Self-supervised Cross-view Representation Reconstruction for Change Captioning Supplementary Material

Yunbin Tu[1], Liang Li[2,6]*, Li Su[1,3]*, Zheng-Jun Zha[4], Chenggang Yan[5,6], Qingming Huang[1,2,3]
[1]University of Chinese Academy of Sciences, Beijing, China
[2]Key Lab of Intelligent Information Processing, ICT, CAS, Beijing, China
[3]Peng Cheng Laboratory, Shenzhen, China
[4]University of Science and Technology of China, Hefei, China
[5]Hangzhou Dianzi University, Hangzhou, China
[6]Lishui Institute of Hangzhou Dianzi University, Hangzhou, China
tuyunbin22@mails.ucas.ac.cn, liang.li@ict.ac.cn, {suli,qmhuang}@ucas.ac.cn

1. Experiments

In this supplementary material, we will show more experimental results. First, we show the implementation details on the four datasets. Second, we provide the discussion of trade-off parameters on the four datasets. Next, we show more ablation studies on the four datasets. Finally, we show more qualitative examples on the four datasets.

1.1. Implementation Details

We provide more implementation details of our method. During training, the batch sizes and learning rates of our method on the four datasets are shown in Table 1. We train the model to convergence with 10K iterations in total. Both training and inference are implemented with PyTorch on an RTX 3090 GPU. The used resources on the four datasets are shown in Table 2. We can find that our method does not need much training time and GPU memory, so it can be easily reproduced by other researchers.

	Batch Size	Learning Rate
CLEVR-Change	128	2×10^{-4}
CLEVR-DC	128	2×10^{-4}
Spot-the-Diff	32	2×10^{-4}
Image Editing Request	16	1×10^{-4}

Table 1. The training parameters on the four datasets.

	Training Time	GPU Memory
CLEVR-Change	3 hours	20G
CLEVR-DC	1.5 hours	15.6G
Spot-the-Diff	20 minutes	5G
Image Editing Request	15 minutes	4.3G

Table 2. Used training time and GPU memory on the four datasets.

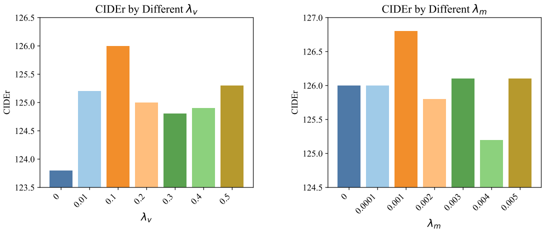


Figure 1. The effects of λ_v and λ_m on CLEVR-Change.

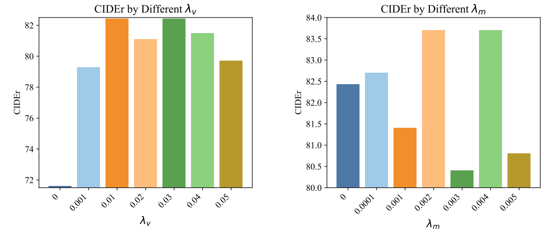


Figure 2. The effects of λ_v and λ_m on CLEVR-DC.

1.2. Study on the Trade-off Parameters

In this section, we discuss the trade-off parameters λ_v and λ_m in Eq. (13) of the main paper on the four datasets. Both parameters are to balance the contributions from the caption generator, SCORER, and CBR. In Figure 1, We first analyze the effect of λ_v on the CLEVR-Change dataset. We find that the performance of SCORER changes under different values, because the model will focus much on one part

*Corresponding authors

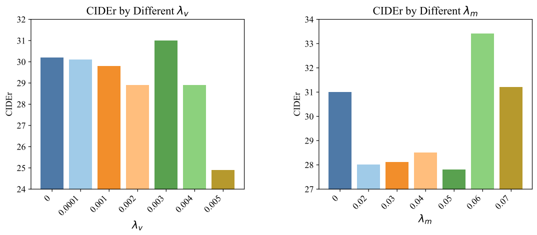


Figure 3. The effects of λ_v and λ_m on Image Editing Request.

