

# Generating Landmark-based Manipulation Instructions from Image Pairs

Sina Zarrieß<sup>1</sup>, Henrik Voigt<sup>1</sup>, David Schlangen<sup>2</sup> and Philipp Sadler<sup>2</sup>

<sup>1</sup>University of Bielefeld <sup>2</sup>University of Potsdam

<sup>1</sup>first.last@uni-bielefeld.de <sup>2</sup>first.last@uni-potsdam.de

## Abstract

We investigate the problem of generating landmark-based manipulation instructions (e.g. *move the blue block so that it touches the red block on the right*) from image pairs showing a *before* and an *after* state in a visual scene. We present a transformer model with difference attention heads that learns to attend to target and landmark objects in consecutive images via a difference key. Our model outperforms the state-of-the-art for instruction generation on the BLOCKS dataset and particularly improves the accuracy of generated target and landmark references. Furthermore, our model outperforms state-of-the-art models on a difference spotting dataset.

## 1 Introduction

When speakers produce instructions for tasks in visual environments, they often use landmarks and complex locative expressions to guide listeners to a goal state. Landmarks are well-known to be highly beneficial for achieving communicative success in situated collaborative dialogue tasks like object search, navigation or manipulation (Dräger and Koller, 2012; Clarke et al., 2013). Yet, the accurate generation of landmark-based instructions has been a long-standing challenge in NLG, as it requires complex visual-spatial and linguistic-pragmatic reasoning (Kelleher and Kruijff, 2006). Recent work on *generating* instructions has mostly looked at the navigation domain (Fried et al., 2018; Schumann and Riezler, 2021), whereas work on instruction *following* has shown great interest in manipulation tasks (Bisk et al., 2016; Misra et al., 2017; Shridhar et al., 2020).

In this paper, we investigate the task of generating landmark-based manipulations instructions from image-only input. We use Bisk et al. (2016)’s BLOCKS dataset as it provides both human-generated instructions and corresponding images of a “before state” and an “after state” (see Figure 1).

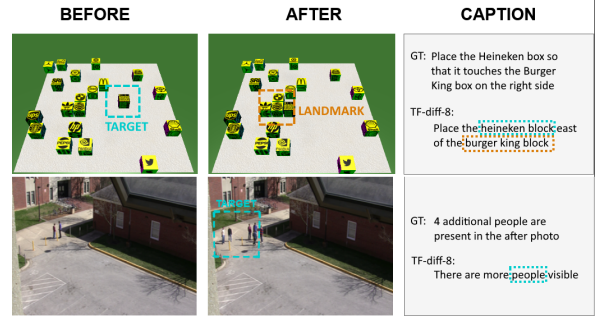


Figure 1: Image-pairs from BLOCKS (top) and Spot-the-diff (bottom) with descriptions generated by our best model. The targets and landmarks are manually highlighted for better view.

We present a transformer-based generation model with a simple but novel difference attention head designed to visually ground complex locative expressions and target-landmark references in image pairs. We show that our model clearly exceeds the performance of Rojowiec et al. (2020)’s existing baseline models on this task, in greatly improving the accuracy of generated target and landmark references. In contrast to other recent instruction generation models (Fried et al., 2017; Köhn et al., 2020; Schumann and Riezler, 2021), our approach does not use any symbolic representations of scene states and trajectories.

A core challenge for instruction generation in our set-up is that the model needs to reason about differences between the “before state” and “after state” represented as an image pair (see Figure 1). As a result of this reasoning, the model should be able to detect the target of the manipulation (e.g. *heineken block*) and verbalizing a suitable description of nearby landmarks (e.g. *east of the burger king block*). We note that the visual reasoning involved here is similar to the problem of spotting image differences or changes, which is a challenging computer vision task (Park et al., 2019; Shi et al., 2020; Oluwasanmi et al., 2019; Gilton et al., 2020).

