Self-supervised Cross-view Representation Reconstruction for Change Captioning

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

Abstract

Change captioning aims to describe the difference between a pair of similar images. Its key challenge is how to learn a stable difference representation under pseudo changes caused by viewpoint change. In this paper, we address this by proposing a self-supervised cross-view representation reconstruction (SCORER) network. Concretely, we first design a multi-head token-wise matching to model relationships between cross-view features from similar/dissimilar images. Then, by maximizing cross-view contrastive alignment of two similar images, SCORER learns two view-invariant image representations in a self-supervised way. Based on these, we reconstruct the representations of unchanged objects by cross-attention, thus learning a stable difference representation for caption generation. Further, we devise a cross-modal backward reasoning to improve the quality of caption. This module reversely models a “hallucination” representation with the caption and “before” representation. By pushing it closer to the “after” representation, we enforce the caption to be informative about the difference in a self-supervised manner. Extensive experiments show our method achieves the state-of-the-art results on four datasets. The code is available at https://github.com/tuyunbin/SCORER.

1. Introduction

Change captioning is a new task of vision and language, which requires not only understanding the contents of two similar images, but also describing their difference with natural language. In real world, this task brings a variety of applications, such as generating elaborated reports about monitored facilities [8, 10] and pathological changes [18, 14].

While single-image captioning is already regarded as a very challenging task, change captioning carries additional difficulties. Simply locating inconspicuous differences is one such challenge (Fig. 1 (a) (b)). Further, in a dynamic environment, it is common to acquire two images under different viewpoints, which leads to pseudo changes about objects’ scale and location (Fig. 1 (c) (d)). As such, change
captioning needs to characterize the real change while resisting pseudo changes. To locate change, the most intuitive way is to subtract two images [22, 7], but this risks computing difference features with noise if two images are unaligned [31]. Recently, researchers [25] find that same objects from different viewpoints would have similar features, so they match object features between two images to predict difference features. This paradigm has been followed by some of the recent works [11, 24, 38, 31, 30].

Despite the progress, current match-based methods suffer from learning stable difference features under pseudo changes. In detail, the matching is directly modeled between two image features, usually by cross-attention. However, the features of corresponding objects might shift under pseudo change. This case is more severe under drastic viewpoint changes (Fig. 1 (d)). Such feature shift appearing in most objects would overwhelm the local feature change, thus making it less effective to directly match two images.

For this challenge, we have two new observations. (1) While the feature difference might be ignored between a pair of similar images, it is hard to be overwhelmed between two images from different pairs. As such, contrastive difference learning between similar/dissimilar images can help the model focus more on the change of feature and resist feature shift. (2) Pseudo changes are essentially different distortions of objects, so they just construct cross-view comparison between two similar images, rather than affecting their similarity. Motivated by these, we study cross-view feature matching between similar/dissimilar images, and maximize the alignment of similar ones, so as to learn two view-invariant image representations. Based on these, we can reconstruct the representations of unchanged objects and learn a stable difference representation.

In this paper, we tackle the above challenge with a novel Self-supervised CrOss-view REpresentation Reconstruction (SCORER) network, which learns a stable difference representation while resisting pseudo changes for caption generation. Concretely, given two similar images, we first devise a multi-head token-wise matching (MTM) to model relationships between cross-view features from similar/dissimilar images, via fully interacting different feature subspaces. Then, by maximizing cross-view contrastive alignment of the given image pair, SCORER learns their representations that are invariant to pseudo changes in a self-supervised way. Based on these, SCORER mines their reliable common features by cross-attention, so as to reconstruct the representations of unchanged objects. Next, we fuse the representations into two images to highlight the unchanged objects and implicitly infer the difference. Through this manner, we can obtain the difference representation that not only captures the change, but also conserves referent information, thus generating a high-level linguistic sentence with a transformer decoder.

To improve the quality of sentence, we further design a cross-modal backward reasoning (CBR) module. CBR first reversely produces a “hallucination” representation with the full representations of sentence and “before” image, where the “hallucination” is modeled based on the viewpoint of “before”. Then, we push it closer to the “after” representation by maximizing their cross-view contrastive alignment. Through this self-supervised manner, we ensure that the generated sentence is informative about the difference.

Our key contributions are: (1) We propose SCORER to learn two view-invariant image representations for reconstructing the representations of unchanged objects, so as to model a stable difference representation under pseudo changes. (2) We devise MTM to model relationships between cross-view images by fully interacting their different feature subspaces, which plays a critical role in view-invariant representation learning. (3) We design CBR to improve captioning quality by enforcing the generated caption is informative about the difference. (4) Our method performs favorably against the state-of-the-art methods on four public datasets with different change scenarios.

