Connecting Vision and Language via Image Retrieval and Captioning

by

Mehrdad Hosseinzadeh

A Thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science
The University of Manitoba
Winnipeg, Manitoba

© Copyright 2021 by Mehrdad Hosseinzadeh

Thesis advisor
Yang Wang

Author
Mehrdad Hosseinzadeh

Connecting Vision and Language via Image Retrieval and Captioning

Abstract

Many real-world problems involve jointly understanding vision and language, e.g. image/video captioning, multi-modal image retrieval, visual question answering. In this thesis, we consider several problems in cross-modal learning from vision and language. First, the problem of composed query image retrieval is studied. In this problem, the objective is to pick the most related images to a given query in a pool of images. The query in this problem consists of a reference image and a modification text describing the desired changes to the reference image. Next, we visit the problem of video captioning for a future event. In this setting, the goal is to take what has happened in a video stream so far and describe what is the most likely to happen next. Moving forward, we focus on the problem of image captioning in two different scenarios. In the first scenario where we call "image change captioning", the task consists of taking two very similar images as input and describe the (subtle) difference between them. In the second scenario, the problem of personalized image captioning is studied in a few-shot settings. Unlike traditional image captioning where a generic sentence is generated for a given user, in our setting, the personality of the user is taken into account for caption generation. However, since collecting data for such a task is a

non-trivial problem, we study the few-shot setting. In this setting, for each new user (and hence new personality trait), only a few pairs of image-caption are available for quick adaption. We explore different ways to establish interactions between vision and language modalities and propose new methods to solve the aforementioned problems. Our proposed methods are evaluated on benchmark datasets for each problem and are compared with other state-of-the-art and/or baseline methods.

Contents

Abstract ii
Table of Contents v
List of Figures vi
List of Tables ix
Acknowledgments xii
Dedication xiii
Publications xiv

1 Introduction **1**

2 Related Work **5**
2.1 Image Retrieval 6
2.2 Captioning of Images and Videos 7
2.3 Change Detection 8
2.4 Future Frame Captioning 9

3 Composed Query Image Retrieval Using Locally Bounded Features **11**
3.1 Introduction 11
3.2 Proposed Method 14
3.2.1 Image Representation with Locally Bounded Features 15
3.2.2 Modification Text Features 18
3.2.3 Feature Fusion 19
3.2.4 Similarity Learning 21
3.2.5 Auxiliary Module 22
3.3 Experimental Setup and Results 24
3.3.1 Results on Fashion200K 25
3.3.2 Results on MIT States dataset 28
3.3.3 Results on CSS Dataset 28
3.4 Ablation Studies and Discussions 30
3.5 Conclusion 32

4 Video Captioning of Future Frames 33
4.1 Introduction . . . 33
4.2 Problem Statement . . . 35
4.3 Proposed Approach . . . 37
4.3.1 Implementation Details . . . 43
4.4 Experiments and Results . . . 44
4.4.1 Datasets . . . 44
4.4.2 Evaluation Metrics . . . 47
4.4.3 Compared Methods . . . 47
4.4.4 Results . . . 48
4.4.5 Ablation Study . . . 49
4.5 Conclusion . . . 51

5 Image Change Captioning by Learning from an Auxiliary Task 53
5.1 Introduction . . . 53
5.2 Background . . . 56
5.3 Our Approach . . . 59
5.3.1 Joint Primary and Auxiliary Networks . . . 60
5.3.2 Model Training . . . 63
5.4 Experimental Results . . . 67
5.4.1 Dataset and Setting . . . 67
5.4.2 Results . . . 69
5.4.3 Ablation Study . . . 70
5.5 Conclusion . . . 72

6 Few-Shot Personality-Specific Image Captioning via Meta-Learning 75
6.1 Introduction . . . 75
6.2 Problem Setup . . . 78
6.3 Our Approach . . . 78
6.3.1 Model Architecture . . . 79
6.3.2 Meta-Training . . . 81
6.3.3 Few-Shot Adaptation to New Personality Traits . . . 82
6.4 Experiments . . . 84
6.4.1 Dataset and Implementation . . . 84
6.4.2 Experimental Results and Comparisons . . . 87
6.4.3 Ablation Studies . . . 89
6.5 Conclusion . . . 95

7 Conclusion and Future Work 96

Bibliography 118

List of Figures

3.1 Overview of the composed query image retrieval. The query consists of a reference image and a sentence describing the changes one wants to be applied for retrieved images. The output is a set of images that are most similar to the query with the requested changes. 14
3.2 Overview of the proposed method. Light blue area (middle) is the main network, light pink (bottom left) is the pretrained visual feature extractor, and light yellow (right) is the auxiliary module, helping the main network to learn a better representation of query and target image by imposing the additional objective function. After a set of region features are extracted for source and target images using the visual extractor several layers of self-attention is applied in *TEP* and *VEP* on modification text and images, respectively. A joint representation of source image and modification text is computed in the cross-modal module using a special form and scaled dot product attention mechanism. The auxiliary module can work as a standalone coarse retrieval network in testing (see Sec. 3.2.5). Best viewed in color. 16
3.3 During training auxiliary module predicts a weight vector $b(\mathcal{I}, \mathcal{M})$ representing the importance of each feature in the vector representation of query and target images, given the requested modifications according to Eq. 3.23. During inference, this module can be activated to reject the distant candidates for the given query at an early stage (see Sec. 3.2.5). 23
3.4 Some qualitative examples of our method. Each row shows the query image, the modification text, and retrieved images. The examples are from Fashion200K, MITStates, and CSS3D datasets, respectively. . . 32

4.1 Given a sequence of frames of what is happening (E_t), the task is to anticipate what will happen in the next event (future) and describe it using a sentence. First row: the general version of the problem in which the task is to generate all the words in the caption for the next event. Second row: conditional future captioning where the task is to take the current event as well as a noun phrase defining the actor of the next event (*e.g.* the athlete in this case) and describe what the actor will do in the next event. 35