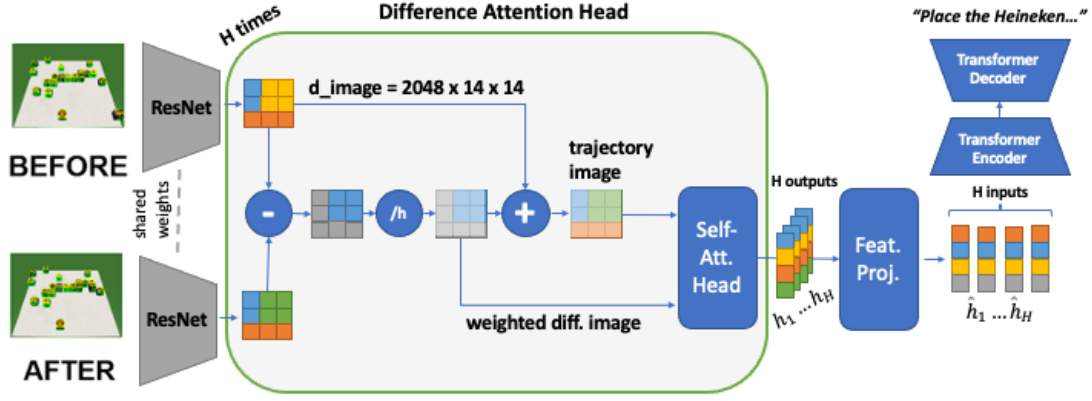


Figure 2: Our difference attention architecture

Thus, for comparison, we use Park et al. (2019)’s model as an additional baseline for instruction generation on BLOCKS. Furthermore, we compare our transformer model against the state-of-the-art on the Spot-the-Diff task with real-world images (Jhamtani and Berg-Kirkpatrick, 2018a).

## 2 Model

We present a transformer-based model that encodes pairs of *before* and *after* images to generate instructions that describe a particular manipulation to be accomplished in a visual scene. To achieve this, the model needs to learn latent visual-linguistic representations that encode information about the change or manipulation shown in the image pair. As shown in Figure 2, its main idea is a difference attention head that computes an attention map for a visual input state conditioned on the difference to its preceding state in the input.

Our starting point is a vanilla transformer model (Vaswani et al., 2017) that implements self-attention heads, which compute attention maps over values  $\mathbf{V}$  given queries  $\mathbf{Q}$  and keys  $\mathbf{K}$  representing elements of, e.g., a word sequence. A straightforward way to process image pairs with these heads is to allocate two of them: one for the *before* image embedding  $v_1$  and one for the *after* image embedding  $v_2$ .

We propose a difference attention head that exploits an explicit representation of the difference between the two embeddings and set this to  $\mathbf{K}$  as a supervision signal that is intended to support the learning of difference-oriented representations. As there is no *before* image for  $v_1$ , we obtain two difference attention heads for an image pair:

- (i)  $h_1$  with  $\mathbf{K} = c_1 = 0$  which attends everywhere equally

- (ii)  $h_2$  with  $\mathbf{K} = c_2 = v_2 - v_1$  which attends on changes specifically

In line with Park et al. (2019), we scale the output of the difference attention with a trainable parameter  $\gamma$  and apply a residual connection:

$$h_i = \gamma \cdot \text{Attention}(v_i, c_i, v_i) + v_i \quad (1)$$

This simple modification to the keys of the self attention heads takes the idea of difference images from Park et al. (2019) and implements them in a similar way as cross-modal attention in V&L transformers (Tan and Bansal, 2019; Lu et al., 2019).

We hypothesize that, to fully leverage the power of difference attention, more heads, i.e. more visual inputs for a specific change, might be beneficial for grounding and generating utterances. Thus we increase the number of difference attention heads to  $H = 8$ , where  $v_H$  is the *after* image, and we compute “in-between image features” for the additional heads as  $v_t = v_1 + c_t$

Intuitively, the “in-between images” represent the trajectory from the *before* to the *after* state (see Figure 2). Formally, we define  $c_t$  as the weighted difference features, where the weight is the relative position in the trajectory between  $v_1$  and  $v_H$ . Thus, each attention head receives image features representing a different degree of the visual change given by  $v_H - v_1$  and accordingly a varying degree of difference features for  $\mathbf{K}$ , where the first head at  $i = 1$  receives no difference features and the final head at  $i = H$  receives the whole difference features as given by the following equation:

$$c_i = \frac{i-1}{H-1} \cdot (v_H - v_1) \text{ where } i \in [1, H] \quad (2)$$

Finally, a single-layer feed forward network maps from the high-dimensional visual image space  $2048 \times 14 \times 14$  to the reduced visual word space of 512 dimension  $\hat{h}_i = r(h_i)$  and a downstream standard transformer receives the stacked sequence of visual words that represent various levels of change as  $V = [\hat{h}_1; \dots; \hat{h}_H]$ .