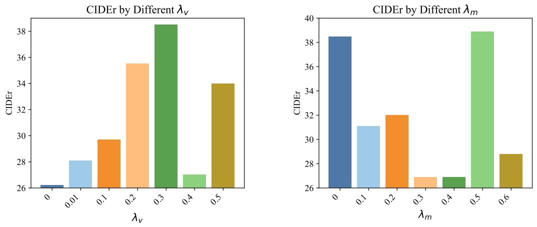


Figure 4. The effects of λ_v and λ_m on Spot-the-Diff.

Ablation	B	M	R	C	S
Subtraction	46.2	31.5	63.4	68.5	13.9
RR	46.9	31.7	64.2	71.6	14.6
SCORER	**49.5**	**33.4**	66.0	82.4	15.8
RR+CBR	47.2	32.3	64.4	73.0	15.0
SCORER+CBR	49.4	**33.4**	**66.1**	**83.7**	**16.2**

Table 3. Ablation study on CLEVR-DC.

Ablation	B	M	R	C	S
Subtraction	6.7	13.7	37.4	22.1	8.7
RR	8.3	14.3	39.2	30.2	12.4
SCORER	9.6	14.6	39.5	31.0	**12.6**
RR+CBR	7.1	14.6	40.6	31.9	12.3
SCORER+CBR	**10.0**	**15.0**	**39.6**	**33.4**	**12.6**

Table 4. Ablation study on Image Editing Request.

Ablation	B	M	R	C	S
Subtraction	7.2	11.9	28.9	29.4	13.0
RR	7.4	12.0	27.6	26.2	14.2
SCORER	9.4	**13.8**	**32.0**	38.5	**19.3**
RR+CBR	8.1	11.7	30.6	32.4	15.6
SCORER+CBR	**10.2**	12.2	31.9	**38.9**	18.4

Table 5. Ablation study on Spot-the-Diff.

but ignore the supervision from the other. We empirically set λ_v to 0.1. Then, we fix λ_v to discuss the effect of λ_m for SCORER+CBR and set it as 0.001. In Figure 2, we also first analyze the effect of λ_v on the CLEVR-DC dataset. We empirically set λ_v to 0.01. Then, we fix λ_v to discuss λ_m. We find that the CIDEr scores are identical when setting λ_m as 0.002 and 0.004. Thus, we further compare their SPICE scores (16.0 vs. 16.2) and set λ_m to 0.004. By that analogy, we discuss the effect of λ_v and λ_m on the Image Editing Request and Spot-the-Diff datasets. We empirically set λ_v and λ_m as 0.003 and 0.06 on Image Editing Request; 0.3 and 0.5 on Spot-the-Diff.

1.3. Ablation Study

We carry out ablation studies to validate the effectiveness of our method. (1) Subtraction is a transformer-based baseline model which computes difference features by direct subtraction. (2) RR refers to vanilla representation reconstruction without cross-view contrastive alignment. (3) SCORER is the proposed self-supervised cross-view representation reconstruction network. (4) CBR means the proposed module of cross-modal backward reasoning.

Results on CLEVR-DC. Table 3 shows the ablation studies of our method on the CLEVR-DC dataset, which are evaluated in terms of total performance. We can draw the same conclusion from the ablative variants. Compared with the baseline model of Subtraction, it is effective to first compute the aligned properties and then deduce the difference features. The proposed SCORER first learns the representations that are invariant under extreme viewpoint changes for a pair of similar images, by maximizing their cross-view contrastive alignment. Then, SCORER can fully mine their common features to reconstruct the representations of unchanged objects, thereby learning a stable difference representation for caption generation. Besides, CBR is helpful to improve the quality of generated sentences, which shows that it does enforce the yielded sentence to be informative about the learned difference.

Results on Image Editing Request. Table 4 shows the ablation studies of our method on the Image Editing Request dataset, where two images in the pair are aligned and the edited objects on this dataset are usually inconspicuous. We can obtain the same observations. Match-based strategy (RR) performs better than the strategy of direct subtraction. The proposed SCORER can fully align and mine the common features between two images, so as to reconstruct reliable unchanged representations for learning a stable difference representation. When we implement CBR, the performance of SCORER+CBR is further boosted, which shows that CBR is helpful to improve captioning quality.