4.2 After training TFP, it is plugged into the main pipeline for the next stage of training. To train the main pipeline, a pre-trained C3D network computes the convolution features for each event (yellow box). Extracted features are then fed into the TFP to obtain the predicted features for the next event (green box). After temporal pooling, the predicted features are combined with the contextual features coming from the current event ($\oplus$) and are input to the captioning module using a projection layer. The captioning module adds the visual features to the word embedding features for each input word, applies n layers of mask convolution to effectively increase the receptive field for each word. The next word is generated by using softmax on top of the classification layer in the captioning module (blue box). 36

4.3 Training of TFP module on a pair of consecutive events at t and t' ($t < t'$). For each event, 3D convolution features are computed using a pre-trained C3D network F_t and $F_{t'}$. A dynamic adaptive pooling layer (DAP) takes the first event's features, F_t and the temporal length of the next event, v', and resizes the first event temporally to match the temporal dimension of the next event. The features are fed into a network consisting of several temporal convolution layers (grey blocks) to obtain the predicted features for the next event. The temporal feature predictor network maintains the temporal dimension throughout the layers. Grey dashed arrows indicate weight sharing. 39

4.4 CIDEr score for different activity classes. While our proposed method works generally well, we found that it works best in the events relating to the sports and has moderate performance in more complex environments such as "*building & repairing furniture*", "*exterior repair, improvements, & decoration*", and "*exterior maintenance, repair, & decoration*". 46

4.5 Qualitative examples. GT and PR are the ground-truth and predicted caption for the next event. In the first and second examples, the proposed method accurately captions the next event. But it fails to describe properly in the third example. 50

5.1 (Best viewed in color) Given two very similar images, the goal of change captioning is to describe the subtle difference between these two images. The difference can be in terms of objects' color, texture, position, addition or removal, etc. 54
5.2 Overview of our approach. Given a triplet (A, B, C) from the training set, where (A, B) are image pairs and C is the caption describing their difference, our method involves jointly training two networks for the primary and the auxiliary tasks. The training involves two stages. In the first stage ("Primary $\rightarrow$ Auxiliary"), we feed (A, B) to the primary network to generate a caption $\hat{C}$, then feed $(A, \hat{C})$ as the input to the auxiliary network. In the second stage, we feed (A, C) to the auxiliary network to retrieve $\hat{B}$ from a set of S candidate images, then feed $(A, \hat{B})$ to the primary network. These two stages form a cycle consistency. The candidate set S is constructed differently depending on the training epoch. See the main text for details. 58
5.3 (Best viewed in color) Examples of two types of pairs in the dataset. (a) and (b) form a "changed pair" since there is a change at the object level (red $\rightarrow$ yellow). (a) and (c) form a "distractor" pair since there is only viewpoint change. 68
5.4 Qualitative examples. The first three rows depict the cases where there is a change at the object level between the two input images. The last example shows a case in which there is no semantic change between two images. 74

6.1 Most existing image captioning models produce generic captions stating obvious things in an image, e.g. "a women is cutting the cake" in this case. These captions are not very engaging for humans. In this work, we consider the personality-specific image captioning. We assume that we know the personality trait (e.g. "sweet", "anxious") of the user. Our goal is to produce image captions with a style that is consistent with the personality of the user. These personality-specific captions are more engaging to the user. 76
6.2 Model Architecture for ConvCap network. ConvCap uses VGG as its visual feature extractor, and replaces recurrent network for caption generation with several masked convolution layers. Refer to [6] for more details. 80
6.3 Qualitative examples of our method. (a) and (b) are cases where our method was successful in describing the image using the given trait. (c) shows an example where our method has described the image partially correct. (d) depicts a case where our method failed to generate a correct caption. GT and PR are ground-truth and predicted captions, respectively. Best viewed in color. 93

List of Tables

3.1 Results on the Fashion200K dataset. The numbers of other approaches are adopted from [117]. The proposed method outperforms other state-of-the-art approaches in terms of $Recall@K$ metrics. In particular, our proposed method gains a 26% performance boost over the previously best result in terms of $Recall@1$. 26

3.2 Results on the MIT States dataset. The numbers of other approaches are adopted from [117]. Our proposed method outperforms other state-of-the-art approaches in terms of $Recall@K$ metrics. In particular, our proposed method gain a 19.67% performance boost in terms of $Recall@1$. 27

3.3 Results on the CSS dataset. The numbers are adopted from [117]. $3D \rightarrow 3D$ is when the query and target images are both in 3D. $2D \rightarrow 3D$ denotes the setting when the query image is a 2D while the target image set is 3D. Our proposed method outperforms other emphasizing its generalization strength to other domains. 29

3.4 Ablation studies results on CSS (3D) dataset. First two rows exhibit the the additional loss function role in overall performance: the performance boosts when the additional loss function is added to the main loss function. Last two rows shows average rejection rate and overall performance when AM is used as an early rejection network in inference mode. 30

4.1 Performance of the proposed method compared to the baseline and oracle method on the first problem setting (i.e. general case). The hyperparameter λ controls the amount of visual contextual information added to the predicted features for the next event. Adding context substantially boosts the performance of the proposed method, especially CIDEr metric. 44

4.2 Performance of the proposed method on the conditional future captioning task compared to the baseline and oracle methods. The hyperparameter λ controls the amount of visual contextual information added to the predicted features for the next event. 45
4.3 Performance of the proposed method in the first problem setting (i.e. general case) using different values of λ. By increasing λ and therefore injecting more contextual information, we obtain better results. However when λ becomes bigger than 0.50, the performance starts to drop. 45