The number of attention heads  $H$  is a hyperparameter, which corresponds to the granularity of the simulated visual trajectory  $\{v_1, \dots, v_t, \dots, v_H\}$  where later images contain more changes from the *before* image  $v_1$ . We report results for 2 and 8 heads, leaving further experimentation for future work. As baselines, we implement two standard transformers that self-attend to the image pair (**TF-self-att-2**) and to the in-between images (**TF-self-att-8**). These are compared to **TF-diff-att-2** and **TF-diff-att-8** correspondingly, the transformers with difference attention.<sup>1</sup>

We encode the *before* and *after* images with a pre-trained ResNet-101 (He et al., 2016) and, optionally, transform it into a sequence with in-between images. This trajectory is passed through a difference attention layer, to obtain a sequence of visual words (see Figure 2). We apply positional encoding to the visual words, as in the standard transformer. These are further processed within the 6 layers of the multi-head-attention-based transformer encoder. In the decoder, an embedding layer first maps the words to vectors and then applies masked-self-attention followed by encoder-decoder attention which relates the visual words to words in the output sequence. In this architecture, difference and self-attention are used consecutively one after the other. In future work, further combinations can be investigated.

### 3 Experiments

#### 3.1 Data

**BLOCKS** (Bisk et al., 2016) is a dataset of movement instructions for blocks on a simple virtual 3D board (see Figure 1). The image pairs have been generated by down-sizing MNIST images, decorating the resulting blocks with digits or brand logos and randomly move the block’s pixels to other positions, one at a time. This sequence in reverse order corresponds to an action sequence for assembling a block configuration that visually represents a number. While BLOCKS was originally designed for instruction following, Rojowiec et al.

(2020) analyze its use for instruction giving. We use the MNIST-logo subset with constellations of up to 20 cubes with distinct logos. It is split into 667/95/181 image pairs for training, validation and testing and 6003/855/1629 captions respectively (9 per image pair).

**Spot-the-Diff** (Jhamtani and Berg-Kirkpatrick, 2018b) provides pairs of similar images extracted from real-word surveillance videos. The image pair shows a scene from the same viewpoint in different, but similar states (according to  $L_2$  distance) resulting in very subtle differences that are difficult to spot. Thus, Jhamtani and Berg-Kirkpatrick (2018b) collected descriptions of these pairs via crowdsourcing and instructed workers to “carefully study the image”, “give sufficient time as some difference may not be obvious” and to provide complete English sentences for each difference. We use the entire dataset of 9524/1634/1404 image-pairs for training, validation and testing and 17676/3310/2107 captions respectively. When an image-pair has less than 3 captions, we re-sample from the given ones, so that during training each pair is seen 3 times per epoch.

#### 3.2 Training and Hyperparameters

We encode the *before* and *after* image separately using a pre-trained ResNet-101 with the last layer cut off which results in image embeddings of size  $2048 \times 14 \times 14$  by applying average pooling. The word embedding layer in the transformer decoder is trained from scratch with a size of  $d = 512$ . We use the Adam optimizer with a learning rate of  $10^{-4}$  and a batch size of 8/16 for training with 8/2 heads respectively. We also perform early stopping after 5 epochs without improvement on the validation set and apply *Label Smoothing* as proposed by Vaswani et al. (2017).

For BLOCKS, it turned out to be necessary to fine-tune the image encoder to recognize the small logos distinguishing the single blocks. The training regime on BLOCKS is a two-stage process: the models (DUDA and our transformer models) are first trained with a frozen, pre-trained image encoder, and then trained fully together to fine-tune the image encoder for this particular task. For Spot-the-diff, we do not fine-tune the image encoder to ensure comparability with previous work.

<sup>1</sup>Code <https://github.com/clp-research/diff-att-transformer>

### 3.3 Evaluation

As the instructions in BLOCKS require detailed descriptions of block configurations, they commonly contain references to target and landmark objects, e.g. *heineken block right of the Burger King block* in Figure 1. If an instruction in BLOCKS does not mention the single correct target, a potential follower will not be able to execute it in any way. For landmarks, there might be several blocks mentioned by different crowd-workers. Since the blocks are generally referred to their logos, the targets in BLOCKS can be detected in human and generated captions with a simple, rule-based instruction parser (Rojowiec et al., 2020). In Spot-the-diff, there might be several target objects referred to by a more complex vocabulary, e.g. *additional people* in Figure 1. The dataset does not provide a language-external annotation for ground-truth target objects and they cannot be easily detected in an automatic way.

