Sliding Window Difference Detection for Change Captioning

Anonymous Author(s)
Submission Id: 1545

ABSTRACT

Change captioning aims to detect the differences between two similar images and generate the change description based on the differences. In the existing studies, to detect whether a region in an image has changed, they compare it with all the regions in another image, *i.e.*, using the *one-to-all* mode. However, it introduces some new problems. 1) inconsistent with the real situation. 2) producing the wrong correspondences between two images, especially when there are two or more same objects in one image. In addition, we find that the description of each change type is always related to some special words. Therefore, it is suitable to introduce the topic knowledge to guide the model to generate the change description more accurately. To address these issues, we propose the Sliding Window Difference Detection (SWDD) model, which consists of the Sliding-Window based Difference Finder (SWDF) encoder and the Topic-Aware Speaker (TAS) decoder. For one pixel in the "before" image, the SWDF generates a sliding window in the "after" image, whose center is the same as the position of the pixel, and judges whether the pixel has changed based on the pixels in the sliding window. With the change regions and the corresponding differences, the TAS generates a more accurate change description guided by the topic signal. The experimental results on the Spot-the-Diff and CLVER-Change datasets show that SWDD has significantly outperformed the baseline, and achieved the state-of-the-art performance under the evaluation metrics.

CCS CONCEPTS

• **Computing methodologies** → **Image representations**; **Natural language generation**.

KEYWORDS

change captioning, sliding window, pixel reconstruction, topic-aware

ACM Reference Format:

Anonymous Author(s). 2018. Sliding Window Difference Detection for Change Captioning. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (MM'22).* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Change Captioning is a challenging task. It requires the model to identify the detailed differences between the two similar images and generate a change description based on the differences. It is widely applied in medical imaging [31], aerial imagery [10], and fault detection [24]. Moreover, detecting the detailed information of the image is helpful to promote other computer communities to develop, such as image captioning [2, 35, 37] and visual question answering [3, 16, 26].

In the existing works, Jhamtani *et al.* [17] propose the Change Captioning task for the first time and collect a dataset, namely Spot-the-Diff, in which each image pair has one or more scene changes. Afterward, Park *et al.* [30] collect a new dataset for the task, called CLVER-Change which introduces distractors into the dataset, such as the changes in viewpoint, zoom, and illumination. As shown in Figure 1(top row), there is no scene change between the "before" and the "after" images, but the viewpoint has changed. As a result, the regions from the same position in the two images are unequal. It is clear that the distractors make the work of detecting the differences between two similar images more difficult.

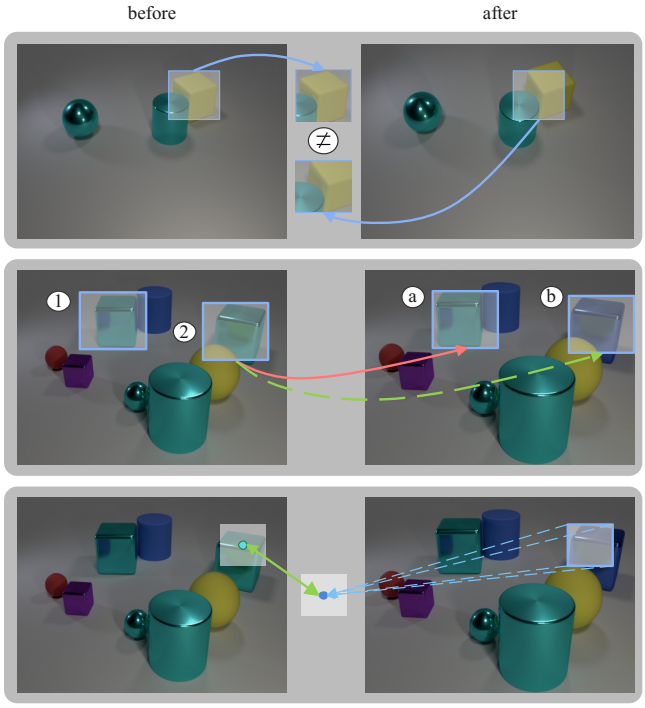

Figure 1: There is no change between the two images and the regions from the same position in the two images are unequal due to the viewpoint change (top row). There is an incorrect correspondence between the two images when using the *one-to-all* mode (middle row). An example of detecting differences in the "before" image by our proposed model, *i.e.,* SWDD (bottom row).

MM'22, October 10–14, 2022, Lisbon, Portugal

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

Table 1: Special word statistics of different change types. Whereinto, C, T, A, D, M, and DI denote color, texture, add, drop, move, and distractor, respectively.

Type	Words	Type	Words
C	red, green, ...	T	metal, shiny, ...
A	added, placed, ...	D	dropped, removed, ...
M	moved, location, ...	DI	remains, same, ...

Soon afterward, the subsequent studies [13, 15, 19, 25, 33] for change captioning are roughly divided into two types, according to the granularity of the extracted image pair's features, one to extract features at the pixel-level and the others to utilize the features at the instance-level. To handle the viewpoint change properly, Shi *et al.* [33] propose the M-VAM encoder to exhaustively measure the feature similarity across different regions in the two images to detect the differences at the pixel-level. Huang *et al.* [15] calculate the similarities between the objects belonging to different images for finding the change regions at the instance-level. However, these studies both use the *one-to-all* mode to detect the differences, *i.e.*, to detect whether a region has changed, they compare it with all the regions in another image. To some extent, they solve the problem of feature misalignment between two images caused by the distractors but also introduce new problems. 1) inconsistent with the real situation using the *one-to-all* mode. As shown in Figure 1(top row), the viewpoint change is always slight and there is no need for global comparisons. 2) producing incorrect correspondences between two images using the *one-to-all* mode, especially when there are two or more same objects in one image. As shown in Figure 1(middle row), the "before" image has two same objects, denoted by ① and ②, respectively. Whereinto, ② changes its color (green→blue) and gets the "after" image, the objects in the "after" image denoted by ⓐ and ⓑ. Ideally, ② will correspond to ⓑ (denoted by the green dotted line) and thus find the color change. However, using the *one-to-all* mode will more likely correspond ② to ⓐ (denoted by the red solid line) and thus consider no scene change. Hence, the incorrect correspondences will lead to misjudgment of the change type, and then harm the performance of the change captioning.

