CLEVR-Ref+: Diagnosing Visual Reasoning with Referring Expressions (Supplementary Material)

Runtao Liu[1], Chenxi Liu[2](✉), Yutong Bai[3], Alan Yuille[2]
[1]Peking University [2]Johns Hopkins University [3]Northwestern Polytechnical University
runtao219@gmail.com cxliu@jhu.edu ytongbai@gmail.com alan.l.yuille@gmail.com

In this supplementary material, we begin by providing network architecture details of IEP-Ref to supplement Section 4.1 of the main paper. We then provide more analysis of the four models' performance on CLEVR-Ref+, to supplement Section 4.2 of the main paper. Finally, we show more qualitative examples (referring expression and ground truth box/mask) from CLEVR-Ref+.

A. Network Architectures in IEP-Ref

In Figure 7 of the main paper, we listed all modules used in our IEP-Ref model (except `Segment`). In IEP-Ref, each of these modules is parameterized with a small fully convolutional network and belongs to one of the following 4 categories:

- **Preprocess**: This component maps the image to the feature tensor. Its output is the input to the `Scene` module. See Table 1 for the network architecture.
- **Unary**: This includes the `Scene`, `Filter_X`, `Unique`, `Relate`, `Same_X` modules. It transforms one feature tensor to another. See Table 2 for the network architecture.
- **Binary**: This includes the `And` and `Or` modules. It transforms two feature tensors to one. See Table 3 for the network architecture.
- **Postprocess**: This only includes the `Segment` module. It transforms the 128-channel feature tensor to a 1-channel segmentation mask. See Table 4 for the network architecture.

Network architectures for **Preprocess**, **Unary**, **Binary** are directly inherited from IEP [3].

Layer	Output size
Input image	$3 \times 320 \times 320$
ResNet101 [1] conv4_6	$1024 \times 20 \times 20$
Conv($3 \times 3, 1024 \to 128$)	$128 \times 20 \times 20$
ReLU	$128 \times 20 \times 20$
Conv($3 \times 3, 128 \to 128$)	$128 \times 20 \times 20$
ReLU	$128 \times 20 \times 20$

Table 1: Network architecture for the **Preprocess** module.

Index	Layer	Output size
(1)	Previous module output	$128 \times 20 \times 20$
(2)	Conv($3 \times 3, 128 \to 128$)	$128 \times 20 \times 20$
(3)	ReLU	$128 \times 20 \times 20$
(4)	Conv($3 \times 3, 128 \to 128$)	$128 \times 20 \times 20$
(5)	Residual: Add (1) and (4)	$128 \times 20 \times 20$
(6)	ReLU	$128 \times 20 \times 20$

Table 2: Network architecture for the **Unary** modules.

Index	Layer	Output size
(1)	Previous module output	$128 \times 20 \times 20$
(2)	Previous module output	$128 \times 20 \times 20$
(3)	Concatenate (1) and (2)	$256 \times 20 \times 20$
(4)	Conv($1 \times 1, 256 \to 128$)	$128 \times 20 \times 20$
(5)	ReLU	$128 \times 20 \times 20$
(6)	Conv($3 \times 3, 128 \to 128$)	$128 \times 20 \times 20$
(7)	ReLU	$128 \times 20 \times 20$
(8)	Conv($3 \times 3, 128 \to 128$)	$128 \times 20 \times 20$
(9)	Residual: Add (5) and (8)	$128 \times 20 \times 20$
(10)	ReLU	$128 \times 20 \times 20$

Table 3: Network architecture for the **Binary** modules.

B. More Model Analysis on CLEVR-Ref+

B.1. Number of Objects in a Scene

We suspect that the more objects in a scene, the harder for the model to carry out the referring reasoning steps. In Figure 1 we plot the performance of each model with respect to the number of objects in a scene. All models drop in performance when the number of objects increases, suggesting that the models tend to struggle when dealing with too many objects.