Results on Spot-the-Diff. Table 5 shows the ablation studies of our method on the Spot-the-Diff dataset. We can find that compared with the baseline model of Subtraction, the improvement is not significant when using the model of representation reconstruction. Our conjecture is that image pairs on this dataset are well-aligned, so direct

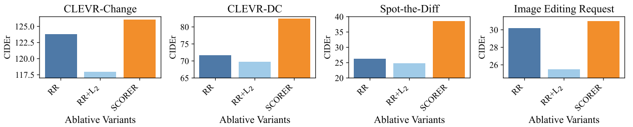


Figure 5. Effect of cross-view contrastive alignment on four datasets.

subtraction also can capture some salient changes. When we perform cross-view contrastive alignment, the performance of SCORER is significantly boosted, which shows that it facilitates correct alignment between unchanged objects, so as to help locate fine-grained changes. When introducing CBR, we observe that the results of RR+CBR and SCORER+CBR are not improved significantly. As we discussed in the main paper, the image pairs on this dataset actually contain one or more changes. For fair-comparison, we conduct experiments mainly based on the single-change setup. This makes the "hallucination" representation, which is reversely modeled by the "before" representation and single-change caption, not fully matched with the "after" representation. In this situation, the performances of RR+CBR and SCORER+CBR do not gain significant improvement.

Effect of Cross-view Contrastive Alignment. We study the effect of cross-view contrastive alignment, which is key to learn view-invariant image representations. Besides, we try L_2 distance metric to achieve this goal by only maximizing alignment of similar images. Fig. 5 shows comparison results among RR (without alignment constraint), RR+L_2, and SCORER. We find that SCORER achieves the best result, while the performance of RR+L_2 is the worst. The comparison results validate that it is necessary to build contrastive alignment between similar/dissimilar images, which helps the model focus more on the change of feature and resist feature shift. As a result, the model can capture the stable difference representation between two images for caption generation.

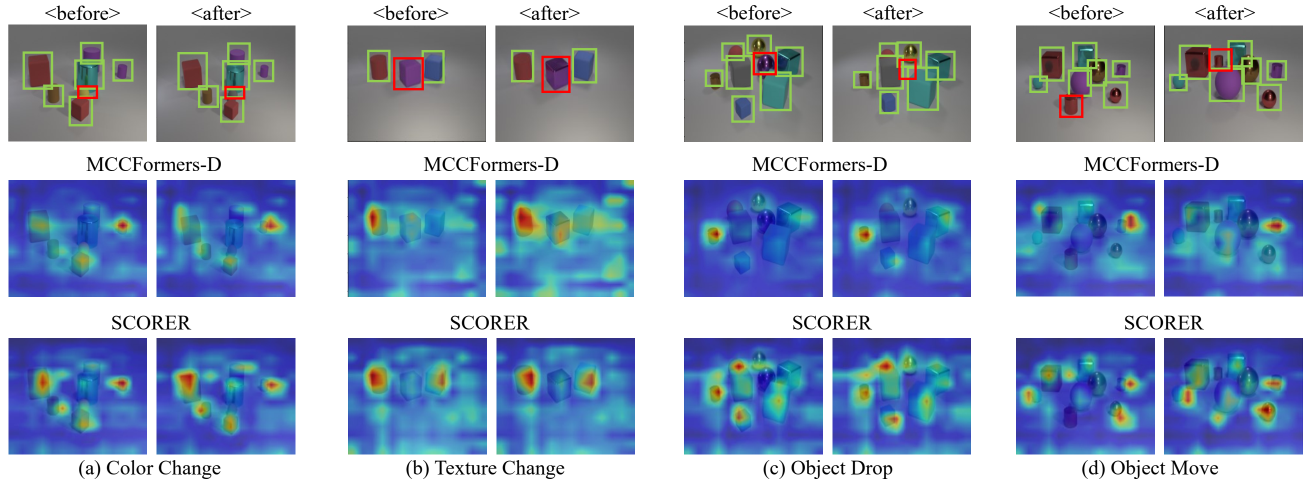
Study of Different Fusion Strategies to Model Difference Representation. In the Eq. (7) of main paper, we obtain the difference representation between two images by concatenating the changed features of each image. In order to validate whether concatenation is a good choice in the case of extreme viewpoint changes, besides concatenation ("cat"), we try to use other fusion strategies to model the difference representation between two images: sum, hadamard product, and respective interaction with words and then "cat". The experiment is conducted on the CLEVR-DC dataset with extreme viewpoint changes. Here, we report the comparison results under CIDEr metric: "cat" (82.4), sum (61.9), hadamard product (61.6), respective interaction with words and then "cat" (81.3). The results validate the effectiveness of our choice: using "cat" to construct an omni-representation of change between cross-view images. With this omni-representation, our model can accurately locate changed regions on two images during word generation, even under extreme viewpoint changes. This benefits from view-invariant representation learning and cross-modal backward reasoning.