Moreover, there are six change types in the CLEVER-Change dataset, *i.e.*, COLOR (C), TEXTURE (T), ADD (A), DROP (D), MOVE (M), and DISTRACTOR (DI), and we find that each type has always related to some special words. As shown in Table 1, C is always related to "green" and "red" words; DI is always related to "same" and "remains" words. Therefore, it is suitable to introduce the topic knowledge to guide the model to generate the change description more accurately.

Based on the outlined problems and the finding, we propose the Sliding Window Difference Detection (SWDD) model for change captioning. In particular, we design the Sliding-Window based Difference Finder (SWDF) as the encoder and the Topic-Aware Speaker as the decoder (TAS). Whereinto, the SWDF consists of two modules, namely the Sliding-Window based Pixel Reconstruction (SWPR) module and the Image Difference Finder (IDF) module, respectively. As shown in Figure 1(bottom row), to detect whether the pixel (denoted by the green point) in the "before" image has changed, the SWPR constructs a sliding window (denoted by a blue bounding box) in the "after" image, whose center is the same as the position of the green point, and then utilizes the pixels in the sliding window to locally reconstruct the target pixel (denoted by the blue point) to deal with the distractors and eliminate the wrong correspondences. Subsequently, the IDF considers both the corresponding pixels and their backgrounds to comprehensively detect the differences. With the change regions and their differences, the TAS generates a more accurate change description guided by the topic signal.

To demonstrate the effectiveness of our proposed method, we conduct extensive experiments on the Spot-the-Diff and CLVER-Change datasets. Our proposed model outperforms several state-of-the-art models on most metrics by a wide margin and achieves the state-of-the-art performance. In a nutshell, the contributions of our work can be summarized as follows:

- We explore the shortcomings of existing works using the *one-to-all* mode and propose the novel SWDD model to deal with the distractors and eliminate the wrong correspondences between the image pair.
- We investigate the effectiveness of the topic knowledge for the first time and design the TAS decoder to generate more accurate descriptions guided by the topic signal.
- By conducting extensive experiments on the benchmark datasets, we demonstrate that our proposed model achieves the state-of-the-art performance. As side contributions, we release our code and parameters to facilitate reproduction at https://github.com/noname0123456/SWDD.git.

2 RELATED WORK

Our work is mainly related to two research areas, image change captioning and topic model. In this section, we introduce these two areas briefly.

2.1 Image Change Captioning

Different from image captioning [2, 11, 14, 22, 35, 37], image change captioning is a more challenging task. It aims to find the difference between the two similar images and generates the change description in natural language. Jhamtani *et al.* [17] propose the change captioning task for the first time and collect the Spot-the-diff dataset for the new task. The Spot-the-diff dataset has around 13K image pairs. Whereinto, there are one or more changes in the image pair, and the description is written by the annotators. The proposed DDLA model generates the change description based on the detected pixel differences. Park *et al.* [30] collect a new dataset, CLEVER-Change. Compared with the Spot-the-diff dataset, it has more image pairs and there are distractors in the dataset. To deal with the distractors, the DUDA model is designed to identify the scene change in the presence of distractors. Shi *et al.* [33] propose the M-VAM model to solve the problem of pixel misalignment caused by the distractors by moving the region in one image to the position of the corresponding region in another image at the pixel-level. Hosseinzadeh *et al.* [13] and Kim *et al.* [19] improve the performance by introducing an auxiliary task, which is generating a new image according to the original image and the change description. The main and auxiliary tasks promote each other by multi-task learning. Recently, some works are devoted to detecting differences at the

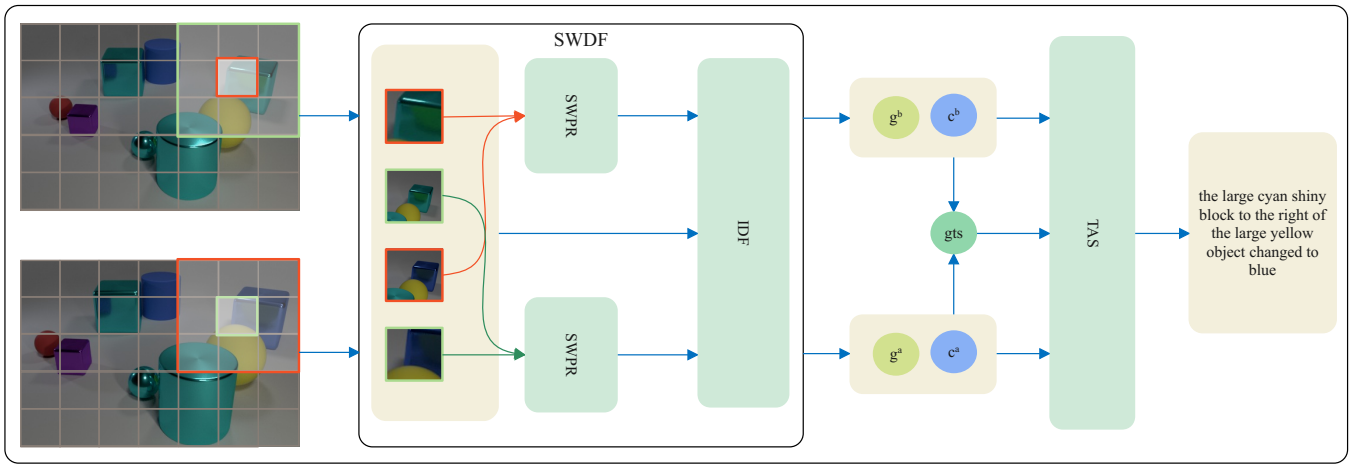

Figure 2: The overview of the SWDD model, which consists of the SWDF encoder and the TAS decoder. Whereinto, the SWDF includes two modules, namely SWPR and IDF, respectively. To detect whether the pixel has changed, we construct a sliding window (taking the sliding window size as 3 for example) in another image whose central position is the same as the pixel, and then they are input into the SWPR and IDF in turn to obtain the change regions (g^b and g^a) and the change differences (c^b and c^a). Finally, with the visual features and the global topic signal (gts), the TAS generates the change description.

instance-level using Faster RCNN [32]. Huang *et al.* [15] design the IFDC model to extract objects from the image pair, find the difference by comparing the objects belonging to different images and then generate the description in terms of object differences. Liao *et al.* [25] construct two scene graphs for the detected objects, the attribute scene graph and the spatial scene graph. The designed SGCC model describes the relative position relationship between objects correctly and overcomes the disturbances from viewpoint changes.