2. Related Work

Change Captioning is a new task in vision-language understanding and generation [13, 19, 17, 29, 5, 35]. The pioneer works [10, 27] describe the difference between two aligned images (Fig. 1 (a) (b)). Since there usually exist viewpoint changes in a dynamic environment, recent works [22, 11] collect two datasets to simulate moderate changes (Fig. 1 (c)) and extreme viewpoint changes (Fig. 1 (d)). To describe the difference under viewpoint changes, previous works [22, 15] compute the difference by direct subtraction, which could compute difference with noise [25]. Recent methods [11, 24, 31, 30, 28, 39] directly match two images to predict difference features. However, due to the influence of pseudo changes, these methods are hard to learn stable difference features. In contrast, our SCORER first learns two view-invariant image representations by maximizing their cross-view contrastive alignment. Then, it mines their common features to reconstruct the representations of unchanged objects, thus learning a stable difference representation for caption generation. We note that the latest work [38] pre-trains the model with three self-supervised tasks, in order to improve cross-modal alignment. Different from it, we enforce the cross-modal alignment by implementing cross-modal backward reasoning in a self-supervised way. Meanwhile, our overall architecture is trained in an end-to-end manner, which improves the training efficiency.

Token-wise Matching has been used in latest image/video retrieval works [37, 36] to compute cross-modal interaction between image/video and text features. However, since pseudo changes would induce feature shift between object pairs, it is insufficient to only match cross-
view features at token level. Hence, we further design a multi-head token-wise matching for finer-level interaction between different feature subspaces of cross-view images. This is key to learn the view-invariant representations.

**Cross-modal Consistency Constraint** is to verify the quality of caption by using it and “before” image to rebuild “after” image. This idea has been tried by the latest works [7, 11]. However, both works only enforce the consistency among the caption, the changed object in “before” and “after” images, while ignoring the constraint for referents. If the changed object is similar to other objects (Fig. 1 (c) (d)), describing both the change and its referent is helpful to convey accurate change information. Considering this, we perform backward reasoning with the full representations of “before” and “after” images, which helps generate a high-level sentence about the change and its referent.

### 3. Methodology

As shown in Fig. 2, our method consists of four parts: (1) A pre-trained CNN encodes a pair of cross-view images into two representations. (2) The proposed SCORER learns two view-invariant representations to reconstruct the representations of unchanged objects and model the difference representation. (3) A transformer decoder translates the difference representation into a high-level linguistic sentence. (4) The proposed CBR improves the quality of sentence via enforcing it to be informative about the difference.

#### 3.1. Cross-view Image Pair Encoding

Formally, given a pair of images “before” $I_{bef}$ and “after” $I_{aft}$, we utilize a pre-trained CNN model to extract their grid features, denoted as $X_{bef}$ and $X_{aft}$, where $X \in \mathbb{R}^{C \times H \times W}$. $C$, $H$, $W$ indicate the number of channels, height, and width. We first project both representations into a low-dimensional embedding space of $\mathbb{R}^D$:

$$X_o = \text{conv}_2(X_o) + \text{pos}(X_o),$$  

(1)

where $o \in \{bef, aft\}$. $\text{conv}_2$ denotes a 2D-convolutional layer; $\text{pos}$ is a learnable position embedding layer.

#### 3.2. Self-supervised Cross-view Representation Reconstruction

The core module of SCORER is the multi-head token-wise matching (MTM). MTM aims to model relationships between cross-view images by performing fine-grained interaction between different feature subspaces, which plays a key role in view-invariant representation learning. In the following, we first elaborate MTM and then describe how to use it for view-invariant representation learning. Finally, we introduce how to reconstruct the representations of unchanged objects for difference representation learning.

##### 3.2.1 Multi-head Token-wise Matching

We first introduce the single-head token-wise matching (TM) and then extend it into the multi-head version. Formally, given a query $Q \in \mathbb{R}^{N \times D}$ and a key $K \in \mathbb{R}^{N \times D}$, we first compute the similarity of $i$-th query token with all key tokens and select the maximum one as its token-wise maximum similarity with $K$. Then, we perform average pooling over the token-wise maximum similarity of all query tokens to obtain the similarity of $Q$ to $K$. By analogy, we compute the average token-wise maximum similarity of $K$ to...
negatives in this batch. First, we reshape tively. Next, we maximize cross-view contrastive alignment trix) of “before” to “after” and “after” to “before”, respec-

resentations of each image are distilled from the other one, 

based on the two view-invariant representations

Further, we extend TM into a multi-head version to jointly

target words are calculated by:

In a training batch, we sample \( B \) image pairs of “before” and “after”. For \( k \)-th “before” image \( X_k^b \), \( k \)-th “after” image \( X_k^a \) is its positive, while other “after” images will be the negatives in this batch. First, we reshape \( \tilde{X} \in \mathbb{R}^{D \times H \times W} \) to \( \tilde{X} \in \mathbb{R}^{N \times D} \), where \( N = HW \) denotes the number of features. Then, we use MTM to compute similarity (\( B \times B \) matrix) of “before” to “after” and “after” to “before”, respectively. Next, we maximize cross-view contrastive alignment between \( X_k^b \) and \( X_k^a \) while minimizing the alignment of non-similar images, by the InfoNCE loss [20]:

where \( \tau \) is the temperature hyper-parameter. In this self-

supervised way, we can make the representations of \( \tilde{X}_{bef} \) and \( \tilde{X}_{aft} \) invariant to pseudo changes, so as to facilitate the following cross-view representation reconstruction.