1.4. Qualitative Analysis

In this section, we provide more visualization results about the alignment of unchanged objects, and the generated captions along with the attention weight at each word on the four datasets.

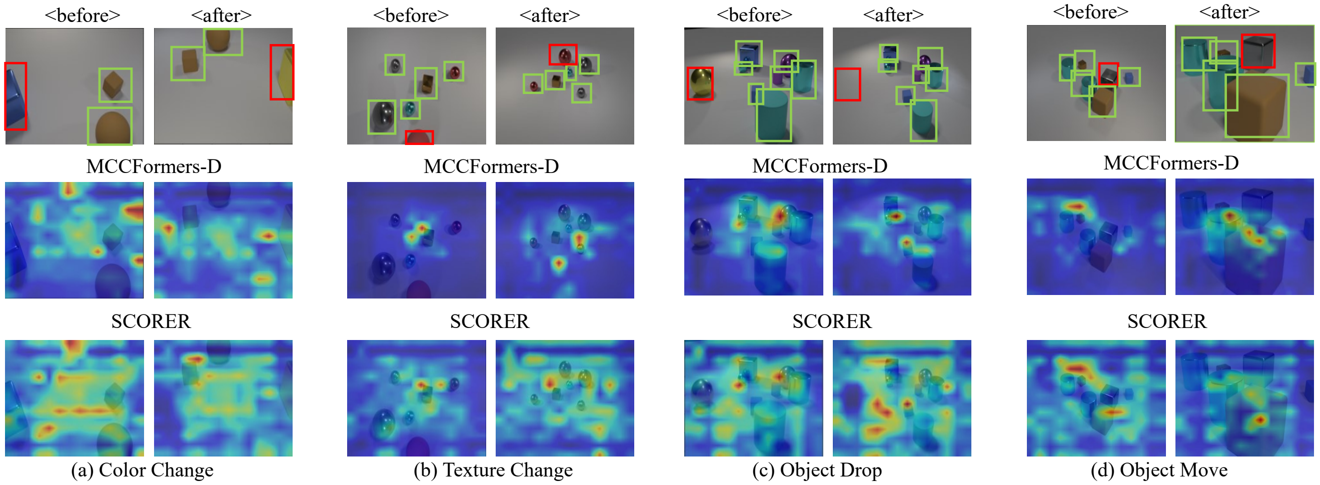
In Figure 6 - 9, we visualize the alignment between unchanged objects under different change types on CLEVR-Change, CLEVR-DC, Image Editing Request, and Spot-the-Diff, respectively. The compared method is MCCFormers-D that is a state-of-the-art method based on transformer. To fully match cross-view images, both SCORER and MCCFormers-D respectively use one image to query the shared objects on the other one, so obtaining two attention maps about cross-view alignment. We find that when directly matching two image features, MCCFormers-D mainly attends to some salient objects. Instead, our SCORER first learns two view-invariant image representations in a self-supervised way, by maximizing their cross-view contrastive alignment. Based on these, SCORER can better align and reconstruct the representations of unchanged objects, so as to facilitate subsequent difference representation learning.

In Figure 10 - 13, we visualize the generated captions along with the attention weight at each word under different change types on CLEVR-Change, CLEVR-DC, Image Editing Request, and Spot-the-Diff, respectively. When predicting the next word, the decoder uses generated words to compute attention over the learned difference representation, which yields a single attention map about cross-modal alignment. We interpolate it on each image to show the localization of before- and after-changed object during word generation. When the attention weight is higher, the localized region is brighter. We observe that when generating the words about the changed object or its referent, SCORER+CBR can adaptively attend to the corresponding regions. This superiority mainly results from the facts that 1) SCORER learns two view-invariant image representations for reconstructing the representations of unchanged objects, so as to learn a stable difference representation for caption generation; 2) cross-modal backward reasoning can improve the quality of generated captions by enforcing the caption to be informative about the learned difference.