5.1 Performance of the proposed method on the entire CLEVR-Change dataset. Metrics indicated by "-" are not reported by the authors. Numbers are taken from respective works. Our proposed method improves the performance of DUDA which uses the same base network. B4, C, M, and R, S are BLEU-4, CIDEr, METEOR, and ROUGE-L, Spice, respectively. 65
5.2 Performance of the proposed method evaluated only on the changed pairs (top) vs. the performance evaluated only on the distractor pairs (bottom) on the CLEVR-Change dataset. B4, C, M, and R, S are BLEU-4, CIDEr, METEOR, and ROUGE-L, Spice, respectively. . . . 66
5.3 Performance of our method against DUDA and DDLA on the Spot-the-diff dataset. B4, C, M, and R are BLEU-4, CIDEr, METEOR, and ROUGE-L, respectively. 66
5.4 Performance of the proposed method when only providing the auxiliary network with the easy set of candidates (first row) vs. the performance when using a dynamic strategy and switching to hard sample sets after a certain epochs. Other settings remain identical in both cases. Our method benefits from the dynamic strategy and the result has been improved. B4, C, M, and R, S are BLEU-4, CIDEr, METEOR, and ROUGE-L, Spice, respectively. 67
5.5 Performance of the proposed method compared with other state-of-the-art approaches on each category of change. The changes categories are : Color Change (C), Texture Change (T), Adding an object (A), Deleting an Object (D), and Moving an Object (M). Numbers for other methods are taken from [82] 72

6.1 Performance of the proposed method on the Personality Caption dataset using our proposed data split protocol. The proposed method outperform the baselines for all the metrics while only using 10 labeled samples for adaptation. 85
6.2 Evaluating different aspect of our method via ablation studies. 86

6.3 Performance of our method when using explicit and implicit personality trait information. Using only implicit information is adequate for the model to pick up on specific clues associated with each personality trait caption style, and learn them using a few provided adaptation samples. 87
6.4 Comparing the performance of our model using different K values. The best performance is achieved using $K = 10$. However, the performance does not degrade severely when we use 5 samples for adaptation instead of 10. In contrast, one can see a significant drop in performance when only one sample is used for adaptation. 88
6.5 Personality traits used in meta-training and meta-testing. 95

Acknowledgments

First and foremost, I would like to thank my supervisor, Prof. Yang Wang, for guiding me through this journey. He is an outstanding researcher, a passionate supervisor, and a brilliant mentor. His thoughtful feedback and insightful comments paved the path for the success of the research in this thesis. I am deeply thankful to him for being always available for advice, discussion, and support. This thesis was not possible without his invaluable vision.

I am also grateful to my committee members, Prof. Carson Leung and Prof. Ho for their continuous and constructive feedback throughout the course of my Ph.D. program. Also, many thanks to Prof. Graham Taylor for serving as the external examiner for this thesis and holding productive discussions during the Ph.D. defense session.

To the support staff at the Computer Science department, at the University of Manitoba who was always ready to help, thank you. Special thanks to Lynne Hermiston for her help all through my journey.

I am fortunate to be surrounded by friends who were always encouraging and supportive. Particularly, I thank Shiva for helping me pick myself up when I was falling off the wagon. I am also deeply thankful to my labmates for the productive discussions we had and the interesting ideas we shared.

It cannot be expressed enough the extent to which I owe my parents and brother for being here. Their encouragement and unmitigated support have always pushed me forward and enlightened my path to success. Last but not least, I'd like to thank my beautiful wife, Hoora, for believing in me more than I did.

To Hoora for her unconditional love, belief, and support.

Publications

- **M. Hosseinzadeh** and Y. Wang. Composed Query Image Retrieval Using Locally Bounded Features. IEEE Conference on Computer Vision and Patter Recognition (CVPR), 2020.

- **M. Hosseinzadeh** and Y. Wang. Video Captioning of Future Frames. IEEE Winter Conference on Applications of Computer Vision (WACV), 2021.

- **M. Hosseinzadeh** and Y. Wang. Image Change Captioning by Learning from an Auxiliary Task. *IEEE Conference on Computer Vision and Patter Recognition (CVPR), 2021*

- **M. Hosseinzadeh** and Y. Wang. Few-Shot Personality-Specific Image Captioning via Meta-Learning. Under Review at *IEEE Transactions on Multimedia, 2021*

Chapter 1

Introduction

Computer vision has benefited from the recent and rapid progress of deep neural networks. Among them, image classification[132; 102; 42; 28] and image segmentation [80; 107; 67; 131], have raised the bar and in some cases even surpassed human evaluations [42]. At its core, computer vision aims to provide an interactive system for a "*user*". Linguistic modality, however, is the natural way that we as humans interact with each other. Arguably, humans prefer to communicate with their peers through natural language. As a result, from the perspective of an end-user, it is more engaging, if a computer vision system is capable to interact with them using natural language. This natural preference is the driving force behind this thesis.

In this thesis, we use this observation and explore avenues that linguistic modality can be incorporated to improve the end-user's interaction with computer vision systems. We utilize image captioning, video captioning, and image retrieval as settings in which the linguistic modality is incorporated.

With the availability of billions of images on the internet, image retrieval has

become an essential part of search engines. Traditionally, keywords were used as queries to retrieve related images. In fact, about one-third of searched conducted on Google are for images [71]. However, a keyword cannot always convey the desired query a user has in mind for retrieval. A recent study [116] reveals that more than 62% of millennials found visual search (i.e. image $\rightarrow$ image) more desirable than keyword search for image retrieval. Many image providers are already shifting their focus towards more interactive image searches. For instance, Pinterest has implemented an image $\rightarrow$ image search service that has over 600 million searches daily [73]. The drawback, however, is that when conducting an image $\rightarrow$ image search, the user must have a query image that exactly frames what he/she has in mind. This is not practical when the user wants to search for similar images that have or lack certain features. Combining an image query along with a textual description expressing the modifications that the user has in mind, can be very helpful in these situations.