As mentioned in the studies [15, 33], they detect differences using the *one-to-all* mode, which is inconsistent with the real situation and preduces wrong correspondences between two images. Therefore, we propose the SWDD model to deal with the distractors and eliminate the wrong correspondences. Different from previous works, we locally reconstruct the target pixel based on the sliding window and consider the background of each pixel. The SWDD model locates the change regions more accurately and comprehensively. Compared with the methods [15, 25] extracting features at the instance-level, we have higher inference efficiency and more detailed information.

2.2 Topic Model

Topic model [5, 9, 23, 27] is one of the most important techniques of data mining. It can identify topics in documents, mine hidden information in a corpus, and extract information from unstructured text. In recent years, it is widely used in generating tasks. Chen *et al.* [6] propose the topic prediction module and then utilize the predicted topic to guide the video caption generation following the encoder-decoder pipeline. Wei *et al.* [36] introduce the bilingual topic knowledge into the neural machine translation, which enables the attention mechanism to focus on the topic-level features for generating accurate target words during translation. Bai *et al.* [4] devise a multi-topic and knowledgeable art description framework, which generates the description for the artwork according to three artistic topics.

Inspired by the above studies, we introduce the topic knowledge for change captioning for the first time and design the TAS decoder to generate the change description more accurately guided by the topic signal.

3 MODEL

Following the encoder-decoder architecture [17, 30], we propose the Sliding Window Difference Detection (SWDD) model for change captioning, which consists of the Sliding-Window based Difference Finder (SWDF) module as the encoder and the Topic-Aware Speaker (TAS) module as the decoder. As illustrated in Figure 2, similar images are input into the SWDF, which includes the Sliding-Window based Pixel Reconstruction (SWPR) module and the Image Difference Finding (IDF) module, used to locally reconstruct the target pixel based on the sliding window and find the differences in terms of the corresponding pixels and their backgrounds, respectively. Afterward, the TAS generates the change description with the visual features guided by the topic signal.

3.1 SWDF Encoder

Given a pair of "before" image I^b and "after" image I^a, we use the pre-trained CNN [21] model to extract the feature maps of them, which are denoted by $f^b \in \mathbb{R}^{W*H*C}$ and $f^a \in \mathbb{R}^{W*H*C}$, respectively. Whereinto, W, H, and C are the width, the height, and the channel size of the feature map, respectively.

3.1.1 SWPR Module. Due to the distractors, the subtraction between f^b and f^a suffers from the change of viewpoint. In traditional methods [15, 33], either pixel-level or instance-level, using the *one-to-all* mode easily produces the wrong correspondences between two images and thus misjudges the type of scene change, especially

when there are two or more same objects in one image. In addition, we find that the change of viewpoint is usually slight. To solve the above problems, we propose the SWPR module to reconstruct the target pixel locally. For each pixel to be detected, the SWPR only focuses on the pixels inside its corresponding sliding window in another image and locally reconstructs its target pixel based on the similarity between them.

In particular, for one pixel whose location is (x, y) in "before" image denoted by $f^b_{(x,y)} \in \mathbb{R}^C$, we generate a sliding window whose central position is also (x, y) in the "after" image denoted by $SW^a_{(x,y)}$, which is a collection of pixels, and it can be formulated as follows,

$$\begin{aligned} SW^a_{(x,y)} = \{ & f^a_{(x',y')} | x - \lfloor ws/2 \rfloor \le x' \le x + \lfloor ws/2 \rfloor, \\ & y - \lfloor ws/2 \rfloor \le y' \le y + \lfloor ws/2 \rfloor \}, \end{aligned} \tag{1}$$

where ws denotes the size of sliding window, which is a hyperparameter. Different from the previous works [15, 33], we only calculate the similarity between $f^b_{(x,y)}$ and $f^a_{(x',y')} \in SW^a_{(x,y)}$, formally,

$$\alpha^b_{(x',y')} = softmax(W^b tanh(W^b_1 f^b_{(x,y)} + W^b_2 f^a_{(x',y')})), \tag{2}$$

where $W^b \in \mathbb{R}^{1*512}$, $W^b_1 \in \mathbb{R}^{512*C}$, and $W^b_2 \in \mathbb{R}^{512*C}$ are learnable parameters. With the similarity scores between pixels, we propose two types of operations to locally reconstruct the target pixels in order to deal with different viewpoint changes. One is to weigh pixels in soft operation, and the other is to select the most similar pixel in hard operation.

$$\begin{cases} f^{b-s}_{(x,y)} = \sum_{(x',y')} \alpha^b_{(x',y')} f^a_{(x',y')}, \\ f^{b-h}_{(x,y)} = \arg\max_{f^a_{(x',y')}} \alpha^b_{(x',y')}, \end{cases} \tag{3}$$

where $f^{b-s}_{(x,y)}$ and $f^{b-h}_{(x,y)}$ denote the target pixels generated by soft operation and hard operation, respectively.

By calculating the differences between the original and the target pixels, we choose the most appropriate one as the final target pixel denoted by $f^{b-t}_{(x,y)}$.

$$f^{b-t}_{(x,y)} = \begin{cases} f^{b-s}_{(x,y)}, & if \quad |f^b_{(x,y)} - f^{b-s}_{(x,y)}| \le |f^b_{(x,y)} - f^{b-h}_{(x,y)}|, \\ f^{b-h}_{(x,y)}, & otherwise. \end{cases} \tag{4}$$

3.1.2 IDF Module. Since the scene change is always at the instance-level which could be regarded as the cluster of pixels, when detecting the differences, we should consider the background information of pixels. Therefore, we propose the IDF module to comprehensively detect the differences in two aspects, one is the difference in the corresponding pixels, and the other is the difference in their backgrounds.