Based on the two view-invariant representations \( \tilde{X}_{bef} \) and \( \tilde{X}_{aft} \), we use a multi-head cross-attention (MHCA) [33] to mine their common features for reconstructing the representations of unchanged objects in each image. Here, representation reconstruction indicates that the unchanged representations of each image are distilled from the other one, e.g., the unchanged representations of \( \tilde{X}_{bef} \) are computed by transferring similar features on \( \tilde{X}_{aft} \) back to the corresponding positions on \( \tilde{X}_{bef} \). In this way, we reconstruct the unchanged representations for each image, respectively:

Then, instead of subtracting them from image representations [25, 31, 30], which leads to information (e.g., referents) loss, we integrate them into image representations to highlight the unchanged objects and deduce the difference information, so as to learn the stable difference representation in each image:

Herein, \( o \in \{bef, aft\} \) and LN is short for LayerNorm [2]. Finally, we obtain the difference representation between two images by fusing \( \tilde{X}^e_{bef} \) and \( \tilde{X}^e_{aft} \), which is implemented by a fully-connected layer with the ReLU function:

where \([\cdot]\) is a concatenation operation.

After leaning \( \tilde{X}_c \in \mathbb{R}^{N \times D} \), we use a transformer decoder [33] to translate it into a sentence. First, the multi-head self-attention takes the word features \( E[W] = \{E[w_1], ..., E[w_m]\} \) (ground-truth words during training, predicted words during inference) as inputs and computes a set of intra-relation embedded word features, denoted as \( \hat{E}[W] \). Then, the decoder utilizes \( \hat{E}[W] \) to query the most related features \( \hat{H} \) from \( \tilde{X}_c \) via the multi-head cross-attention. Afterward, the \( \hat{H} \) is passed to a feed-forward network to obtain an enhanced representation \( \hat{H} \). Finally, the probability distributions of target words are calculated by:

where \( W_h \in \mathbb{R}^{D \times U} \) and \( b_h \in \mathbb{R}^U \) are the parameters to be learned; \( U \) is the dimension of vocabulary size.

To improve the quality of generated sentence, we devise the CBR to first reversely model a “hallucination” representation with the sentence and “before” image. Then, we push it closer to the “after” representation to enforce the sentence to be informative about the difference. Concretely, we first fuse \( \tilde{H} \in \mathbb{R}^{m \times D} \) by the mean-pooling operation to obtain a sentence feature \( \tilde{T} \). Then, we broadcast \( \tilde{T} \in \mathbb{R}^{D \times H \times W} \) and concatenate it with \( \tilde{X}_{bef} \), so as to obtain the “hallucination” \( \tilde{X}_{hal} \):

where \( \tilde{X}_{hal} \in \mathbb{R}^{D \times H \times W} \).
\( \hat{X}_{hal} \) and \( \hat{X}_{bef} \) are kept as the same shape to ensure that spatial information is not collapsed. Next, we capture the relationships between different locations in \( \hat{X}_{hal} \) based on the multi-head self-attention (MHSA), which is essential for backward reasoning and computed by:

\[
\hat{X}_{hal} = \text{conv}_2[\text{MHSA} (\hat{X}_{hal}, \hat{X}_{hal}, \hat{X}_{hal})],
\]

(10)

Since the “hallucination” representation is produced based on the viewpoint of “before” representation, it is less effective to directly match it with the “after” representation.

To this end, we sample unrelated representations of “hallucination” and “after” from different pairs, which are as erroneous candidates for CBR. Similarly, in each batch, for \( k \)-th “hallucination” \( \hat{X}^h_k \), \( k \)-th “after” \( \hat{X}^a_k \) is its positive, while the other “after” images will be the negatives. Also, we use MTM to capture relationships between positive/negative pairs. Subsequently, we maximize cross-view contrastive alignment of positive pairs by the InfoNCE loss [20], which is similar to Eq. (4):

\[
\mathcal{L}_{cm} = \frac{1}{2}(\mathcal{L}_{h2a} + \mathcal{L}_{a2h}).
\]

(11)

Through this self-supervised manner, we make the sentence sufficiently describe the difference information.