Composed query image retrieval is a more interactive form of image retrieval where the user is capable to query an image and describe the changes that he/she want to be applied to the query. The system is supposed to retrieve any image that is similar to the query one but differs from it with regard to the entered description. this composed query does not limit the user to only query images similar to the one they need to be retrieved. In other words, the textual description makes the query system more engaging for the user. We propose a novel solution for this problem where every word in the input text is compared with every region of the image. Establishing an explicit relationship between parts of two modalities improves the performance of our method on several benchmark datasets.

Image/Video captioning is the other computer vision area where the goal is to bring users, especially those with special needs, a more interactive system. This is especially important for users with different disabilities such as visual impairments. According to a recent survey, people with disabilities are less likely to use digital devices, mainly due to the accessibility issues [17]. Therefore, conveying visual information via natural language seems to be the most natural way to increase accessibility. Facebook and Instagram add automatic image descriptions in the meta-data of the images uploaded to their website [22]. This enables the visually impaired users to be able to understand the visual contents and interact with the website. Microsoft has also integrated a similar tool into their Office products [114].

We investigate three different scenarios for image/video captioning. First, we propose and tackle the problem of future video captioning. In this setting, the goal is to anticipate what would most likely happen next, based on a sequence of frames in a video stream that has been observed so far. Next, a sentence is generated describing the likely-to-happen event in the near future.

In other applications such as home surveillance systems, it is important to detect subtle changes between two scenes and let the user know about the change. A natural way to interact with the user in such a case is to describe the change via a natural language sentence. This problem is referred to as image change captioning and is very challenging since the change between two images can be very subtle. Therefore, we need to make sure that our system learns the details of the problem as well. We propose a novel training scheme that leverages the auxiliary task notation. In other words, we couple the learning of the task of change captioning (i.e. the primary

task) with learning an auxiliary task (i.e. composed query image retrieval.) in a complementary fashion.

Finally, we argue that perhaps the most important part of an engaging multi-modal system is to speak to the user in "his/her language". Each and every user has a different personality that requires a unique interaction style. In the case of image captioning, this unique style can be seen as generating sentences that carry the tonality of the user's own personality. However, the personality of the user has been ignored in captioning systems so far. The main roadblock is that collecting a sufficiently large dataset that includes an order of thousand examples for each personality is almost impossible. We, therefore, propose an alternative that bypasses this roadblock and allows us to have a personalized image captioning system. Our method is based on the recent advances in few-shot learning techniques. Specifically, we propose to learn a network that can learn to adapt to a new user's personality given only a few annotated pairs of image-caption in the style of the personality. This is a much more feasible approach since we only need to ask the user to annotate a few images in his/her tone, before being able to acquire their style and serve to generate personalized captions.

Chapter 2

Related Work

There has been lots of work on computer vision tasks that involve multi-modal data (e.g. images and text), such as visual question answering (VQA) [5; 104; 8; 106], image captioning [115; 125]. VQA approaches take an image as the reference and try to answer textual questions about the image. Image captioning takes an image as the input and produces a textual description of the image. Xu et al. [125] introduce the notion of attention in the visual domain with application to image captioning. Recently, self-attention [110] has been popular in many computer vision tasks [106; 105; 94]. We also use variations of self-attention in our work to capture a richer representation of images and the modification text. Our work is related to two lines of research, namely, image captioning and change detection. We briefly review relevant works on these topics.

2.1 Image Retrieval

While traditional image retrieval approaches use manually designed features from the image [20], most modern image retrieval approaches use some form of deep learning [118; 44]. Depending on the type of queries, image retrieval falls into several categories. Content-based image retrieval (CBIR) is the problem setting in which the query is in the form of a single image. This setting has been extensively explored for the tasks of face recognition and product search [83; 96; 133]. Another line of research formulates CBIR as learning hashing codes from images such that the query and the corresponding retrieved image(s) have a smaller distance in a certain space (*e.g.* Hamming, Euclidean). Deep quantization network [16] aims to find more optimal hash codes for images by putting a constraint on the quantization error. Deep Cauchy hashing [15] uses a pairwise loss function in the Cauchy distribution which explicitly forces similar images to have a distance smaller than a certain radius in the Hamming space.

There is also work on using other modalities as the query for image retrieval. In [12; 89], the use of a coarse sketch of an image as the query is explored. This setting makes the problem more challenging yet more practical for users. By modeling the query as a textual input, Wang et al. [120] propose a dual network that learns to push the textual input and the corresponding image together in an embedding space. Our work is different from all these approaches in that our query is a reference image *and* a modification text requested by the user to be applied to the image. Vo et al. [117] propose the first work on using this type of composed queries for image retrieval.

2.2 Captioning of Images and Videos

Image captioning [77; 125; 5; 127; 7; 47; 24; 21] has been studied extensively in recent years. Vinyals et al. [115] propose one of the earliest methods for image captioning which utilize LSTM modules. Xu et al. [125] extend the former model by introducing visual attention. Visual attention has been proved to be an effective technique for the image captioning models and has been used extensively since then [24; 21; 47; 79].

While early works on image captioning mainly use recurrent modules (specifically LSTM layers) for the caption generation, there has also been work on using the convolutional or self-attention mechanism. Anega et al. [7] replace the LSTM module with a fully convolutional module that noticeably improves the training time of these networks. With the introduction of transformers and self-attention layers, BERT-based models [110; 26] have been shown to be successful for a variety of language tasks. These model have also been applied in vision-language tasks as well [106; 21; 47; 58; 60; 119].