We measure the overlap of generated and human captions with BLEU-4, METEOR, CIDEr and SPICE, using the API of Chen et al. (2015). Furthermore, for BLOCKS, we rely on Rojowiec et al. (2020)’s parser which detects expressions (phrases) referring to targets and landmarks in ground-truth and generated instructions. Following Rojowiec et al., we compute these word or phrase accuracies: (i) **target**: correctly generated targets, given all generated target phrases (ii) **landmark**: correctly generated landmarks, mentioning one of the landmarks logos from the set of landmarks found in the ground-truth instructions (iii) **spatial**: correctly generated words not contained in target and landmark phrases, as a simple metric for measuring overlap of spatial expressions.

## 4 Results

Qualitative samples of generation outputs are shown in Figure 1 and in the Appendix.

### 4.1 General performance

Table 1 shows the results for instruction generation on BLOCKS: the TF-diff-att-8 transformer achieves the best performance on all metrics. It outperforms the baseline transformers with self attention (TF-self-att-2/8) by a considerable margin. It also clearly improves two state-of-the-art baselines for instruction generation and change captioning. We note that our version of DUDA trained on BLOCKS improves considerably over the results

Model	B	M	C	Target	Landm	Spatial
LSTM+Att*	0.38	0.28	0.27	0.11	0.28	-
DUDA	0.53	0.37	0.96	0.59	0.42	0.66
TF-self-att-2	0.34	0.28	0.35	0.19	0.26	0.76
TF-self-att-8	0.44	0.32	0.66	0.37	0.45	0.72
TF-diff-att-2	0.55	0.38	1.06	0.73	0.40	0.80
<b>TF-diff-att-8</b>	<b>0.68</b>	<b>0.43</b>	<b>1.52</b>	<b>0.86</b>	<b>0.73</b>	<b>0.83</b>

Table 1: BLOCKS results: B(LEU-4), M(eteor), C(ider) and word accuracies (see Section 3.3), LSTM+Att\* as reported in Rojowiec et al. (2020).

Model	B	M	C	S
DUDA*	0.081	0.115	0.34	-
FCC*	0.099	0.129	0.368	-
SDCM*	0.098	0.127	0.363	-
DDLA*	0.085	0.12	0.328	-
M-VAM + RAF*	0.111	0.129	0.425	0.171
TF-self-att-2	0.109	0.135	0.777	0.197
TF-self-att-8	0.110	0.136	0.786	0.191
<b>TF-diff-att-2</b>	<b>0.117</b>	<b>0.137</b>	<b>0.843</b>	<b>0.205</b>
TF-diff-att-8	0.113	0.136	0.842	0.202

Table 2: Spot-the-diff results: B(LEU-4), M(eteor), C(IDEr), S(PICE). \*Models as reported in Shi et al. (2020)

by Rojowiec et al. (2020), but not over our TF-diff models.

Results on Spot-the-diff are shown in Table 2. Generally, existing systems (mostly developed in the CV community) still obtain relatively low overlap scores on this task (with, e.g., BLEU scores around or below 0.1). Here, again, the difference attention transformers, TF-diff-att-2 and TF-diff-att-8, outperform the vanilla self-attention transformers. They also improve over the state-of-the-art set by the M-VAM model on Spot-the-diff, with a particularly strong increase of the CIDEr score (0.425 and 0.843 respectively). In contrast to BLOCKS, we see a small advantage of the TF-diff-att-2 over TF-diff-att-8. We will discuss this effect in detail in the following Section.

### 4.2 In-between images and landmarks

Results in Table 1 indicate that the accurate generation of landmark references is a harder task than spotting and referring to target objects. The competitive DUDA model achieves 59% acc. on targets and only 42% acc. on landmarks – an effect which has not been reported in the original DUDA paper by Park et al. (2019). This pattern is expected as the region of the target object is more or less ex-



plicitly represented in the difference image. The landmarks objects, on the other hand, do not move from the *before* to the *after* state and the model has to learn to attend to objects nearby the difference regions.