3.5. Joint Training

The proposed overall network is trained in an end-to-end manner by maximizing the likelihood of the observed word sequence. Given the ground-truth words \((w^*_1, \ldots, w^*_m)\), we minimize the negative log-likelihood loss:

\[
\mathcal{L}_{cap}(\theta) = -\sum_{t=1}^{m} \log p_{\theta}(w^*_t \mid w^*_{<t}),
\]

(12)

where \( p_{\theta}(w^*_t \mid w^*_{<t}) \) is computed by Eq. (8), and \( \theta \) are the parameters of the network. Besides, the network is self-supervised by the losses of two contrastive alignments. Hence, the final loss function is optimized as follows:

\[
\mathcal{L} = \mathcal{L}_{cap} + \lambda_u\mathcal{L}_{cv} + \lambda_m\mathcal{L}_{cm},
\]

(13)

where \( \lambda_u \) and \( \lambda_m \) are the trade-off parameters, which are discussed in the supplementary material.

4. Experiments

4.1. Datasets

**CLEVR-Change** is a large-scale dataset [22] with moderate viewpoint change. It has 79,606 image pairs, including five change types, i.e., “Color”, “Texture”, “Add”, “Drop”, and “Move”. We use the official split with 67,660 for training, 3,976 for validation and 7,970 for testing.

**CLEVR-DC** is a large-scale dataset [11] with extreme viewpoint shift. It includes 48,000 pairs with same change types as CLEVR-Change. We use the official split with 85% for training, 5% for validation, and 10% for testing.

**Image Editing Request** dataset [27] includes 3,939 aligned image pairs with 5,695 editing instructions. We use the official split with 3,061 image pairs for training, 383 for validation, and 495 for testing.

**Spot-the-Diff** dataset [10] includes 13,192 aligned image pairs from surveillance cameras. Following SOTA methods, we mainly evaluate our model in a single change setting. Based on the official split, the dataset is split into training, validation, and testing with a ratio of 8:1:1.

4.2. Evaluation Metrics

Following the current state-of-the-art methods, five metrics are used to evaluate the generated sentences, i.e., BLEU-4 (B) [21], METEOR (M) [3], ROUGE-L (R) [16], CIDEr (C) [34], and SPICE (S) [1]. The results are computed based on the Microsoft COCO evaluation server [4].

4.3. Implementation Details

For a fair comparison, we follow the SOTA methods to use a pre-trained ResNet-101 [6] to extract grid features of an image pair, with the dimension of 1024 × 14 × 14. We first project these features into a lower dimension of 512. The proper head and layer numbers of SCORER are discussed below. The head and layer numbers in the decoder are set to 8 and 2 on the four datasets. During training, We use Adam optimizer [12] to minimize the negative log-likelihood loss of Eq. (13). During inference, the greedy decoding strategy is used to generate captions. Both training and inference are implemented with PyTorch [23] on an RTX 3090 GPU. More implementation details are described in the supplementary material.

4.4. Performance Comparison

4.4.1 Results on the CLEVR-Change Dataset.

We compare with the state-of-the-art methods in: 1) total performance under both semantic and pseudo changes; 2) semantic change; 3) different change types. The comparison methods are categorized into 1) end-to-end training: DUDA [22], DUDA+ [7], RNet+SSP [31], VACC [11], SRDRL+AVS [32], SGCC [15], MCCFormer-D [24], IFDC [9], BDLSR [26], NCT [30], and VARD-Trans [28]; 2) reinforcement learning: M-VAM+RAF [25]; 3) pre-training: PCL w/ pre-training [38].

In Table 1, our method achieves the best results on all metrics against the end-to-end training methods. Besides, our method performs much better than these two methods augmented by pre-training and reinforcement learning.
Table 1. Comparison with the state-of-the-art methods on CLEVR-Change under the settings of total performance and semantic change.