There has also been lots of work on captioning in the video domain [57; 113; 112; 126; 128]. The method in [112] utilizes LSTM units in an encoder-decoder fashion for the video captioning task. It extracts both appearance and optical flow features of frames, then feds them through their proposed model to generate captions. Inspired by the success of self-attention [110] and transformer networks, [138; 140] propose end-to-end video dense captioning systems by establishing a more explicit relationship between visual and textual modalities. Another attempt to densely describe a video is made by [29]. It uses a cycle-consistency scheme to train the network without the

corresponding temporal annotations of events in the video. Zhang et al. [135] further utilize the syntax of the sentence to generate more plausible captions for videos. All of these approaches assume that we have access to the entire video.

Felsen et al. [30] study the problem of predicting the players' next moves in water polo and basketball videos. However, their approach is constrained to the special case of those sports and a limited set of moves. Additionally, there have been proposed approaches that aim to predict the future in a pixel-level space. Using generative adversarial networks, [2] has established a framework to predict the next frame(s) directly in RGB space based on what has been the sequence of frames so far. Other works [69; 70] have considered predicting future semantic or instance segmentation of future frames.

2.3 Change Detection

Detecting changes in an environment have been an active field of study in computer vision. There has been work on finding the change between two (or more) images. Change detection has been applied in aerial and satellite imagery [66; 109; 129] for applications such as natural disaster management [37], observing land dynamics [54], etc.

Most change detection tasks focus on finding changes in the pixel space. However, if we want the system to interact with users, it is more favorable to present the detected change in a human-readable form, such as describing the change using natural language. This has led to a new line of research called the image change captioning [49]. Early works in this area use the Spot-the-diff dataset [49] which has

two major flaws: 1) it assumes that there is always a change between each pair of images and 2) it is a relatively small dataset (∼13K images in total). To overcome this issue, Park et al. [82] propose a new problem called the robust change captioning and introduce a new larger dataset called the CLEVR-Change that is based on the popular CLEVR engine [52]. This dataset better evaluates the performance of change captioning systems since some of the pairs in the dataset are the same image from different viewpoints. They also propose an attention-based model called DUDA. Recently, Shi et al. [100] propose a method to simulate human visual attention for better localization of the change.

2.4 Future Frame Captioning

Future frame captioning is related to several lines of research in computer vision, including image/video captioning, future prediction, and moment/event detection in videos. In image captioning, the goal is to generate a sentence describing an input image. Existing image captioning models often use recurrent neural networks (RNNs), specifically the long-short term memory (LSTM) variant [43]. Karpathy et al. [53] tackle the problem using a combination of CNNs for extracting features from the image and an RNN for generating the caption. Some works [125; 6; 68] use an attention mechanism to focus on visual features from different parts of the input image when generating each word.

For the future frame captioning, on the other hand, the goal is to use visual information for current frames to anticipate the most likely scenario that follows and describe it using a sentence. To the best of our knowledge, we are the first to explore

this setting.

Chapter 3

Composed Query Image Retrieval Using Locally Bounded Features

3.1 Introduction

Image retrieval [98; 64] problem has always been at the heart of computer vision research for its practical applications in query-based systems. Image retrieval systems can be used for many downstream tasks, such as person re-identification [124; 59; 32] and product search [39; 1]. A challenge of image retrieval is how to formulate the query in a way that captures the user's intention as much as possible. A *de facto* paradigm in image retrieval systems is to take a query/reference image, process it and return a set of candidate images as the most similar ones to the input query.

Despite the simplicity of formulating a user query this way, it suffers from a fundamental problem – it requires users to express precisely what they have in mind using a single image. In many cases, it is not practical to assume that the user's

intention can be conveyed using a single image as the query. In this chapter, we consider the composed query image retrieval problem first introduced in [117]. In this problem setting, the query to an image retrieval system consists of an image and the desired modification expressed in terms of a sentence. This sentence states the changes a user wants to be applied to the query image. This setting gives the user the flexibility to express their intention in a more natural and meaningful way – the user does not need to express his/her query using only a query image. We call it the *composed query* since the query is composed of a reference image and an accompanying modification text. Figure 3.1 shows an illustration of the composed query image retrieval setting.

A modification text usually refers to one or more "*entities*" in the image that should be changed. For example, in Fig. 3.1, the desired change ("make bottom right gray object purple") is only related to a few entities in the image (the bottom right object in this case). This is the key motivation for our proposed method. In contrast to other approaches in the image retrieval domain [117; 98] which consider an image as a whole, we propose to treat an image as a set of local "*entities*". We argue that grounding the input modification text to different semantic areas in the reference image is crucial for the composed query image retrieval problem.

To this end, our proposed method first extracts the features for a set of local areas in the image. We name each of these local regions an "*entity*". The set of features and the modification text are then processed using separate branches with self-attention layers. Later a cross-modal module learns a joint representation of the query image and the modification text by leveraging attention mechanism to correlate each word

to each entity in the image. During testing, a candidate target image is processed and represented through its entities' features. The joint representation of the query and the target images are then compared to each other for retrieval. Our proposed method also includes an auxiliary module that enhances the representation learning process through an additional objective function. The formulation of this module allows us to use it as a standalone coarse retrieval network during inference. This module can be used to quickly filter out the most dissimilar candidate images for each query without passing them to the main pipeline.

The contributions of this work are manifold:

- We propose a novel approach for the composed query image retrieval task. Different from previous work [117] that represents an image as a whole, our method considers the image as a set of local semantic entities. This allows our method to effectively a capture detailed relationship between the modification text and each local entity in the image.

- We propose an auxiliary module that further improves the performance. It can also be used to improve the efficiency during testing by filtering out candidate images without sacrificing too much in terms of the accuracy.

- The proposed method is extensively evaluated on three benchmark datasets and consistently outperforms other state-of-the-art approaches.