Layer	Output size
Previous module output	$128 \times 20 \times 20$
Unary module	$128 \times 20 \times 20$
Conv$(1 \times 1, 128 \to 128)$	$128 \times 20 \times 20$
ReLU	$128 \times 20 \times 20$
Bilinear upsample	$128 \times 320 \times 320$
Conv$(1 \times 1, 128 \to 128)$	$128 \times 320 \times 320$
ReLU	$128 \times 320 \times 320$
Conv$(1 \times 1, 128 \to 32)$	$32 \times 320 \times 320$
ReLU	$32 \times 320 \times 320$
Conv$(1 \times 1, 32 \to 4)$	$4 \times 320 \times 320$
ReLU	$4 \times 320 \times 320$
Conv$(1 \times 1, 4 \to 1)$	$1 \times 320 \times 320$

Table 4: Network architecture for the `Segment` module.

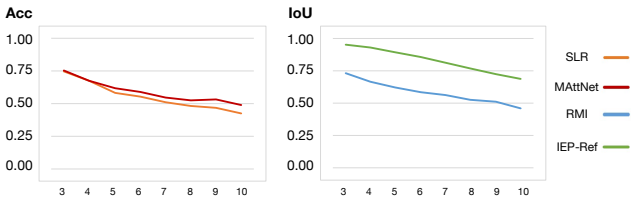


Figure 1: Effect of number of objects in a scene on referring detection or segmentation performance.

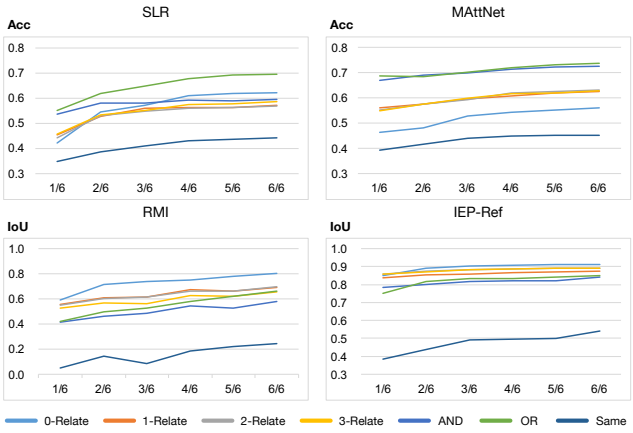


Figure 2: Performance across different referring expression categories throughout training. We inspect the performance every 1/6 of the entire training iterations.

B.2. Schedule of Acquiring Reasoning Abilities

We are interested to see if throughout the training process, the network exhibit a schedule of acquiring various reasoning abilities (e.g. spatial reasoning, logic etc). From Figure 2, it seems that no such schedule was developed, and performance steadily increase across different referring expression categories. This may be due to the random sampling during training, instead of active learning (c.f. [4]).

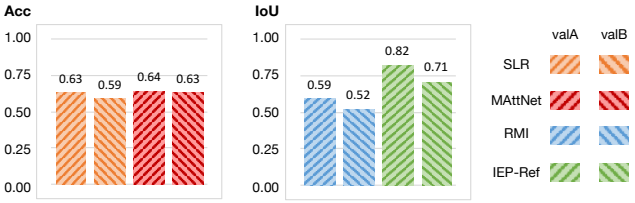


Figure 3: Different models' performance on *valA* and *valB* of the CLEVR CoGenT data.

B.3. Novel Compositions

To further test the models' generalization ability, we also conducted experiments on the Compositional Generalization Test (CoGenT) data provided by CLEVR [2]. Here models are trained on objects with only a subset of all combinations, and then tested on both the same subset of combinations (*valA*) and another subset of combinations (*valB*). Results are summarized in Figure 3. We see a very small gap for detection models, suggesting that they have learned compositionality to generalize well. The gap for segmentation models, on the other hand, is larger.

C. More Data Examples from CLEVR-Ref+

The remaining pages show random images, referring expressions, and the referring ground truth from our CLEVR-Ref+ dataset. In particular, we choose at least one example from each referring expression category (the 7 middle columns in Table 3 of the main paper). We show both detection ground truth (Figure 4) and segmentation ground truth (Figure 5).