<table>
<thead>
<tr>
<th>Method</th>
<th>B</th>
<th>M</th>
<th>R</th>
<th>C</th>
<th>S</th>
<th>B</th>
<th>M</th>
<th>R</th>
<th>C</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCL w/ Pre-training (AAAI 2022) [38]</td>
<td>51.2</td>
<td>36.2</td>
<td>71.7</td>
<td><strong>128.9</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>M-VAM+RAF (ECCV 2020) [25]</td>
<td>51.3</td>
<td>37.8</td>
<td>70.4</td>
<td>115.8</td>
<td>30.7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DUDA (ICCV 2019) [22]</td>
<td>49.3</td>
<td>39.7</td>
<td>71.5</td>
<td>123.0</td>
<td>29.6</td>
<td>52.7</td>
<td>36.2</td>
<td>69.8</td>
<td>116.6</td>
<td>30.3</td>
</tr>
<tr>
<td>M-VAM+RAF+CC (ECCV 2020) [25]</td>
<td>51.3</td>
<td>37.8</td>
<td>70.4</td>
<td>115.8</td>
<td>30.7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>R^3Net+SSP (EMNLP 2021) [31]</td>
<td>54.7</td>
<td>39.8</td>
<td>73.1</td>
<td>123.0</td>
<td>32.6</td>
<td>52.7</td>
<td>36.2</td>
<td>69.8</td>
<td>116.6</td>
<td>30.3</td>
</tr>
<tr>
<td>VACC (ICCV 2021) [11]</td>
<td>52.4</td>
<td>37.5</td>
<td>-</td>
<td>114.2</td>
<td>31.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SGCC (ACM MM 2021) [15]</td>
<td>51.1</td>
<td>40.6</td>
<td>73.9</td>
<td>121.8</td>
<td>32.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SRDRL+AVS (ACL 2021) [32]</td>
<td>54.9</td>
<td>40.2</td>
<td>73.3</td>
<td>122.2</td>
<td>32.9</td>
<td>52.7</td>
<td>36.4</td>
<td>69.7</td>
<td>114.2</td>
<td>30.8</td>
</tr>
<tr>
<td>MCCFormers-D (ICCV 2021) [24]</td>
<td>52.4</td>
<td>38.3</td>
<td>-</td>
<td>121.6</td>
<td>26.8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>IFDC (TMM 2022) [9]</td>
<td>49.2</td>
<td>32.5</td>
<td>69.1</td>
<td>118.7</td>
<td>-</td>
<td>47.2</td>
<td>39.3</td>
<td>63.7</td>
<td>105.4</td>
<td>-</td>
</tr>
<tr>
<td>NCT (TMM 2023) [30]</td>
<td>55.1</td>
<td>40.2</td>
<td>73.8</td>
<td>124.1</td>
<td>32.9</td>
<td>53.1</td>
<td>36.5</td>
<td>70.7</td>
<td>118.4</td>
<td>30.9</td>
</tr>
<tr>
<td>SCORER (Ours)</td>
<td>55.8</td>
<td>40.8</td>
<td>74.0</td>
<td>126.0</td>
<td>33.0</td>
<td>54.1</td>
<td>37.4</td>
<td>71.5</td>
<td>122.0</td>
<td>31.2</td>
</tr>
<tr>
<td>SCORER+CBR (Ours)</td>
<td>56.3</td>
<td>41.2</td>
<td>74.5</td>
<td>126.8</td>
<td>33.3</td>
<td><strong>54.4</strong></td>
<td><strong>37.6</strong></td>
<td><strong>71.7</strong></td>
<td>122.4</td>
<td>31.6</td>
</tr>
</tbody>
</table>

Table 2. A detailed breakdown of evaluation on CLEVR-Change with different change types: “(CL) Color”, “(T) Textur”, “(A) Add”, “(D) Drop”, and “(MV) Move”. PT is short for pre-training.

<table>
<thead>
<tr>
<th>Method</th>
<th>CL</th>
<th>T</th>
<th>A</th>
<th>D</th>
<th>MV</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCL w/ PT</td>
<td>131.2</td>
<td>101.1</td>
<td><strong>133.3</strong></td>
<td>116.5</td>
<td>81.7</td>
</tr>
<tr>
<td>M-VAM+RAF</td>
<td>122.1</td>
<td>98.7</td>
<td>126.3</td>
<td>115.8</td>
<td>82.0</td>
</tr>
<tr>
<td>DUDA</td>
<td>120.4</td>
<td>86.7</td>
<td>108.2</td>
<td>103.4</td>
<td>56.4</td>
</tr>
<tr>
<td>DUDA+</td>
<td>120.8</td>
<td>89.9</td>
<td>119.8</td>
<td>123.4</td>
<td>62.1</td>
</tr>
<tr>
<td>R^3Net+SSP</td>
<td>139.2</td>
<td>123.5</td>
<td>122.7</td>
<td>121.9</td>
<td>88.1</td>
</tr>
<tr>
<td>SRDRL+AVS</td>
<td>136.1</td>
<td>122.7</td>
<td>121.0</td>
<td>126.0</td>
<td>78.9</td>
</tr>
<tr>
<td>BDLSCR</td>
<td>136.1</td>
<td>122.7</td>
<td>121.0</td>
<td>126.0</td>
<td>78.9</td>
</tr>
<tr>
<td>IFDC</td>
<td>133.2</td>
<td>99.1</td>
<td>128.2</td>
<td>118.5</td>
<td>82.1</td>
</tr>
<tr>
<td>NCT</td>
<td>140.2</td>
<td>128.8</td>
<td>128.4</td>
<td>129.0</td>
<td>86.0</td>
</tr>
<tr>
<td>SCORER</td>
<td>143.2</td>
<td>135.2</td>
<td>129.4</td>
<td>132.6</td>
<td>91.6</td>
</tr>
<tr>
<td>SCORER+CBR</td>
<td><strong>146.2</strong></td>
<td><strong>133.7</strong></td>
<td><strong>131.1</strong></td>
<td><strong>133.9</strong></td>
<td><strong>92.2</strong></td>
</tr>
</tbody>
</table>

We note that SCORER outperforms MCCFormers-D by a large margin. MCCFormers-D is a classic match-based method that directly correlates two image representations to learn a difference representation, which is then fed into a transformer decoder for caption generation. Different from it, our SCORER first learns two view-invariant image representations by maximizing their cross-view contrastive alignment. Then, SCORER reconstructs the representations of unchanged objects, so as to learn a stable difference representation under pseudo changes for caption generation.