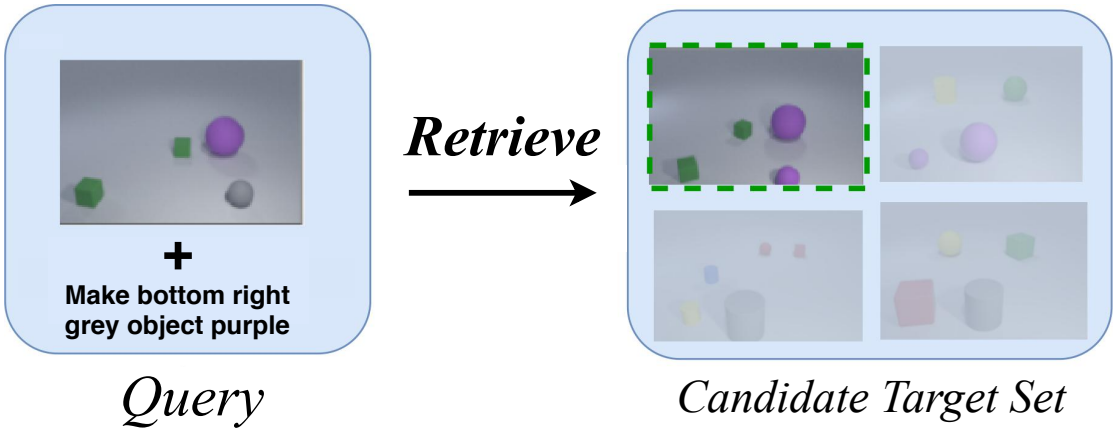


Figure 3.1: Overview of the composed query image retrieval. The query consists of a reference image and a sentence describing the changes one wants to be applied for retrieved images. The output is a set of images that are most similar to the query with the requested changes.

3.2 Proposed Method

Let $(\mathcal{I}, \mathcal{M}, \mathcal{I}_t)$ be the query image, the modification text, and a candidate target image, respectively. Our proposed method first defines and extract features for a set of local regions in the image. We use these features as the representation of the image, $\mathcal{I}$. Then we learn a joint representation $f(\mathcal{I}, \mathcal{M})$ that captures the visual and linguistic information from $(\mathcal{I}, \mathcal{M})$. This is achieved using self and cross modal attention mechanisms, correlating each word in the modification text to each region in the image. We also learn a feature representation $g(\mathcal{I}_t)$ for the target image. For the ground-truth target image, $\mathcal{I}_t$, the two vectors $f(\mathcal{I}, \mathcal{M})$ and $g(\mathcal{I}_t)$ are expected to be similar. Additionally, we also proposed an auxiliary module (AM) that helps the model to learn better representations for query and target by imposing an $L2$ loss on

the learnt representations. The formulation of AM allows us to use it as a standalone coarse retrieval network; it can reject the most dissimilar target candidates for each query in an early stage and without entering the main pipeline. During inference, AM first predicts an importance vector based on the joint representations of the query image and modification text. Based on the predicted vector, it then computes a weighted representation of each target candidate. Finally, those candidates whose weighted representations are in distant with that of joint representation of query by a predefined threshold, are rejected and not entered into the next fine retrieval stage. Empirically we show this simple strategy can effectively filter out more than 60% of test candidates per query. Figure 3.2 depicts the overall architecture of our method.

3.2.1 Image Representation with Locally Bounded Features

Previous approaches for image retrieval tasks usually consider the entire image as a *single* entity, *i.e.* processing the entire image at once using a CNN [117; 44]. While this works well in the traditional image retrieval settings, the composed query image retrieval problem requires a richer and more detailed understanding of the image. In this chapter, we propose to divide the image into locally bounded entities and process the image at the region level.

Region Visual Features: Given an input image $\mathcal{I}$, we first apply a pre-trained region proposal network [90] to extract K regions in the image. Each region is then represented as a CNN feature vector, i.e. $\mathcal{I} = \{e^1, e^2, ..., e^K\}$ where $e^i \in \mathbb{R}^{d_e}$ ($d_e = 2048$) is the feature vector of the i-th region.