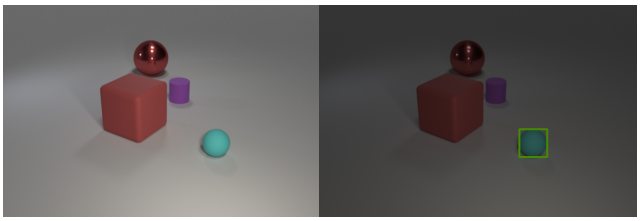
References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016. 1

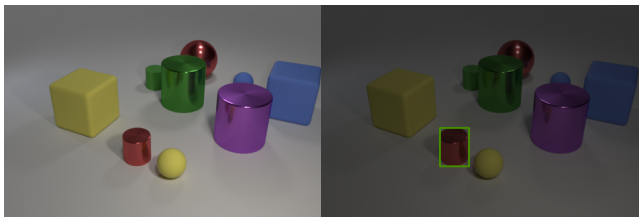
[2] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, pages 1988–1997. IEEE Computer Society, 2017. 2

[3] J. Johnson, B. Hariharan, L. van der Maaten, J. Hoffman, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick. Inferring and executing programs for visual reasoning. In *ICCV*, pages 3008–3017. IEEE Computer Society, 2017. 1

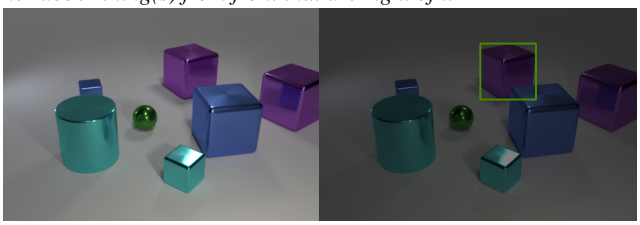
[4] I. Misra, R. B. Girshick, R. Fergus, M. Hebert, A. Gupta, and L. van der Maaten. Learning by asking questions. In *CVPR*, pages 11–20. IEEE Computer Society, 2018. 2



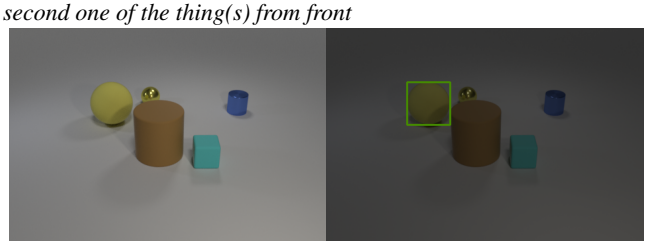
(a) *Look at matte thing that is on the left side of the red object that is behind the second one of the object(s) from right; The first one of the rubber thing(s) from front that are right of it*



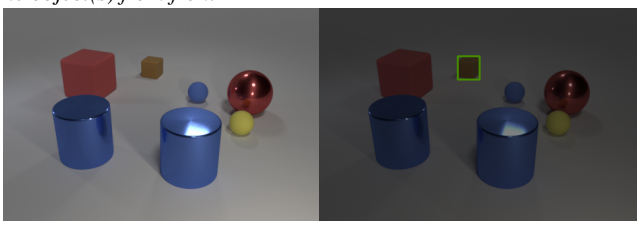
(b) *The objects that are the seventh one of the thing(s) from right that are in front of the nineth one of the thing(s) from front or the second one of the thing(s) from front*



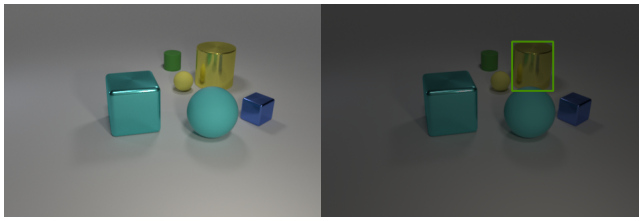
(c) *The big metallic object(s) that are both to the left of the third one of the large thing(s) from left and on the right side of the first one of the object(s) from front*