In Table 2, under the detailed change types, our method surpasses the current methods by a large margin in almost every category. Under the most difficult type “Move”, our SCORER+CBR achieves the relative improvement of 4.7% against R^3Net+SSP. This validates the necessary of view-invariant representation learning. Moreover, under different settings, CBR helps yield an extra performance boost, which shows it does improve captioning quality.

4.4.2 Results on the CLEVR-DC Dataset

On CLEVR-DC with extreme viewpoint changes, we compare SCORER/SCORER+CBR with several state-of-the-art methods: DUDA/DUDA+ [22], M-VAM/M-VAM+ [25], VA/VACC [11], MCCFormers-D [24], NCT [30], and VARD-Trans [28]. For fair-comparison, we compare them based on the usage of cross-modal consistency constraint. We implement MCCFormers-D based on the released code on CLEVR-DC and Image Editing Request datasets.

The results are shown in Table 3. Our SCORER achieves the best results on most metrics. This benefits from learning two view-invariant representations to reconstruct representations of unchanged objects, thus learning a stable difference representation under extreme viewpoint changes. When we implement CBR, the performance of SCORER+CBR is further boosted, especially achieving 16.7% improvement against VACC on CIDEr. This shows that our CBR can calibrate the model to generate a linguistic sentence describing the change and its referent.

4.4.3 Results on the Image Editing Request Dataset

To validate the generalization of our method, we conduct the experiment on a challenging dataset of Image Editing Request (IER). We compare with the following SOTA methods: DUDA [22], Dyn rel-att [27], MCCFormers-D [24], BDLSCR [26], NCT [30], and VARD-Trans [28].
Table 3. Comparison with the SOTA methods on CLEVR-DC.

<table>
<thead>
<tr>
<th>Method</th>
<th>B</th>
<th>M</th>
<th>C</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUDA [22]</td>
<td>40.3</td>
<td>27.1</td>
<td>56.7</td>
<td>16.1</td>
</tr>
<tr>
<td>M-VAM [25]</td>
<td>40.9</td>
<td>27.1</td>
<td>60.1</td>
<td>15.8</td>
</tr>
<tr>
<td>VA [11]</td>
<td>44.5</td>
<td>29.2</td>
<td>70.0</td>
<td>15.1</td>
</tr>
<tr>
<td>MCCFormers-D [24]</td>
<td>46.9</td>
<td>31.7</td>
<td>76.9</td>
<td>14.6</td>
</tr>
<tr>
<td>NCT [30]</td>
<td>47.5</td>
<td>32.5</td>
<td>77.2</td>
<td>15.6</td>
</tr>
<tr>
<td>VARD-Trans [28]</td>
<td>48.3</td>
<td>32.4</td>
<td>76.6</td>
<td>15.4</td>
</tr>
<tr>
<td>SCORER</td>
<td><strong>49.5</strong></td>
<td><strong>33.4</strong></td>
<td><strong>82.4</strong></td>
<td><strong>15.8</strong></td>
</tr>
<tr>
<td>M-V AM+CC [25]</td>
<td>41.7</td>
<td>27.5</td>
<td>62.0</td>
<td>16.4</td>
</tr>
<tr>
<td>V ACC [11]</td>
<td>44.5</td>
<td>29.2</td>
<td>70.0</td>
<td>17.1</td>
</tr>
<tr>
<td>MCCFormers-D+CC</td>
<td>46.9</td>
<td>31.7</td>
<td>76.9</td>
<td>15.6</td>
</tr>
<tr>
<td>NCT+CC [30]</td>
<td>47.5</td>
<td>32.5</td>
<td>77.2</td>
<td>15.6</td>
</tr>
<tr>
<td>VARD-Trans+CC</td>
<td>48.3</td>
<td>32.4</td>
<td>76.6</td>
<td>15.4</td>
</tr>
<tr>
<td>SCORER+CBR</td>
<td><strong>49.5</strong></td>
<td><strong>33.4</strong></td>
<td><strong>82.4</strong></td>
<td><strong>15.8</strong></td>
</tr>
</tbody>
</table>

Table 4. Comparison with the SOTA methods on IER.