We first calculate the pixel difference by substraction between the original pixel and its target pixel, formally.

$$d^{b-p}_{(x,y)} = f^b_{(x,y)} - f^{b-t}_{(x,y)}, \tag{5}$$

where $d^{b-p}_{(x,y)}$ represents the pixel difference. To represent the background of $f^b_{(x,y)}$, we construct a sliding window $SW^b_{(x,y)}$ in the "before" image like above mentioned and make a mean-pooling on it to get the background representation $b^b_{(x,y)}$. In the same way, we

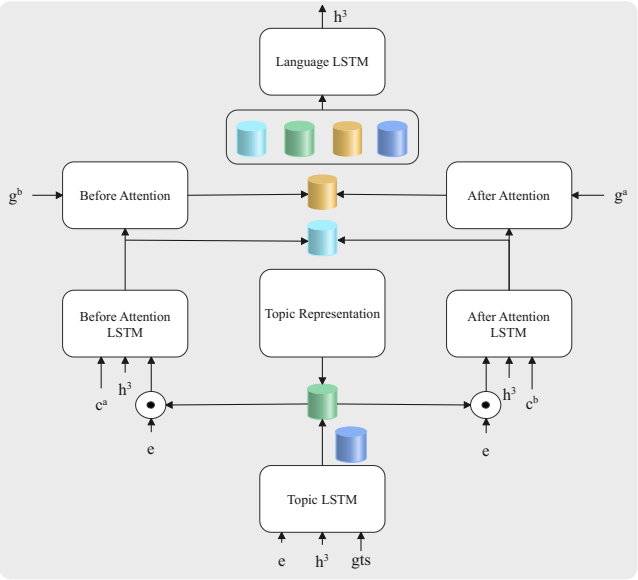

Figure 3: The detailed structure of the TAS. There are four LSTMs in TAS, namely Topic LSTM, Before Attention LSTM, After Attention LSTM, and Language LSATM, which are used to generate the topic signal, attend to the "before" image, attend to the "after" image, and generate the description, respectively.

can obtain the background representation of $f^{b-t}_{(x,y)}$, which is denoted by $b^{b-t}_{(x,y)}$. Afterward, we calculate the background difference as follows,

$$d^{b-b}_{(x,y)} = b^b_{(x,y)} - b^{b-t}_{(x,y)}, \tag{6}$$

where $d^{b-b}_{(x,y)}$ denotes the background difference. Finally, we fuse the pixel difference and the background difference to obtain the whole difference representation $d^b_{(x,y)}$,

$$d^b_{(x,y)} = \lambda * d^{b-p}_{(x,y)} + (1 - \lambda) * d^{b-b}_{(x,y)}, \tag{7}$$

where λ is a hyperparameter.

Based on the original pixels and their corresponding differences, we adopt the DUDA model to calculate the change probabilities p^b of all pixels in the "before" image following the previous work [30].

$$p^b = DUDA([f^b; d^b]), \tag{8}$$

where [;] denotes the concatenation operation. Subsequently, we selectively keep the change regions and change differences based on the change probabilities.

$$g^b = p^b \odot f^b \tag{9}$$

$$c^b = p^b \odot d^b \tag{10}$$

where g^b and c^b represent the change regions and change differences eventually retained, and $\odot$ denotes hadamard product.

In the same way, we can obtain the change regions and change differences of the "after" image, which are denoted by g^a and c^a, respectively.

3.2 TAS Decoder

Inspired by the architecture [2], we design the novel TAS decoder to generate the change description guided by the topic signal, which is helpful to generate the keywords related to the change type and so as to generate the description more accurately. As shown in Figure 3, there are four LSTMs in TAS, namely Topic LSTM used to generate the topic signal, Before Attention LSTM and After Attention LSTM used to pay attention to the "before" and "after" images, and Language LSTM used to generate the description, respectively, whose hidden states are denoted by h^0, h^1, h^2, and $h^3 \in \mathbb{R}^{1000}$, separately. Next, we introduce the generation process in detail.

We first set the number of topics to m, which is a hyperparameter, and then randomly initialize the topic representations, denoted by $M \in \mathbb{R}^{m\times 1000}$, where each row represents a topic and is continuously updated as training progresses. In order to implement the guidance of the topic signal during the generation process, we generate two kinds of topic signals, namely global topic signal (gts) and temporal topic signal (tts), respectively. Whereinto, gts is used to guide the generation of tts at each generation step and tts is input into subsequent components to strengthen the topic guidance.

To generate the gts, we first calculate the global topic distribution p_g^t based on the mean-pooled change differences, *i.e.*, $\overline{c^a}$ and $\overline{c^b}$, and then perform a weighted sum of the topic representations M based on p_g^t.

$$p_g^t = softmax(W_g^t[\overline{c^a}; \overline{c^b}]), \tag{11}$$

$$gts = M^{\mathrm{T}} p_g^t, \tag{12}$$

where $W_g^t \in \mathbb{R}^{m\times 2C}$ is a learnable parameter. Afterward, the gts is input into the Topic LSTM to generate the tts at time step t. Except for gts, the input x_t^0 of the Topic LSTM also includes the embedding e_t of the input word and the previous hidden state h_{t-1}^3 of the Language LSTM. The operation of the Topic LSTM over a single step is as follows:

$$x_t^0 = [e_t; gts; h_{t-1}^3], \tag{13}$$

$$h_t^0 = LSTM(x_t^0, h_{t-1}^0). \tag{14}$$

Subsequently, we calculate the similarity between h_t^0 and the topic representations M as the temporal topic distribution p_t^t. By weighing the sum of M based on p_t^t, we can obtain the tts. It can be formulated as follows,

$$p_t^t = softmax(Mh_t^0), \tag{15}$$

$$tts = M^{\mathrm{T}} p_t^t. \tag{16}$$

To enhance the topic guidance, we perform multiplication on the tts and e_t to generate the topic-aware input features for the Before Attention LSTM and the After Attention LSTM. Besides, the regions focused on the "before" and "after" images should restrict and complement each other, so we input $\overline{c^a}$ and $\overline{c^b}$ into the Before Attention LSTM and the After Attention LSTM, respectively. The whole input x_t^1 and x_t^2 of Before Attention LSTM and After Attention LSTM are as follows,

$$\begin{cases} x_t^1 = [e_t \odot tts; \overline{c^a}; h_{t-1}^3], \\ x_t^2 = [e_t \odot tts; \overline{c^b}; h_{t-1}^3]. \end{cases} \tag{17}$$