Region Positional Features: Composed queries often contain positional words

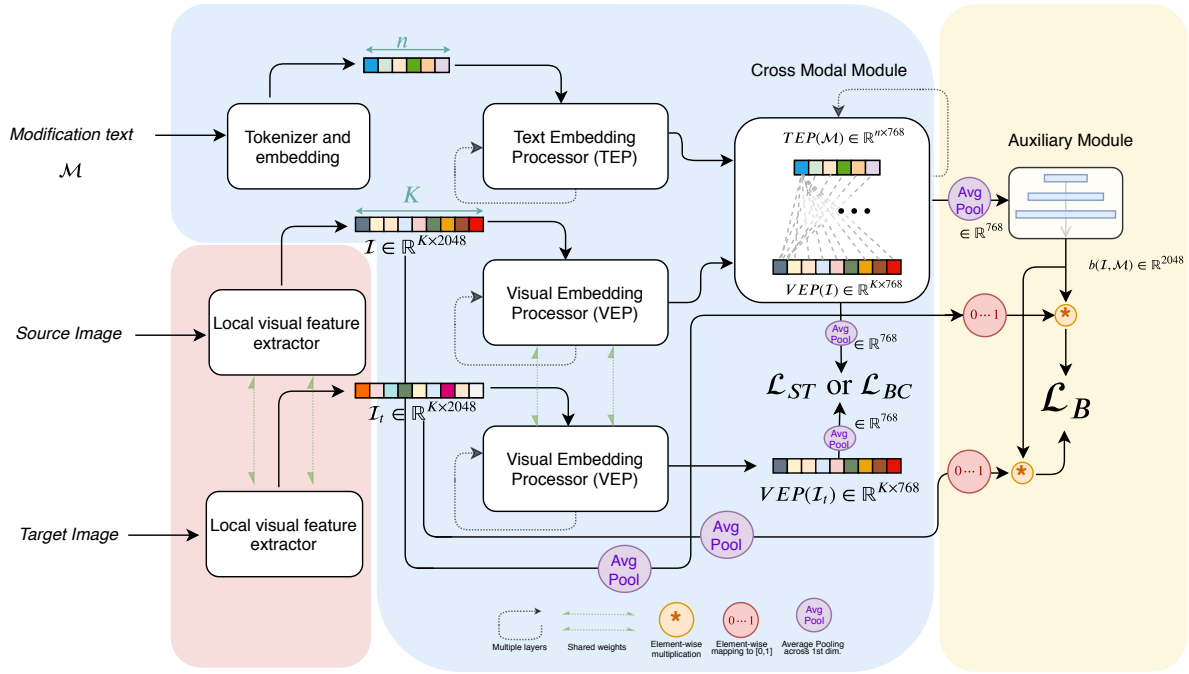


Figure 3.2: Overview of the proposed method. Light blue area (middle) is the main network, light pink (bottom left) is the pretrained visual feature extractor, and light yellow (right) is the auxiliary module, helping the main network to learn a better representation of query and target image by imposing the additional objective function. After a set of region features are extracted for source and target images using the visual extractor several layers of self-attention is applied in *TEP* and *VEP* on modification text and images, respectively. A joint representation of source image and modification text is computed in the cross-modal module using a special form and scaled dot product attention mechanism. The auxiliary module can work as a standalone coarse retrieval network in testing (see Sec. 3.2.5). Best viewed in color.

(e.g. "*replace the oval right to the circle with a red triangle.*"). For this task, it is important to effectively represent the layout of the image and the spatial relationships

between different objects in the image. In order to capture the spatial information of each region, we calculate a positional feature vector $p^i \in \mathbb{R}^{d_p}$ encoding the normalized (x-location,y-location, width, height) information of the i-th region as:

$$p^i = Linear\Big([N(x^i), N(y^i), N(w^i), N(h^i)]\Big) \tag{3.1}$$

where $[\cdot]$ is concatenation operator, (x^i, y^i, w^i, h^i) denote (x-location, y-location, width, height) of the i-th region. $N(\cdot)$ normalizes its input between 0 to 1. We then use a linear layer to map the result to a d_p-dimensional vector ($d_p = 2048$).

Image Representation: Finally, we average the visual and positional features for each region, and pass through a linear layer to change the feature dimension for each region to $d_v = 768$. Then we use a self-attention based multi-layer visual embedding processing (VEP) module to get the final feature representation of the image $V(\mathcal{I})$:

$$c^i = Linear(avg(e^i, p^i)) \tag{3.2}$$

$$\mathcal{C}_1 = \{c^1, c^2, ..., c^K\} \tag{3.3}$$

$$V(\mathcal{I}) = VEP(\mathcal{C}_1) \tag{3.4}$$

where $\mathcal{C}_1$ is the input to the first layer of VEP. Generally, the l-th layer of VEP takes the output of previous layer $(l-1)$ as the input, then applies a scaled dot-product attention (as self–attention mechanism) [110; 26] and passes it through a linear layer to generate the input for the next layer:

$$\mathcal{C}_{l+1} = Linear(SA(\mathcal{C}_l)), \text{where} \tag{3.5}$$

$$SA(C_l) = Softmax\left(\frac{\mathcal{C}_l {\mathcal{C}_l}^T}{\sqrt{d_v}}\right)\mathcal{C}_l \tag{3.6}$$

where $SA(\cdot)$ denotes the self attention operation, $C_l, C_{l+1} \in \mathbb{R}^{K \times d_v}$ and $d_v = 768$ is the feature dimension of VEP. In the end, the output of the last layer of VEP is used

as the image representation $V(\mathcal{I}) \in \mathbb{R}^{K \times d_v}$. We can then perform an average pooling over the first dimension of $V(\mathcal{I})$ to obtain a feature vector as:

$$g(\mathcal{I}) = Pool(V(\mathcal{I})) \tag{3.7}$$

where $Pool(\cdot)$ denotes the average pooling operation and $g(\mathcal{I}) \in \mathbb{R}^{d_v}$ is the visual feature vector of the image $\mathcal{I}$.

3.2.2 Modification Text Features

In this section, we introduce a textual embedding processing (TEP) module that processes the composed query sentence $\mathcal{M}$ which is a sequence of n words. We start with tokenizing the sentence using WordPiece [123; 26] to obtain the split word list $\{w^i\}_{i=1}^n$. Each word and its absolute position in the sentence are then mapped to a vector of size $d_w = 768$ (*i.e.* the same dimension as $g(\mathcal{I})$) using two separate embedding layers, namely $Emb(\cdot)$ and $\mathcal{P}(\cdot)$, respectively. The final representation for the i-th word in the sentence is then $w_e^i = Emb(w^i) + \mathcal{P}(w^i)$. The initial input to TEP is then the sequence of word representations $\mathcal{W}_1 = \{w_e^i\}_1^n$. Similar to the visual embedding module, the textual embedding processing module consists of multiple layers where each layer is a self-attention module followed by a linear transformation to shape the final representation. The output of each layer in TEP is the input to the next layer:

$$\mathcal{W}_{l+1} = Linear(SA(\mathcal{W}_l)), \text{where} \tag{3.8}$$

$$SA(\mathcal{W}_l) = Softmax\left(\frac{\mathcal{W}_l {\mathcal{W}_l}^T}{\sqrt{d_w}}\right)\mathcal{W}_l \tag{3.9}$$

$T(\mathcal{M}) \in \mathbb{R}^{n \times d_w}$ is the output of the last layer of TEP.