<table>
<thead>
<tr>
<th>Method</th>
<th>B</th>
<th>M</th>
<th>R</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUDA+ [7]</td>
<td>6.5</td>
<td>12.4</td>
<td>37.3</td>
<td>22.8</td>
</tr>
<tr>
<td>Dyn rel-att [27]</td>
<td>6.7</td>
<td>12.8</td>
<td>37.5</td>
<td>26.4</td>
</tr>
<tr>
<td>MCCFormers-D</td>
<td>8.3</td>
<td>14.3</td>
<td>39.2</td>
<td>30.2</td>
</tr>
<tr>
<td>BDLSCR [26]</td>
<td>6.9</td>
<td>14.6</td>
<td>38.5</td>
<td>27.7</td>
</tr>
<tr>
<td>NCT [30]</td>
<td>8.1</td>
<td>15.0</td>
<td>38.8</td>
<td>34.2</td>
</tr>
<tr>
<td>VARD-Trans [28]</td>
<td><strong>10.0</strong></td>
<td><strong>14.8</strong></td>
<td><strong>39.0</strong></td>
<td><strong>35.7</strong></td>
</tr>
<tr>
<td>SCORER</td>
<td>9.6</td>
<td>14.6</td>
<td>39.5</td>
<td>31.0</td>
</tr>
<tr>
<td>SCORER+CBR</td>
<td><strong>10.0</strong></td>
<td><strong>15.0</strong></td>
<td><strong>39.6</strong></td>
<td><strong>33.4</strong></td>
</tr>
</tbody>
</table>

Table 5. Comparison with the SOTA methods on Spot-the-Diff.

<table>
<thead>
<tr>
<th>Method</th>
<th>B</th>
<th>M</th>
<th>R</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subtraction</td>
<td>53.3</td>
<td>38.8</td>
<td>72.1</td>
<td>119.7</td>
</tr>
<tr>
<td>RR</td>
<td>55.1</td>
<td>40.5</td>
<td>73.6</td>
<td>123.8</td>
</tr>
<tr>
<td>SCORER</td>
<td>55.8</td>
<td>40.8</td>
<td>74.0</td>
<td>126.0</td>
</tr>
<tr>
<td>SCORER+CBR</td>
<td><strong>56.3</strong></td>
<td><strong>41.2</strong></td>
<td><strong>74.5</strong></td>
<td><strong>126.8</strong></td>
</tr>
</tbody>
</table>

Table 6. Ablation study of each module on CLEVR-Change.

<table>
<thead>
<tr>
<th>Ablation</th>
<th>B</th>
<th>M</th>
<th>R</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subtraction</td>
<td>53.3</td>
<td>38.8</td>
<td>72.1</td>
<td>119.7</td>
</tr>
<tr>
<td>RR</td>
<td>55.1</td>
<td>40.5</td>
<td>73.6</td>
<td>123.8</td>
</tr>
<tr>
<td>SCORER</td>
<td>55.8</td>
<td>40.8</td>
<td>74.0</td>
<td>126.0</td>
</tr>
<tr>
<td>SCORER+CBR</td>
<td><strong>56.3</strong></td>
<td><strong>41.2</strong></td>
<td><strong>74.5</strong></td>
<td><strong>126.8</strong></td>
</tr>
</tbody>
</table>

4.4.4 Results on the Spot-the-Diff Dataset

Table 4 shows SCORER+CBR outperforms the SOTA methods on most metrics. Especially on BLEU-4, SCORER+CBR obtains the relative improvement of 23.5% against the latest method NCT (TMM 2023). The edited objects are usually inconspicuous. This indicates that the proposed method can fully mine the common features by maximizing cross-view contrastive alignment between two images, so as to accurately describe which part of the “before” image has been edited. Further, the generated sentence is refined in the process of cross-modal backward reasoning.

4.5. Ablation Study and Analysis

Table 6 shows ablation study of each module under total performance. Subtraction indicates directly subtracting two images; RR means vanilla representation reconstruction. We find that RR is much better than Subtraction, showing that match-based strategy is more reliable than direct subtraction under pseudo changes. When we maximize cross-view contrastive alignment of two images, SCORER+CBR does not gain significant improvement.

In short, compared with the state-of-the-art methods in different change scenarios, our method achieves the impressive performance. The superiority mainly results from 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 generating a linguistic sentence; 2) CBR can further improve the quality of generated sentence.
Table 7. Ablation study on CLEVR-Change under the evaluation of semantic change and only pseudo change.

<table>
<thead>
<tr>
<th>Method</th>
<th>Semantic Change</th>
<th>Only Pseudo Change</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B</td>
<td>M</td>
</tr>
<tr>
<td>Subtraction</td>
<td>50.2</td>
<td>34.1</td>
</tr>
<tr>
<td>RR</td>
<td>53.3</td>
<td>37.1</td>
</tr>
<tr>
<td>SCORER</td>
<td>54.3</td>
<td>37.5</td>
</tr>
<tr>
<td>RR+CBR</td>
<td>54.1</td>
<td>37.4</td>
</tr>
<tr>
<td>SCORER+CBR</td>
<td>54.4</td>
<td>37.6</td>
</tr>
</tbody>
</table>

Table 3. Ablation study of MTM on four datasets.

![Ablation studies of MTM on four datasets.](image)

Figure 4. Effect of head number of SCORER on four datasets.

![Effect of head number of SCORER on four datasets.](image)

Figure 5. Effect of layer number of SCORER on four datasets.