We input the x_t^1 into the Before Attention LSTM to get the hidden state h_t^1. Afterward, the h_t^1 is as the key input into the Before Attention to attend to the special regions in the "before" image. Formally,

$$h_t^1 = LSTM(x_t^1, h_{t-1}^1), \tag{18}$$

$$w_{(x,y)}^b = softmax(W^{b-a} tanh(W_1^{b-a} g_{(x,y)}^b + W_2^{b-a} h_t^1)) \tag{19}$$

$$att_t^b = \sum_{(x,y)} w_{(x,y)}^b g_{(x,y)}^b \tag{20}$$

where $W^{b-a} \in \mathbb{R}^{1\times 512}$, $W_1^{b-a} \in \mathbb{R}^{512\times C}$, and $W_2^{b-a} \in \mathbb{R}^{512\times 1000}$ are learnable parameters, and the att_t^b is the attention vector of the "before" image at the time step t. In the same way, we can get the hidden state h_t^2 of the After Attention LSTM and the attention vector att_t^a of the "after" image. And then, we perform addition on the att_t^b and att_t^a, the h_t^1 and h_t^2 to get the final attended vector att_t and the hidden state h_t^{att}, respectively.

$$\begin{cases} att_t = att_t^b + att_t^a, \\ h_t^{att} = h_t^1 + h_t^2. \end{cases} \tag{21}$$

With the attended feature and the topic signal, we construct the whole input x_t^3, which is fed into the Language LSTM for generating the description.

$$x_t^3 = LSTM(tts, att_t, h_t^0, h_t^{att}) \tag{22}$$

$$h_t^3 = LSTM(x_t^3, h_{t-1}^3) \tag{23}$$

$$p(y_t|y_{1:t-1}) = softmax(W^o h_t^3) \tag{24}$$

where $W^o \in \mathbb{R}^{v\times 1000}$ (v denotes the size of vocabulary) is a learnable parameter, y_t is the ground truth word at time step t, and $y_{1:t-1}$ refers to a sequence of words $Y = \{y_1, ..., y_{t-1}\}$.

Furthermore, we constrain the gts by the change type to generate the correct topic signals at each generation step.

$$p(c|I^b, I^a) = softmax(W^c gts) \tag{25}$$

where $W^c \in \mathbb{R}^{6\times 1000}$ (6 is the number of the change types) is a learnable parameter, and c denotes the ground truth change type. Therefore, for the input image pair, the total loss is:

$$L = -log(p(c|I^b, I^a)) - 1/T \sum_{t=1}^{T} log(p(y_t|y_{1:t-1})) \tag{26}$$

where T is the length of the ground truth description.

4 EXPERIMENTS

4.1 Datasets and Evaluation Metrics

4.1.1 Datasets. In this paper, we conduct our experiments on the Spot-the-Diff [17] and Clever-Change [30] datasets. The Spot-the-Diff dataset has around 13K image pairs with corresponding change descriptions, which are extracted from VIRAT surveillance video dataset [28]. The CLEVER-Change dataset is built based on the CLEVR engine [18]. In the dataset, for one "before" image, it corresponds to two "after" images. One is created by changing the camera position, zoom, or illumination (*i.e.*, Distractor), and the other is created by changing objects (*i.e,* Scene Change). There are six change types, COLOR (C), TEXTURE (T), MOVE (M), ADD (A), DROP (D), and Distractor (DI), which denote one object changes its color or texture, or location, one object is newly placed or removed, and no scene change, respectively.

4.1.2 Evaluation Metrics. Following the previous work [30], we use the four most common metrics, *i.e.*, BLEU-4 [29], METEOR [8], CIDEr [34], and SPICE [1] to evaluate the performances of our proposed model.

4.2 Experimental Settings

Similar to [30, 33], we use the ResNet-101 [12] pre-trained on the ImageNet [7] to extract features of the image pair. The shape of the feature map is $14 \times 14 \times 1024$. The hyperparameters of ws, λ, and m are set to 3, 0.9 and 14, respectively. In the TAS, the sizes of the word embedding, the hidden state, and the topic signal are set to 1000, respectively. We train our model for 70 epochs using the Adam Optimizer [20]. The batch size is 64 and the initial learning rate is set to 5e-4. We decay the learning rate by 0.5 every 10 epochs.

4.3 Baselines

We divide the baselines into two groups, one to extract features of image pair at the pixel-level and the other to extract features at the instance-level, denoted by Group A (GA) and Group B (GB), respectively.

The baselines in GA are as follows,

- DUDA: Park *et al.* [30] propose the DUDA model equipping with a dual attention mechanism and a dynamic speaker, used to detect the differences in the image pair and generate change description based on the differences, jointly.
- M-VAM: Shi *et al.*[33] propose the M-VWM encoder to exhaustively measure the feature similarity across different regions in the two images to detect the differences.
- M-VAM-RAF: Based on M-VAM, Shi *et al.*[33] propose the reinforcement attention fine-tuning process to settle down the exposure bias problem.
- DUDA+TIRG: Based on the DUDA, Hosseinzadeh *et al.* [13] propose an auxiliary task, *i.e.*, image generation, to promote the change captioning by multi-task learning.
- VACC: Similar to DUDA+TIRG, Kim *et al.* [19] also introduce a new task to assist chang captioning.

The baselines in GB are as follows,

- IFDC: Huang *et al.* [15] propose the IFDC model to extract features at instance-level and detect the differences between objects belonging to different images.
- SGCC: Liao *et al.* [25] propose the SGCC model to construct two scene graphs for the detected objects and generate change description based on the scene graphs.