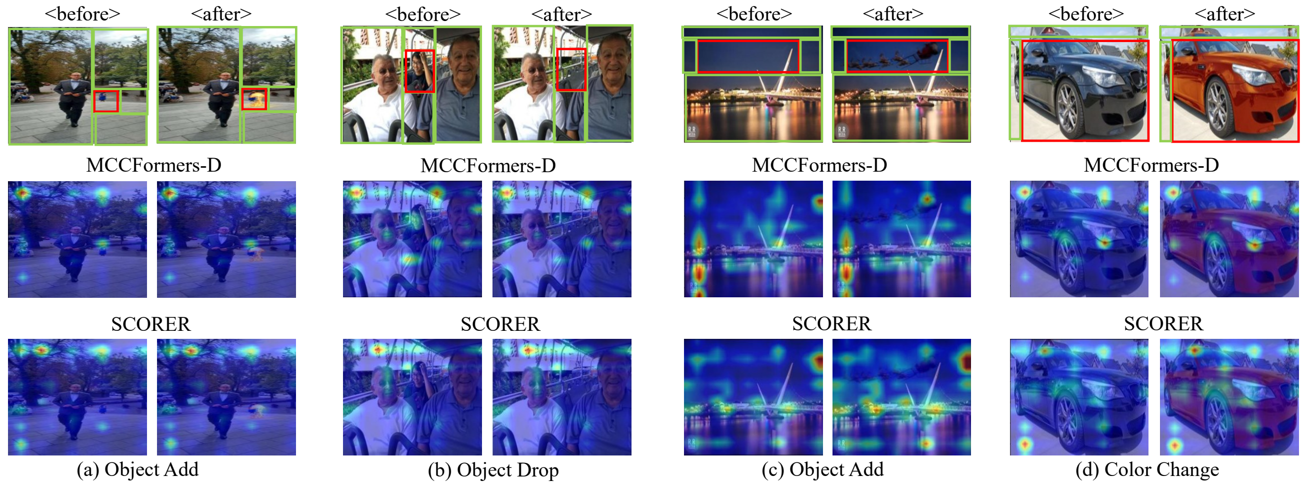
(a) Color Change (b) Texture Change (c) Object Drop (d) Object Move

Figure 6. Visualization of the alignment of unchanged objects on CLEVR-Change, computed by MCCFormers-D and our SCORER.



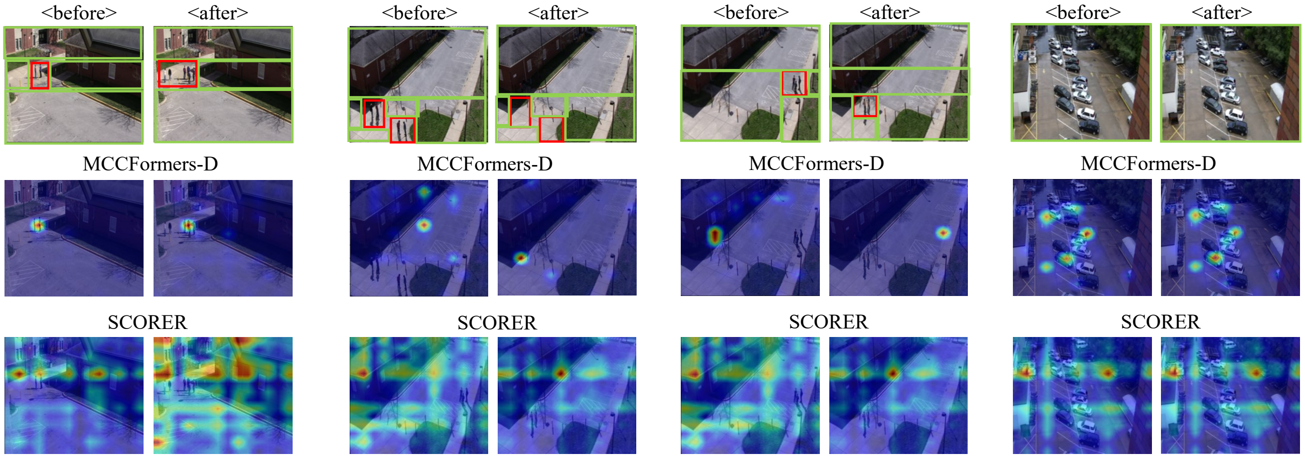
(a) Color Change (b) Texture Change (c) Object Drop (d) Object Move

Figure 7. Visualization of the alignment of unchanged objects on CLEVR-DC, computed by MCCFormers-D and our SCORER.