![Effect of layer number of SCORER on four datasets.](image)

Besides, when we augment RR and SCORER with CBR, both RR+CBR and SCORER+CBR achieve better performances. This not only validates that CBR improves captioning quality, but also proves that CBR is generalizable.

Table 7 shows the ablation study of each module under semantic change and only pseudo change, separately. We can obtain observations similar to the total performance. Besides, we find that SCORER is much better than RR under semantic change, but under only pseudo change, SCORER brings less gain. This results from that in this case, the learned difference representation contains less information, making SCORER difficult to align it with words. By contrast, SCORER+CBR significantly improves RR on both settings, which shows that SCORER and CBR supplement each other. More ablation studies on the other datasets are in the supplementary material.

**Ablation Study of MTM.** Instead of using MTM to perform fine-grained matching between different feature subspaces of cross-view images, we use max/mean-pooling to obtain the global feature of each image and compute their similarity. Besides, we implement TM without multi-head operation. The results in Fig. 3 show that MTM achieves the best results, which demonstrates that it plays a critical role in view-invariant representation learning. Besides, only implementing token-wise matching is not better than simple mean-pooling. Our conjecture is that the changed object commonly appears in a local region with weak feature, so it is insufficient to reveal this slight difference by only interacting features at token level. As such, it is necessary to match two image features at finer level, i.e., subspace level.

**Effect of Head Number of SCORER.** We further investigate the effect of head number for SCORER, i.e., the head number of MTM and MHCA (Eq. (5)). The results are shown in Fig. 4. We find that the best results are achieved on the four datasets when setting the head number as 8.

**Effect of Layer Number of SCORER.** We investigate the effect of layer number for SCORER in Fig. 5. On four datasets, we find that increasing the layer number does not bring better performance, because deeper layers could result in the problem of over-fitting. Besides, the layer number is the deepest on Spot-the-Diff. Our conjecture is that objects have no good postures and background information is more complex in a surveillance scenario. As such, we empirically set proper layer number of 2, 1, 3, and 2 on four datasets.

### 4.6. Captioning and change localization results with varied viewpoints

To intuitively evaluate the efficacy of our method to handle pseudo changes, we show the captioning (Fig. 6 (a)) and change localization (Fig. 6 (b)) results of SCORER+CBR and SOTA method MCCFormers-D [24] with varied viewpoints. The amount of viewpoint change is measured by the IoUs of objects’ bounding boxes across an image pair (lower IoU means higher difficulty). For change localiza-
there are two people in the after image.

(b) Spot-the-Diff: Object Add

there are two people in the after image.

(c) Image Editing Request: Object Drop

Figure 7. Three cases from different scenarios, where the generated captions along with the attention weight at each word are visualized.

Figure 8. Visualization of the alignment of unchanged objects computed by MCCFormers-D [24] and our SCORER.

4.7. Qualitative Analysis

To intuitively evaluate our method, we conduct qualitative analysis on the four datasets. Fig. 7 illustrates three cases in different change scenarios. For each case, we visualize the generated caption along with the attention weight at each word. When the weight is higher, the region is brighter. We observe that when generating the words about the changed object or its referents, SCORER+CBR can adaptively locate the corresponding regions. In Fig. 8, we visualize the alignment between unchanged objects under different change scenes. The compared method is the SOTA method MCCFormers-D [24]. We implement it based on the released code. We find that when directly correlating two image features, MCCFormers-D only aligns salient objects between two images. Instead, our SCORER first learns two view-invariant representations in a self-supervised way. Based on these, SCORER can better align and reconstruct the representations of unchanged objects, so as to facilitate subsequent difference representation learning. More qualitative examples are shown in the supplementary material.

5. Conclusion

This paper proposes a novel SCORER to learn a stable difference representation while resisting pseudo changes. SCORER first learns two view-invariant image representations in a self-supervised way, by maximizing the cross-view contrastive alignment of two images. Based on these, SCORER mines their common features to reconstruct the representations of unchanged objects. This helps learn a stable difference representation for caption generation. Further, we design the CBR to improve captioning quality by enforcing the yielded caption is informative about the difference in a self-supervised manner. Extensive experiments show that our method achieves the state-of-the-art results on four public datasets with different change scenarios.

Acknowledgements

This work was supported by the National Key Research and Development Program of China under Grant (2018AAA0102000), National Nature Science Foundation of China (62322211, U21B2024, 61931008, 62071415, 62236008, U21B2038), Fundamental Research Funds for the Central Universities, “Pioneer”, Zhejiang Provincial Natural Science Foundation of China (LDT23F01011F01, LDT23F01015F01, LDT23F01014F01) and “Leading Goose” R&D Program of Zhejiang Province (2022C01068), and Youth Innovation Promotion Association of Chinese Academy of Sciences (2020108).
References


[31] Yunbin Tu, Liang Li, Chenggang Yan, Shengxiang Gao, and Zhengtao Yu. R3Net: relation-embedded representation re-