4.4 Experimental Results

As shown in Table 2, we compare the performances with the baselines in two groups on the CLEVR-Change dataset. Compared with GA, we achieve the best performances at all metrics and outperform the baselines by a wide margin. We push the B@4, ME, CD, and SP from 52.4, 37.8, 115.8, and 31.1 to 55.2, 39.9, 123.1, and 32.6, respectively. Compared with GB, we also achieve the best performances on B@4, CD, and SP. The performance on ME is still a competitive result compared with other baselines. Besides, as we all know, extracting features at the pixel-level is easier than at the instance-level, and we also do not require further processing of the extracted features, such as clustering, predicting attributes. Therefore, our model has higher inference efficiency than the baselines in GB. Overall, our model outperforms all baselines and achieves the state-of-the-art performance. We own the significant improvement to the sliding-window based pixel reconstruction and the guidance of the topic signal. Different from the baselines, the SWDD model does not produce wrong correspondences between the image pair when dealing with the distractors, and with the guidance of the topic signal, it easily generates the keywords related to the change type and then generates the change description more accurately. These results again verify the contributions of our proposed model.

Table 2: The performances of SWDD and other state-of-the-art methods on the entire CLEVER-Change dataset. B@4, ME, RG, CD and SP denote BLEU-4, METEOR, ROUGE-L, CIDEr-D and SPICE, respectively. Best performance is highlighted in bold.

	B@4	ME	CD	SP
DUDA (2019)	47.3	33.9	112.3	24.5
M-VAM (2020)	50.3	37.0	114.9	30.5
M-VAM-RAF (2020)	51.3	37.8	115.8	30.7
DUDA+TIRG (2021)	51.2	37.7	115.4	31.1
VACC (2021)	52.4	37.5	114.2	31.0
IFDC (2021)	49.2	32.5	118.7	-
SGCC (2021)	51.1	**40.6**	121.8	32.2
SWDD (Ours)	**55.2**	39.9	**123.1**	**32.6**

The detailed breakdown of the evaluation in terms of the change types is shown in Table 3. Obviously, in each metric, not less than half of the performances are optimal. In some other performances, they are also competitive. In particular, the "Drop" case has the largest improvement (from 118.5 to 128.1) among the six types for CD. For SP, "Color", "Texture", and "Add" cases have also obvious improvements (from 30.0 to 31.5, from 31.1 to 32.7, and from 30.8 to 32.0). For the "Distractor" case, we obtain the suboptimal performance on CD, which is slightly inferior to the M-VAM-RAF, but the performances of most other change types are better than its. It can be seen that our model can well balance the scene change and distractors and achieve the optimal performance in total.

To further evaluate our model in a realistic situation, we also conduct experiments on the Spot-the-Diff dataset. As shown in Table 4, it is clear that we obtain the optimal performances on ME, CD, and SP, and the performance on B@4 is slightly inferior to the best, but it is still competitive with the other baselines. Obviously, the SWDD outperforms the previous best model by a wide margin and achieves the state-of-the-art performance. The reason is that in a realistic situation, the distractors are ubiquitous, and the description still contains obvious topic information. As a result, the SWDD still performs well in a realistic situation.

4.5 Ablation Study

In order to investigate the effect of each module in SWDD, we conduct extensive ablation experiments on the CLEVER-Change dataset in Table 5. We construct two baselines, denoted by BA

Table 3: The performances of SWDD and other state-of-the-art methods on CLEVER-Change by change types.

	CD						ME						SP					
	C	T	A	D	M	DI	C	T	A	D	M	DI	C	T	A	D	M	DI
DUDA (2019)	120.4	86.7	108.2	103.4	56.4	110.8	32.8	27.3	33.4	31.4	23.5	43.2	21.2	18.3	22.4	22.2	15.4	28.4
M-VAM-RAF (2020)	122.1	98.7	126.3	115.8	82.0	**122.6**	35.8	32.3	37.8	36.2	27.9	**66.4**	28.0	26.7	30.8	32.3	22.5	33.4
IFDC (2021)	133.2	99.1	**128.2**	118.5	**82.1**	114.2	33.1	27.9	36.2	31.4	31.2	40.1	-	-	-	-	-	-
SGCC (2021)	128.0	122.9	117.1	116.9	77.1	116.5	37.8	**36.1**	38.9	36.7	**32.8**	52.2	30.0	31.1	30.8	30.1	**25.3**	**35.0**
SWDD (Ours)	**139.8**	**125.0**	124.1	**128.1**	78.6	116.9	**38.9**	35.8	**39.3**	**37.2**	29.1	50.9	**31.5**	**32.7**	**32.0**	**32.5**	23.5	34.8

Table 4: The performances of SWDD and other state-of-the-art methods on the Spot-the-Diff dataset. Best performance is highlighted in bold.

	B@4	ME	CD	SP
DDLA (2018)	8.5	12.0	32.8	-
DUDA (2019)	8.1	11.5	34.0	-
M-VAM (2020)	10.1	12.4	38.1	14.0
M-VAM-RAF (2020)	**11.1**	12.9	42.5	17.1
DUDA+TIRG (2021)	8.1	12.5	34.5	-
VACC (2021)	9.7	12.6	41.5	-
IFDC (2021)	8.7	11.7	37.0	-
SGCC (2021)	8.6	13.1	42.9	17.2
SWDD (Ours)	10.3	**13.4**	**45.0**	**17.6**

Table 5: Ablation studies on CLEVR-Change. w/ means with.

	B@4	ME	CD	SP
BA (baseline)	47.3	33.9	112.3	24.5
BB (baseline)	50.3	37.0	114.9	30.5
w/ soft operation	54.3	39.3	118.2	32.4
w/ hard operation	54.6	39.6	119.0	32.5
w/ background	54.8	39.7	120.1	32.6
w / 3 LSTMs	52.2	37.8	w / TAS	55.2
39.9	123.1	32.6		

and BB, respectively. Whereinto, BA detects the differences by subtraction between two images like DUDA, and the BB uses the *one-to-all* mode to find the differences at pixel-level like M-VAM.

4.5.1 The effect of SWDF. Compared with the baselines, we detect the differences by the SWDF. There are three variants in the group.