We observe that in-between images give a very clear performance boost for the realization of landmark references. Thus, the TF-diff-att-8 model improves the landmark accuracy of TF-diff-att-2 and DUDA by more than 30%, cf. Table 1. From this, we conclude that the in-between images combined with difference attention heads allow the transformer model to not only attend to target objects but also to “close-by” landmark objects, i.e. relating the *before* to the *after* image.

On Spot-the-diff, we do not find a clear positive effect of the in-between images, cf. Table 2. However, as discussed in Section 4.1, the differences between models on Spot-the-diff are generally much smaller than on BLOCKS, which likely results from the different nature of the two tasks: the main challenge in Spot-the-diff is to detect and accurately describe extremely small objects, that can be difficult to spot even for humans. At the same time, qualitative inspections of the actual descriptions in Spot-the-diff reveals that they contain much less complex spatial expressions or landmarks. Thus, our results on Spot-the-diff complement rather than contradict results on BLOCKS, and indicate that difference attention with in-between images is particularly helpful for grounding and generating linguistically complex landmark expressions.

### 4.3 Discussion

Our results are in line with other approaches showing the effectiveness of customized transformer architectures for complex linguistic-visual reasoning (Herdade et al., 2019; Cornia et al., 2020). Our difference attention is tailored to the landmark-based generation task, but generalizes to images from virtual (BLOCKS) and real environments (Spot-the-Diff), and is substantially simpler than, e.g., vision models for difference spotting (Shi et al., 2020). Approaches for video captioning (Zhou et al., 2018; Sun et al., 2019) predict key frames to describe things happening in a video with many frames. Our approach is complementary as we augment an image pair with only two frames to obtain in-between frames that are useful for grounding locative expressions and landmarks.

We took inspiration from the DUDA model (Park

et al., 2019) which dynamically attends to *before*, *after* and *difference* images during sequence generation. We carry this idea over to the transformer architecture which attends to all inputs simultaneously, by adding a difference-attention layer that allows the input of fine-granular visual changes between two images at once. Our results show that this approach performs better than dual attention or self-attention alone.

We observe that the different evaluation metrics yield roughly consistent model comparisons, i.e. models with lower overlap scores tend to achieve lower reference-related accuracies. It is worth noting though that the BLEU/Meteor score indicates smaller differences between certain models than the target accuracy: DUDA and TF-diff-att-2 seem to perform almost on par in terms BLEU and Meteor (see Table 1), but the target accuracy indicates that TF-diff-att-2 references are much more accurate. This underlines the fact that n-gram overlap scores in this NLG domain do not constitute a fully satisfactory approximation of instruction quality. An important direction for future work is to design interactive human evaluation settings for these tasks as standard off-line ratings might not be appropriate here (see examples in Appendix for illustration).

## 5 Conclusion

We investigate language generation for landmark-based instructions, and difference spotting. We proposed a simple difference attention head that relates consecutive images in an input trajectory via a difference key. Our method sets a new state-of-the-art on BLOCKS (Bisk et al., 2016) and Spot-the-diff (Jhamtani and Berg-Kirkpatrick, 2018b). Our findings are in line with Park et al., in that attention mechanisms based on image differences are highly effective for learning to reason for language generation from image pairs. We show that generating instructions with accurate landmark expressions is a challenging task for models at the intersection of Language & Vision, which can be tackled with customized attention mechanisms.

## Acknowledgements

We want to thank the anonymous reviewers for their comments. This research/work was partially funded by the *Deutsche Forschungsgemeinschaft* (DFG, German Research Foundation) – 423217434 (RECOLAGE) grant.