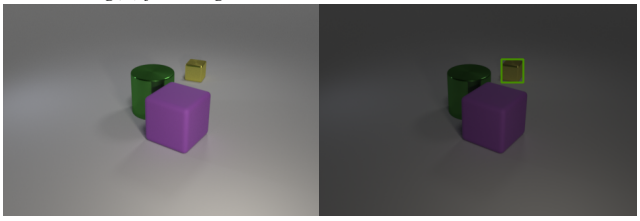
(d) *The fully visible yellow ball(s)*



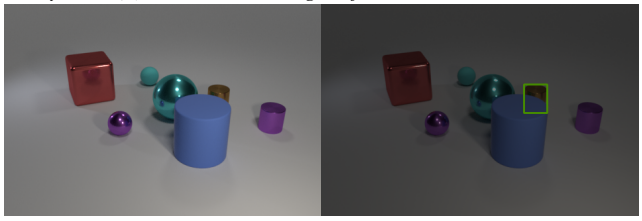
(e) *Any other things that are the same shape as the fourth one of the rubber thing(s) from right*



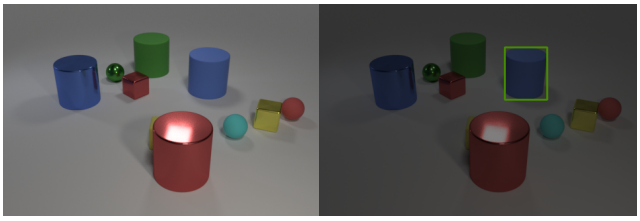
(f) *Find object that is behind the fifth one of the object(s) from left; The cylinder(s) that are to the right of it*



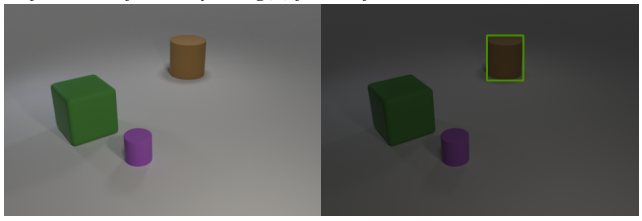
(g) *Look at partially visible object(s); The second one of the thing(s) from left that are on the right side of it*



(h) *The second one of the shiny cylinder(s) from right that are to the right of the thing that is behind the thing that is on the left side of the first one of the tiny thing(s) from left*

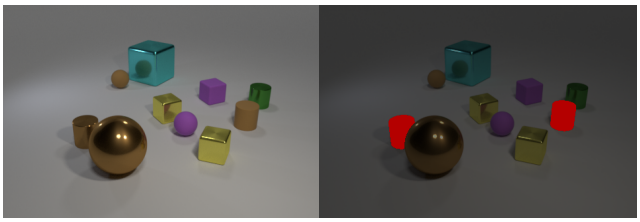


(i) *The blue things that are either the fourth one of the thing(s) from right or the first one of the tiny ball(s) from front*

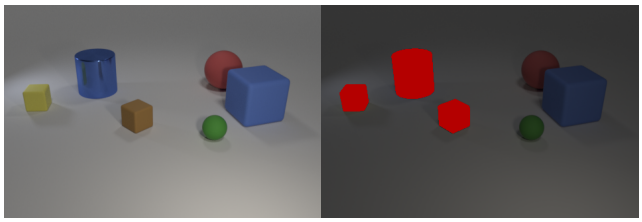


(j) *The matte object(s) that are behind the second one of the cylinder(s) from right and on the right side of the first one of the object(s) from left*

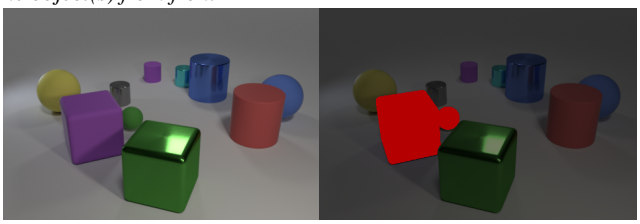
Figure 4: Referring object detection examples from CLEVR-Ref+.