- w/ soft operation: Based on the generated sliding window, we reconstruct the target pixel in the soft operation, *i.e.*, weighing pixels in the sliding window. Compared with BA, the performances increase by 14.8%, 15.9%, 5.3%, and 32.2% on metrics B@4, ME, CD, and SP, respectively. Compared with BB, the performances increase by 8.0%, 6.2%, 2.9%, and 6.2% on metrics B@4, ME, CD, and SP, respectively. Intuitively, the SWDF is effective and significantly improves the performance. It is owing to this that the SWDF locally reconstructs the target pixel based on the sliding window, which is consistent with the real situation and eliminates the wrong correspondences between the image pair introduced by the two baselines.
- w/ hard operation: We also propose the hard operation to locally reconstruct the target pixel, *i.e.*, selecting the most similar pixel, to supplement the first variant. The final target pixel is reconstructed as shown in the Formulation 4. Compared with the first variant, this variant improves the performances on all metrics, which shows that it is effective. That is because the changes in the viewpoint are usually different, and the combination of these two operations covers it more effectively.
- w/ background: Based on the second variant, when detecting whether a pixel has changed, we introduce its background information. The final pixel difference is represented as shown in the Formulation 7. We compare the performances with those of the second variant and find that all metrics are better. It can be seen that introducing the background information is helpful. It owes to the fact that the change is always at the instance-level, which could be regarded as a cluster of pixels. As a result, introducing the background information is beneficial to detect differences comprehensively.

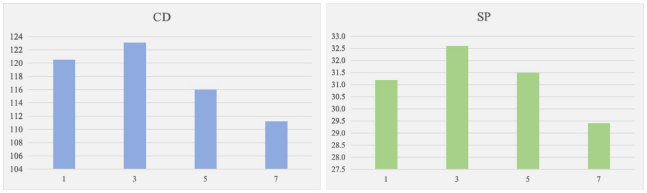

Figure 4: The performances on CD and SP with various sliding window sizes.

4.5.2 The effect of TAS. We propose the TAS module to replace the original decoder, whose performances are better in all metrics. However, only the CIDEr score has improved greatly (from 120.1 to 123.1). By comparing the generated captions of various models, we suggest that the phenomenon may be caused by the *tf-idf* based metric (*i.e.*, CIDEr). Guided by the topic signal, the model more accurately generates the keywords related to the change type, which tend to have higher *tf-idf* values. It can be seen that with TAS, the model can generate a more accurate change description.

In addition, we have done a great quantity of experiments with different sliding window sizes. As illustrated in Figure 4, when the

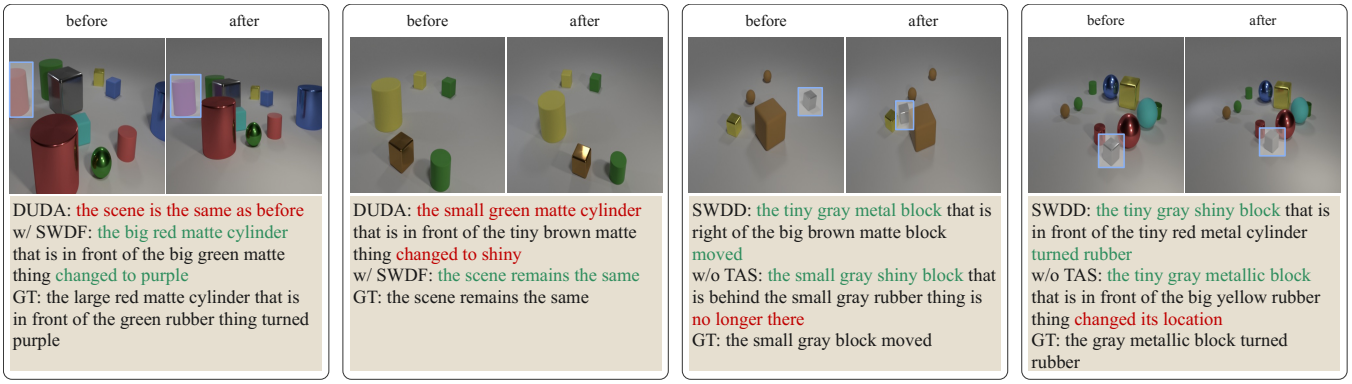

Figure 5: Case studies on the CLEVER-Change dataset.

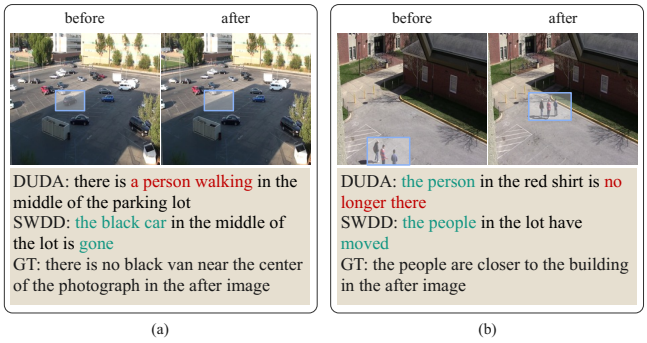

Figure 6: Case studies on the Spot-the-Diff dataset.

size of the sliding window is set to 3, the model obtains the greatest performances, which are better than those of the size set to 1, *i.e.*, direct subtraction of two input images. Besides, with the increase in size, the performances decrease gradually. Based on these observations, it can be further demonstrated that the viewpoint change is usually slight, using the *one-to-all* mode that is inconsistent with the real situation, and easily generates wrong correspondences between two images. Therefore, generating a sliding window with an appropriate size for local pixel reconstruction proposed in this paper is a better choice.

4.6 Case Study

As shown in Figure 5 and Figure 6, we randomly select four examples on the CLEVER-Change dataset and two examples on the Spot-the-Diff dataset to show the effectiveness of our proposed SWDD, respectively. We use the blue bounding boxes to represent the changed regions in the image pair. In the descriptions generated by various models, we color the changed object and the change type, where green means consistent and red means inconsistent with the ground truth, respectively. Whereinto, "w/", "w/o", and "GT" denote "with", "without" and "ground truth", respectively.