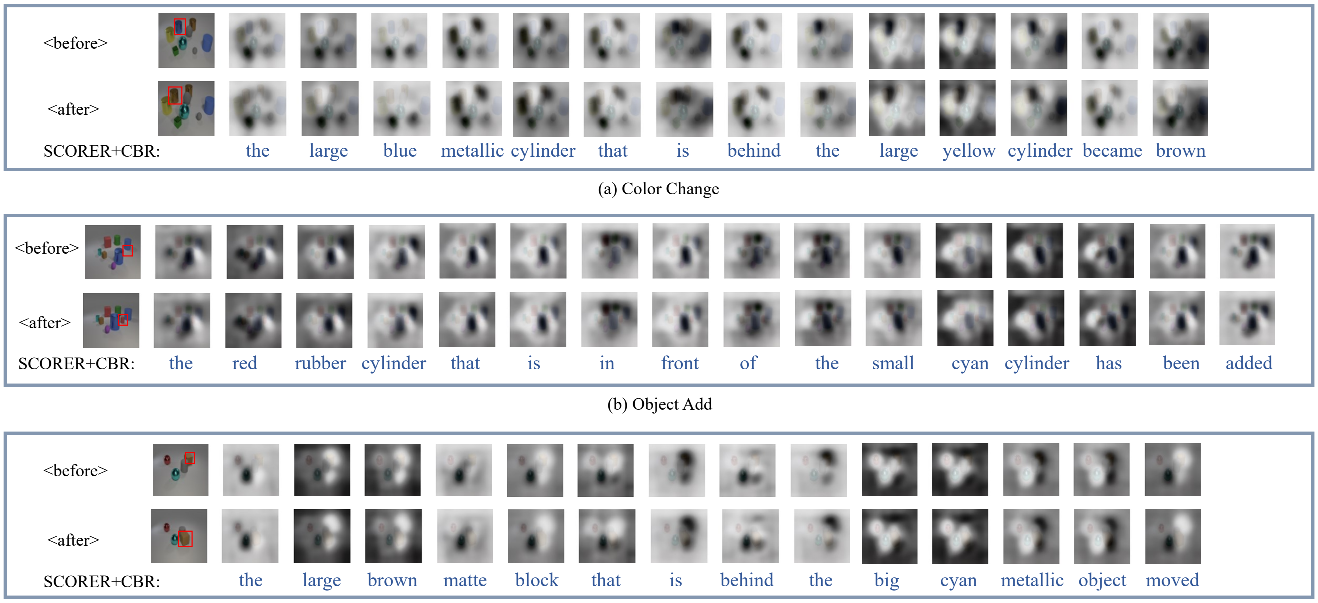
(a) Object Add (b) Object Drop (c) Object Add (d) Color Change

Figure 8. Visualization of the alignment of unchanged objects on Image Editing Request, computed by MCCFormers-D and our SCORER.



(a) Object Add (b) Object Drop (c) Object Move (d) No Change

Figure 9. Visualization of the alignment of unchanged objects on Spot-the-Diff, computed by MCCFormers-D and our SCORER.