## References

- Yonatan Bisk, Deniz Yuret, and Daniel Marcu. 2016. [Natural language communication with robots](#). [Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies](#).
- Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. [arXiv preprint arXiv:1504.00325](#).
- Alasdair Daniel Francis Clarke, Micha Elsner, and Hannah Rohde. 2013. Where’s wally: The influence of visual salience on referring expression generation. [Frontiers in psychology](#), 4:329.
- Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. 2020. Meshed-memory transformer for image captioning. In [Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition](#), pages 10578–10587.
- Markus Dräger and Alexander Koller. 2012. Generation of landmark-based navigation instructions from open-source data. In [Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics](#), pages 757–766.
- Daniel Fried, Jacob Andreas, and Dan Klein. 2017. Unified pragmatic models for generating and following instructions. [arXiv preprint arXiv:1711.04987](#).
- Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. 2018. Speaker-follower models for vision-and-language navigation. In [Advances in Neural Information Processing Systems](#), pages 3314–3325.
- Davis Gilton, R. Luo, R. Willett, and G. Shakhnarovich. 2020. Detection and description of change in visual streams. [ArXiv](#), abs/2003.12633.
- 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](#), pages 770–778.
- Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. 2019. [Image captioning: Transforming objects into words](#). In [Advances in Neural Information Processing Systems](#), volume 32. Curran Associates, Inc.
- Harsh Jhamtani and Taylor Berg-Kirkpatrick. 2018a. Learning to describe differences between pairs of similar images. In [Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing \(EMNLP\)](#).
- Harsh Jhamtani and Taylor Berg-Kirkpatrick. 2018b. Learning to describe differences between pairs of similar images. In [EMNLP](#).
- John Kelleher and Geert-Jan M Kruijff. 2006. Incremental generation of spatial referring expressions in situated dialog. In [Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics](#), pages 1041–1048.
- Arne Köhn, Julia Wichlacz, Álvaro Torralba, Daniel Höller, Jörg Hoffmann, and Alexander Koller. 2020. [Generating instructions at different levels of abstraction](#). In [Proceedings of the 28th International Conference on Computational Linguistics](#), pages 2802–2813, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In [Advances in Neural Information Processing Systems](#), pages 13–23.
- Dipendra Misra, John Langford, and Yoav Artzi. 2017. [Mapping instructions and visual observations to actions with reinforcement learning](#). In [Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing](#), pages 1004–1015, Copenhagen, Denmark. Association for Computational Linguistics.
- Ariyo Oluwasanmi, Muhammad Umar Aftab, Eatedal Alabdulkreem, Bulbula Kumeda, Edward Y. Baagyere, and Zhiquang Qin. 2019. Captionnet: Automatic end-to-end siamese difference captioning model with attention. [IEEE Access](#), 7:106773–106783.
- Dong Huk Park, Trevor Darrell, and Anna Rohrbach. 2019. Robust change captioning. In [Proceedings of the IEEE International Conference on Computer Vision](#), pages 4624–4633.
- Robin Rojowiec, Jana Götze, Philipp Sadler, Henrik Voigt, Sina Zarriß, and David Schlangen. 2020. [From “before” to “after”: Generating natural language instructions from image pairs in a simple visual domain](#). In [Proceedings of the 13th International Conference on Natural Language Generation](#), pages 316–326, Dublin, Ireland. Association for Computational Linguistics.
- Raphael Schumann and Stefan Riezler. 2021. [Generating landmark navigation instructions from maps as a graph-to-text problem](#). In [Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing \(Volume 1: Long Papers\)](#), pages 489–502, Online. Association for Computational Linguistics.

Xiangxi Shi, Xu Yang, Jiuxiang Gu, Shafiq R. Joty, and Jianfei Cai. 2020. Finding it at another side: A viewpoint-adapted matching encoder for change captioning. *ArXiv*, abs/2009.14352.

Kumar Shridhar, Harshil Jain, Akshat Agarwal, and Denis Kleyko. 2020. [End to end binarized neural networks for text classification](#). In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 29–34, Online. Association for Computational Linguistics.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. [VideoBERT: A Joint Model for Video and Language Representation Learning](#). In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7463–7472. ISSN: 2380-7504.

Hao Tan and Mohit Bansal. 2019. [LXMERT: Learning Cross-Modality Encoder Representations from Transformers](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. 2018. [End-to-End Dense Video Captioning with Masked Transformer](#). In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8739–8748. ISSN: 2575-7075.

