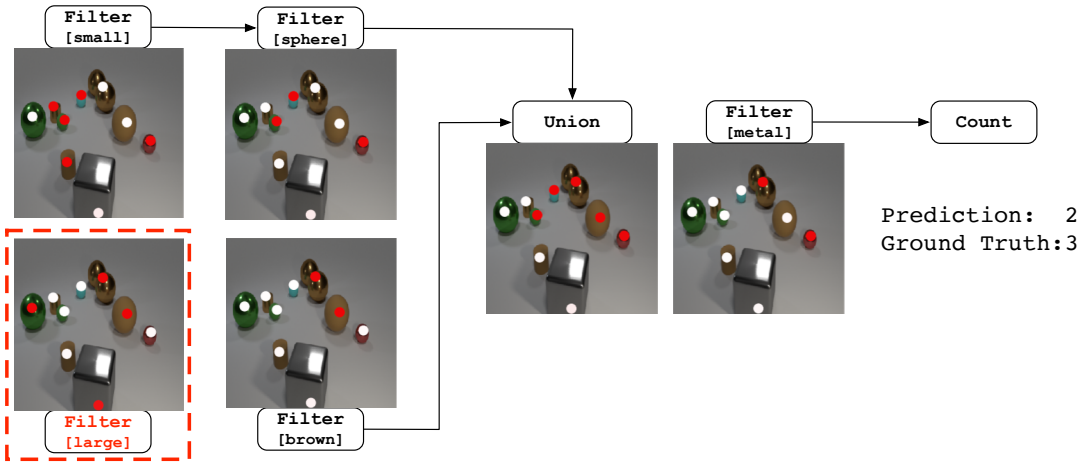


Figure 2: Failure cases caused by occluded objects. We show the high-resolution images here, and we can see that 1) in the left image, there is a small blue cube behind the large blue cube occluded; 2) in the right image, there is a red cylinder behind the large brown cylinder occluded. These occluded objects do not have corresponding dots in the reasoning results, leading to a wrong prediction "No" while the actual answer is "Yes".



Question: There is a object that is both behind the cyan object and in front of the small metal cylinder ; what size is it ?



Question: What number of metal objects are large brown objects or tiny spheres ?

Figure 3: Failure cases caused by the inaccurate object division. Top case: two dots are assigned to the same object. Bottom case: two adjacent objects with the same attribute values (*i.e.*, large, brown, sphere, metal) are recognized as one object, which makes the predicted number less than the ground truth answer.

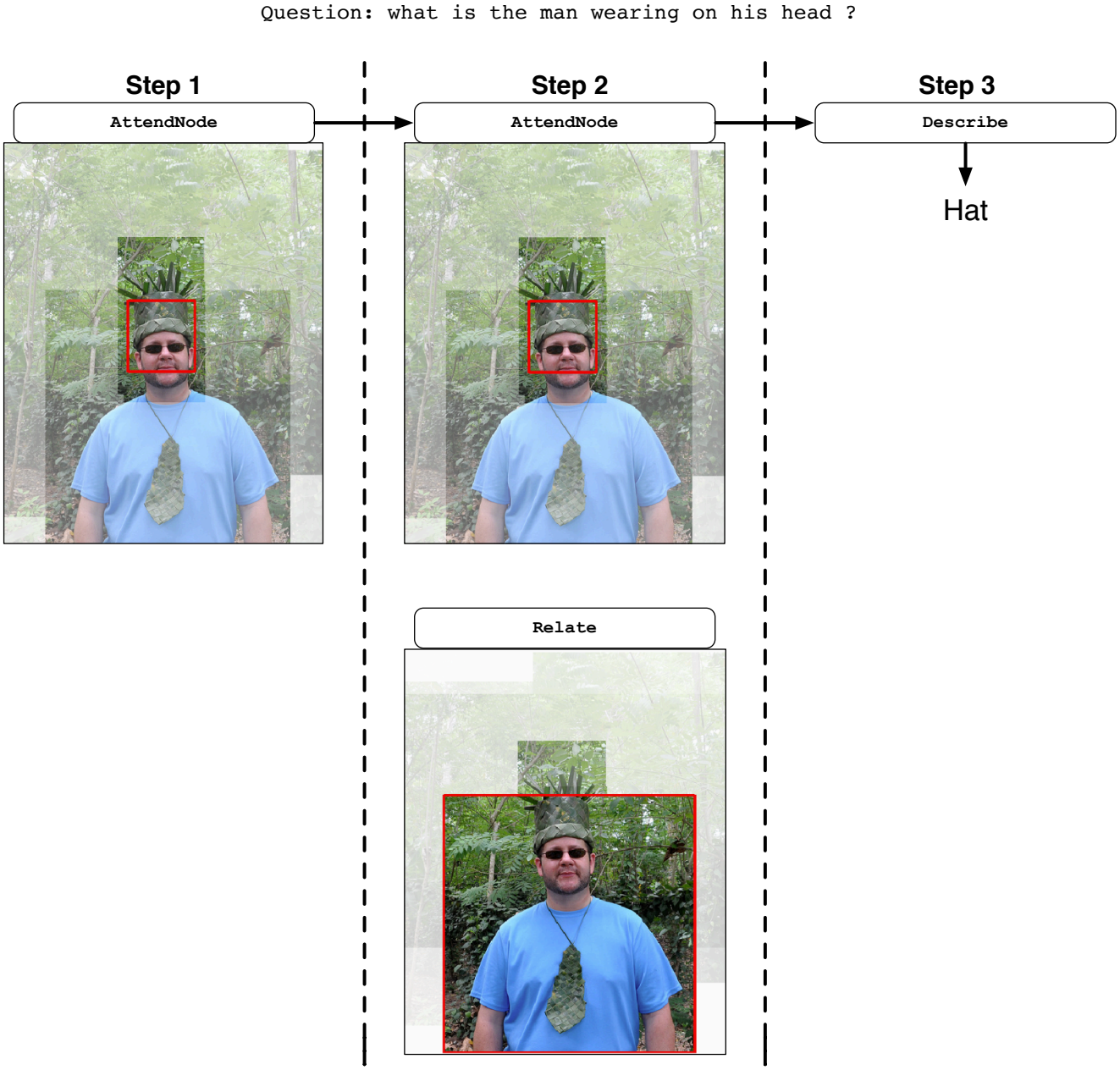


Figure 4: A typical case of the VQAv2.0 dataset. The top row lists the modules with the maximum probability at each step. We can see that even the question "what is the man wearing on his head" explicitly requires the relationship understanding, the module `Relate` is still not necessary as the target region can be directly found. We argue the simplicity of question annotations is a major shortcoming of the VQAv2.0 dataset.