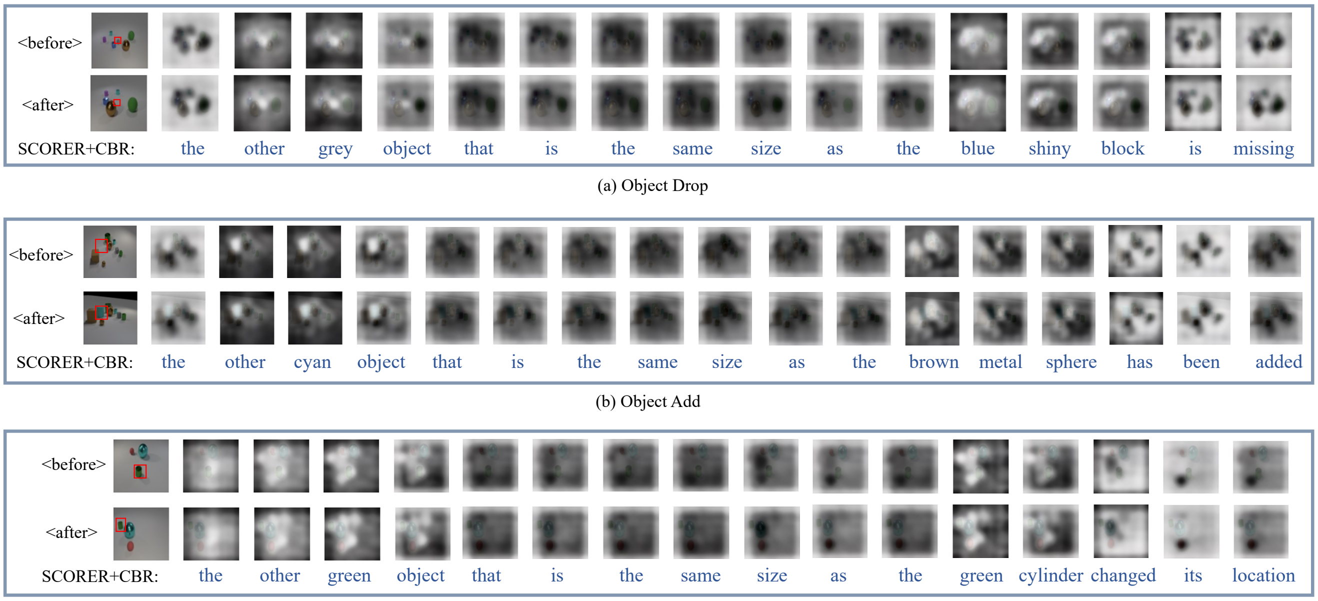

<before>
<after>
SCORER+CBR:
the large blue metallic cylinder that is behind the large yellow cylinder became brown
SCORER+CBR:
the red rubber cylinder that is in front of the small cyan cylinder has been added
SCORER+CBR:
the large brown matte block that is behind the big cyan metallic object moved

(a) Color Change

(b) Object Add

(c) Object Move

Figure 10. Three cases about “Color Change”, “Add”, and “Move” from CLEVR-Change, where the generated captions along with the attention weight at each word are visualized.

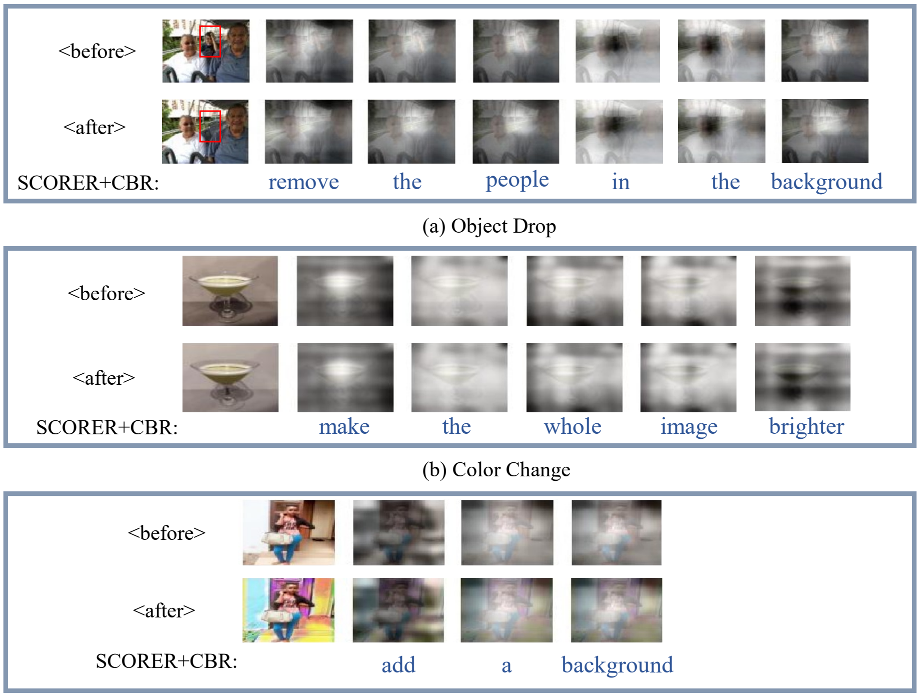

<before>
<after>
SCORER+CBR:
the other grey object that is the same size as the blue shiny block is missing
<before>
<after>
SCORER+CBR:
the other cyan object that is the same size as the brown metal sphere has been added
<before>
<after>
SCORER+CBR:
the other green object that is the same size as the green cylinder changed its location