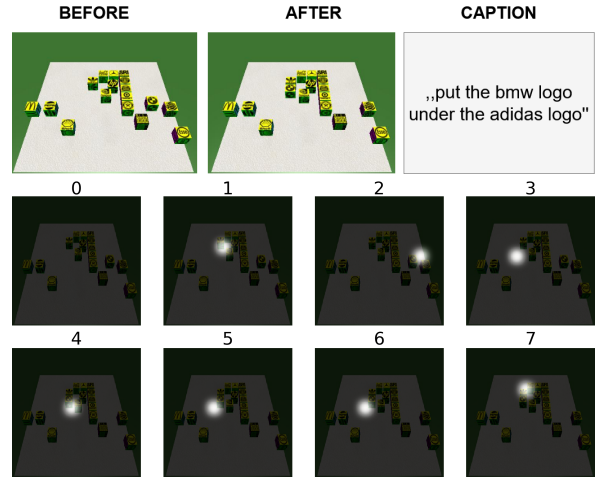


Figure 3: TF-diff-att-8: example caption and attention map on BLOCKS

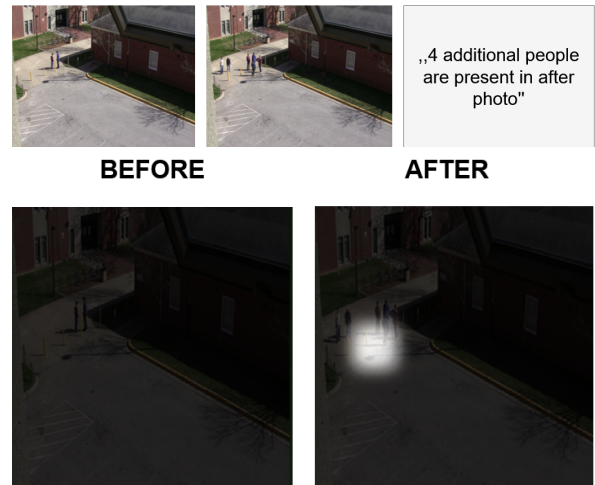


Figure 4: TF-diff-att-2 attention map on Spot-the-diff for the example from Fig. 1

## A Appendix



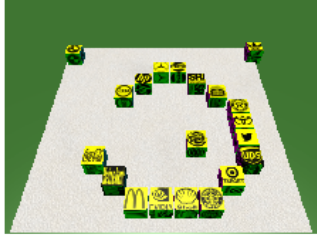
### A.1 Attention maps

Figure 3 shows an attention map for the TF-diff-att-8 model on BLOCKS. The map suggests that the model was able to precisely locate target and landmark objects. Figure 4 shows an example of a very accurate attention map computed by the TF-diff-att-2 model on Spot-the-diff.

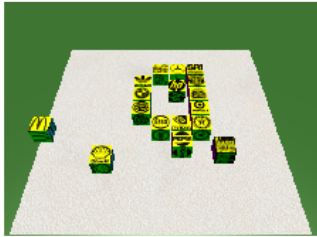

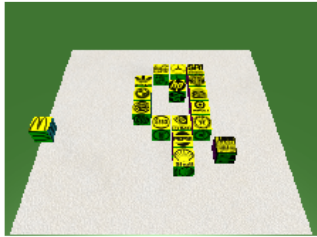
## A.2 Qualitative results

We present six generated example outputs drawn from the testset. We show one of the references along with the text generated by the **TF-diff-att-8** model. Additionally, the visual input is presented as the *before* image on the left, the *after* image on the right and an *in-between* image in the middle ( $t = 4$ ).




Example 1

<b>Reference</b>	place the coca cola box so that it's lower right corner touches the upper left corner of the heineken box		
<b>TF-diff-att-8</b>	move the coca cola block diagonally above and to the left of the heineken block.		
	8_num0_16.png 		8_num0_17.png 

Example 2

<b>Reference</b>	shell moves two spaces left and 3/4 space down from heineken.		
<b>TF-diff-att-8</b>	place the shell block south of the pepsi block.		
	99_num9_13.png 		99_num9_14.png 

Example 3 (wrong landmark)

<b>Reference</b>	place adidas so it's right side completely touches coca-cola.		
<b>TF-diff-att-8</b>	put the adidas block in the first open space to the left of the mercedes block.		
	39_num3_19.png 		39_num3_20.png 



Example 4

<b>Reference</b>	esso is placed so that its upper right corner is touching the lower left corner of nvidia
<b>TF-diff-att-8</b>	move esso so it is below and to the left of nvidia
	<div>88_num8_08.png</div>

Example 5 (wrong)

<b>Reference</b>	place the esso block in the same column as the coca cola block, one and a half block spaces above the coca cola block.
<b>TF-diff-att-8</b>	place the shell block north of the stella artois block.
	<div>8_num0_00.png</div>

Example 6 (wrong landmark)

<b>Reference</b>	stella artois is placed directly to the left of texaco
<b>TF-diff-att-8</b>	move the stella artois block to the left of the target block.
	<div>88_num8_06.png</div>