In Figure 5(a), the DUDA model fails to find the color change of the large red matte cylinder, which is caused by the position of the two objects in the image pair being different due to the viewpoint change. In Figure 5(b), there is no scene change between the image pair. Due to the different illuminations, the DUDA mistakenly considers that the texture of the small green matte cylinder has changed. However, when we use the SWDF as the encoder, we successfully address these problems. It is clear that the SWDF can effectively detect image differences in the presence of the distractors. In Figure 5(c), removing the TAS module from the SWDD model, the new model still correctly detects the changed object, *i.e.*, the small gray block, but it misidentifies the change type (MOVE → DROP). Figure 5(d) has the same problem. Therefore, it can be concluded that the TAS decoder is helpful to generate the keywords related to the change type, so as to generate descriptions more accurately.

In Figure 6(a), the DUDA model does not identify the real changed object, *i.e.*, the black car, so the description generated by it is far away from the ground truth description. In Figure 6(b), the description generated by the DUDA model includes the real changed object, but it still misjudges the change type (MOVE → DROP). However, in all of these cases above, the SWDD correctly generates the description. It can be seen that the SWDD proposed in this paper is still effective in a real situation.

5 CONCLUSION

In this paper, we propose the novel Sliding Window Difference Detection (SWDD) model, which consists of the SWDF encoder and the TAS decoder. Compared with other methods, the SWDF encoder comprehensively detects the differences by local pixel reconstruction based on the sliding window and considering the background information, which deals with the distractors well and eliminates the wrong correspondences. Besides, the TAS decoder introduces the topic knowledge for change captioning for the first time, which is helpful to generate keywords related to the change type and then generate the change description more accurately. The experimental results on the Spot-the-Diff and the CLEVER-change datasets demonstrate the effectiveness of our proposed model.

REFERENCES

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*. Springer, 382–398.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6077–6086.

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*. 2425–2433.

[4] Zechen Bai, Yuta Nakashima, and Noa Garcia. 2021. Explain Me the Painting: Multi-Topic Knowledgeable Art Description Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5422–5432.

[5] Kayhan Batmanghelich, Ardavan Saeedi, Karthik Narasimhan, and Sam Gershman. 2016. Nonparametric spherical topic modeling with word embeddings. In *Proceedings of the conference. Association for computational linguistics. Meeting*, Vol. 2016. NIH Public Access, 537.

[6] Shizhe Chen, Jia Chen, and Qin Jin. 2017. Generating video descriptions with topic guidance. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*. 5–13.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[8] Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*. 376–380.

[9] Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics* 8 (2020), 439–453.

[10] Lionel Gueguen and Raffay Hamid. 2015. Large-scale damage detection using satellite imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1321–1328.

[11] Longteng Guo, Jing Liu, Jinhui Tang, Jiangwei Li, Wei Luo, and Hanqing Lu. 2019. Aligning linguistic words and visual semantic units for image captioning. In *Proceedings of the 27th ACM International Conference on Multimedia*. 765–773.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[13] Mehrdad Hosseinzadeh and Yang Wang. 2021. Image Change Captioning by Learning from an Auxiliary Task. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2725–2734.

[14] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. 2019. Attention on attention for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4634–4643.

[15] Qingbao Huang, Yu Liang, Jielong Wei, Cai Yi, Hanyu Liang, Ho-fung Leung, and Qing Li. 2021. Image Difference Captioning with Instance-Level Fine-Grained Feature Representation. *IEEE Transactions on Multimedia* (2021).

[16] Qingbao Huang, Jielong Wei, Yi Cai, Changmeng Zheng, Junying Chen, Ho-fung Leung, and Qing Li. 2020. Aligned dual channel graph convolutional network for visual question answering. In *Proceedings of the 58th annual meeting of the association for computational linguistics*. 7166–7176.

[17] Harsh Jhamtani and Taylor Berg-Kirkpatrick. 2018. Learning to describe differences between pairs of similar images. *arXiv preprint arXiv:1808.10584* (2018).

[18] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2901–2910.

[19] Hoeseong Kim, Jongseok Kim, Hyungseok Lee, Hyunsung Park, and Gunhee Kim. 2021. Agnostic Change Captioning with Cycle Consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2095–2104.

[20] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012).

[22] Guang Li, Linchao Zhu, Ping Liu, and Yi Yang. 2019. Entangled transformer for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8928–8937.

[23] Shaohua Li, Tat-Seng Chua, Jun Zhu, and Chunyan Miao. 2016. Generative topic embedding: a continuous representation of documents. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 666–675.

[24] Bin Liao, You Du, and Xiangyun Yin. 2020. Fusion of Infrared-visible images in UE-IoT for Fault point detection based on GAN. *IEEE Access* 8 (2020), 79754–79763.

[25] Zeming Liao, Qingbao Huang, Yu Liang, Mingyi Fu, Yi Cai, and Qing Li. 2021. Scene Graph with 3D Information for Change Captioning. In *Proceedings of the 29th ACM International Conference on Multimedia*. 5074–5082.

[26] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3195–3204.

[27] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic evaluation of topic coherence. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*. 100–108.

[28] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. 2011. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR 2011*. IEEE, 3153–3160.

[29] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

[30] Dong Huk Park, Trevor Darrell, and Anna Rohrbach. 2019. Robust change captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4624–4633.

[31] Julia Patriarche and Bradley Erickson. 2004. A review of the automated detection of change in serial imaging studies of the brain. *Journal of digital imaging* 17, 3 (2004), 158–174.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28 (2015).

[33] Xiangxi Shi, Xu Yang, Jiuxiang Gu, Shafiq Joty, and Jianfei Cai. 2020. Finding it at another side: A viewpoint-adapted matching encoder for change captioning. In *European Conference on Computer Vision*. Springer, 574–590.

[34] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4566–4575.

[35] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3156–3164.

[36] Xiangpeng Wei, Yue Hu, Luxi Xing, Yipeng Wang, and Li Gao. 2019. Translating with bilingual topic knowledge for neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7257–7264.

[37] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*. PMLR, 2048–2057.