(a) Object Drop

(b) Object Add

(c) Object Move

Figure 11. Three cases about “Drop”, “Add”, and “Move” from CLEVR-DC, where the generated captions along with the attention weight at each word are visualized.

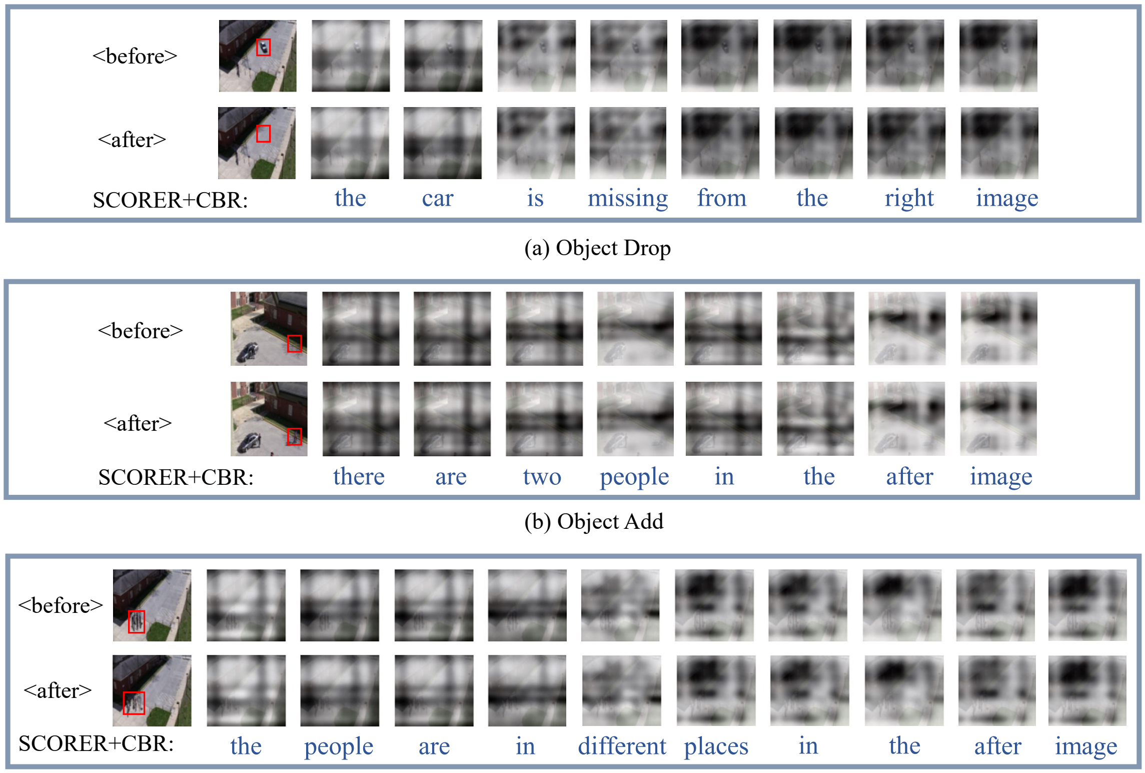


(a) Object Drop

(b) Color Change

(c) Object Add

Figure 12. Three cases about “Drop”, “Color Change”, and “Add” from Image Editing Request, where the generated captions along with the attention weight at each word are visualized.


<before>
<after>
SCORER+CBR:
the car is missing from the right image
<before>
<after>
SCORER+CBR:
there are two people in the after image
<before>
<after>
SCORER+CBR:
the people are in different places in the after image

(a) Object Drop

(b) Object Add

(c) Object Move

Figure 13. Three cases about “Drop”, “Add”, and “Move” from Spot-the-Diff, where the generated captions along with the attention weight at each word are visualized.