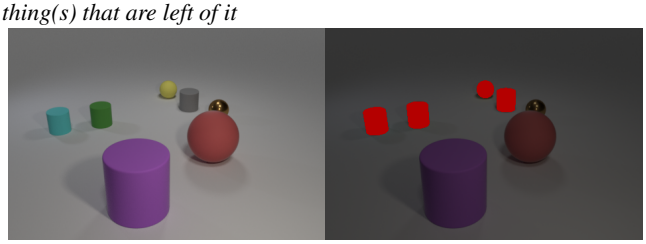
(a) *Any other things that are the same shape as the seventh one of the object(s) from front*



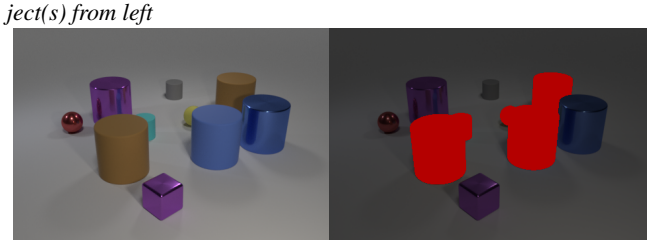
(b) *Look at rubber ball that is to the left of the red ball(s); The thing(s) that are left of it*



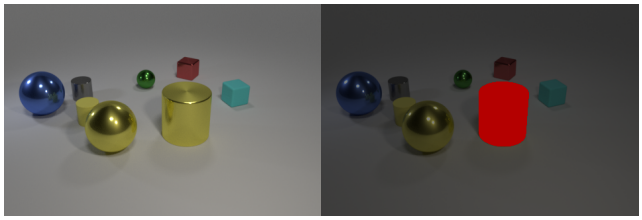
(c) *The rubber object(s) that are to the right of the sixth one of the rubber thing(s) from right and to the left of the fifth one of the object(s) from left*



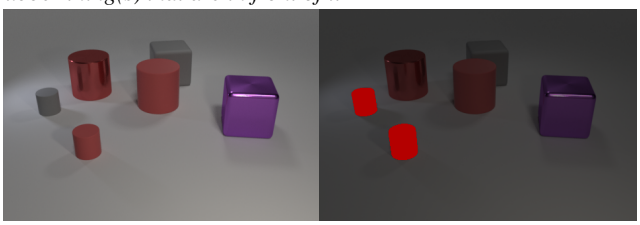
(d) *The fully visible small thing(s)*



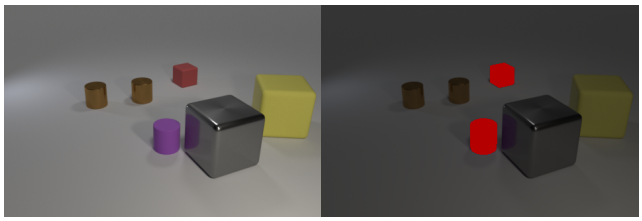
(e) *Look at tiny rubber cylinder that is behind the tiny object that is on the right side of the seventh one of the cylinder(s) from front; The rubber thing(s) that are in front of it*



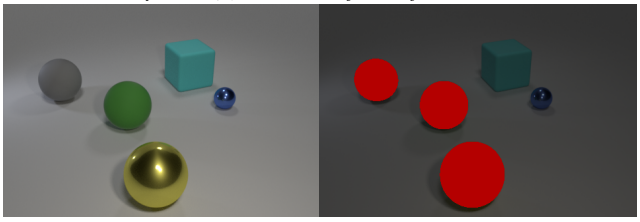
(f) *The big things that are the sixth one of the object(s) from left or the seventh one of the object(s) from right*



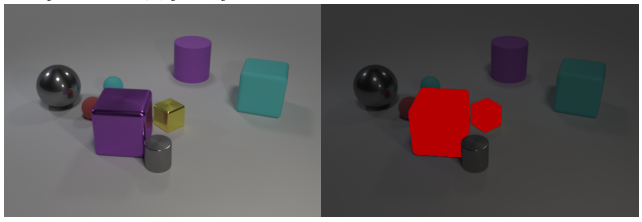
(g) *Find the second one of the red rubber thing(s) from left; The fully visible rubber cylinder(s) that are in front of it*



(h) *Any other tiny object(s) made of the same material as the second one of the cube(s) from front*



(i) *Look at object that is to the right of the fourth one of the big object(s) from front; The ball(s) that are to the left of it*



(j) *The metallic object(s) that are behind the fourth one of the object(s) from right and in front of the fourth one of the thing(s) from front*

Figure 5: Referring image segmentation examples from CLEVR-Ref+.