3.2.3 Feature Fusion

For the composed query image retrieval task, the query consists of a reference image and a modification text expressed as a sentence. It is important to have an effective way of integrating the information from these two different modalities. The method in [117] directly combines the feature vector of the entire query sentence with the feature vector of the entire image. We argue that this is not the most effective way to perform the fusion. Intuitively, the composed query image retrieval task requires a detailed understanding of the linguistic information of the words and the visual information in different regions in the image. In this section, we incorporate a cross-modal attention module to fuse these two modalities.

The cross-modal attention module consists of L layers in which language and visual features are fused. Each layer (except the last layer) consists of two parallel similar sub-modules (with independent weights) processing visually attended language features and linguistically attended visual features. Intuitively, these two sub-modules progressively generate a richer representation of language and visual features. This results in a joint representation (denoted as $f(\mathcal{I}, \mathcal{M})$) of the source image and the modification text. We use $V_0(\mathcal{I}) \in \mathbb{R}^{K \times d_v}$ to denote the visual features of regions in the image $\mathcal{I}$ (Sec. 3.2.1) and $T_0(\mathcal{M}) \in \mathbb{R}^{n \times d_w}$ to denote the linguistic features of the modification text $\mathcal{M}$ (Sec. 3.2.2). More specifically in the l-th layer ($l = 0, 1, ..., L-1$) of this module, the linguistically attended visual features are computed as follows:

$$\widehat{V}_l = CA\Big(V_l(\mathcal{I}), T_l(\mathcal{M})\Big), \text{where} \tag{3.10}$$

$$CA\Big(V_l(\mathcal{I}), T_l(\mathcal{M})\Big) = Softmax\Bigg(\frac{V_l(\mathcal{I})T_l(\mathcal{M})^T}{\sqrt{d_v}}\Bigg)T_l(\mathcal{M}) \tag{3.11}$$

$$V_{l+1} = Linear\Big(SA\big(\widehat{V}_l\big)\Big), \text{where} \tag{3.12}$$

$$SA\big(\widehat{V}_l\big) = Softmax\left(\frac{\widehat{V}_l\widehat{V}_l^T}{\sqrt{d_v}}\right)\widehat{V}_l \tag{3.13}$$

where $V_l(\mathcal{I}) \in \mathbb{R}^{K\times d_v}$ is the language attended visual features input of the l-th layer. $SA(\cdot)$ is self attention operation. $CA(\cdot,\cdot)$ is the multi-modal version of scaled dot product attention where key and value pair comes from one modality and query from the other modality. $T_l \in \mathbb{R}^{n\times d_w}$ is the visually attended language feature at the l-th layer. Similarly, visually attended linguistic feature are also calculated:

$$\widehat{T}_l = CA\Big(T_l(\mathcal{M}), V_l(\mathcal{I})\Big), \text{where} \tag{3.14}$$

$$CA\Big(T_l(\mathcal{M}), V_l(\mathcal{I})\Big) = Softmax\left(\frac{T_l(\mathcal{I})V_l(\mathcal{M})^T}{\sqrt{d_w}}\right)V_l(\mathcal{M}) \tag{3.15}$$

$$T_{l+1} = Linear\Big(SA\big(\widehat{T}_l\big)\Big), \text{where} \tag{3.16}$$

$$SA\big(\widehat{T}_l\big) = Softmax\left(\frac{\widehat{T}_l\widehat{T}_l^T}{\sqrt{d_w}}\right)\widehat{T}_l \tag{3.17}$$

Finally, the joint representation of query image and modification text is determined as:

$$f(\mathcal{I},\mathcal{M}) = Pool(V_L) \tag{3.18}$$

where $f(\mathcal{I},\mathcal{M}) \in \mathbb{R}^{d_v}$. Note that here we use the pooled language–attended visual features as the joint representation. This allows the auxiliary module to use the visual features of the target image and query and predict the most important bits of visual information that need to be preserved across source and target images (see next Section 3.2.5).

3.2.4 Similarity Learning

Given a query image $\mathcal{I}$, a modification text $\mathcal{M}$, and a set of k candidate target images $C = \{\mathcal{I}_t\} \bigcup \{\mathcal{I}_{c_i}\}_{i=1}^{k-1}$ where $\mathcal{I}_t$ is the ground truth target image for the $(\mathcal{I}, \mathcal{M})$ pair. The main learning objective is to learn the model parameters so that the joint representation $f(\mathcal{I}, \mathcal{M})$ of the query image $\mathcal{I}$ and the modification text $\mathcal{M}$ is close to the representation $g(\mathcal{I}_t)$ of the target image $\mathcal{I}_t$, while being far apart from the feature representation of other candidate images. We can formulate the objective as follows:

$$sim(f(\mathcal{I}, \mathcal{M}), g(\mathcal{I}_t)) \gg sim(f(\mathcal{I}, \mathcal{M}), g(\mathcal{I}_{c_i})) \tag{3.19}$$

where $i = 1, 2, \cdots, k-1$. Here $sim(\cdot, \cdot)$ can be any similarity function. Similar to [117], we use the dot product as the similarity function $sim(\cdot, \cdot)$.

Following [117], we consider two different loss functions for the learning, namely the soft triplet loss and the batch classification loss. The soft triplet loss is defined as follows:

$$\mathcal{L}_{ST} = \sum_{i=1}^{k-1} \log \left(1 + \frac{\exp(sim(f(\mathcal{I}, \mathcal{M}), g(\mathcal{I}_t)))}{\exp(sim(f(\mathcal{I}, \mathcal{M}), g(\mathcal{I}_{c_i})))}\right) \tag{3.20}$$

$\mathcal{L}_{ST}$ is then summed across the query images in the batch.

The batch classification loss views the metric learning as a batch-based classification problem in which the modified image representation should be closest to the respective ground truth target image comparing to *all* the other target candidates in the batch:

$$\mathcal{L}_{BC} = \frac{1}{|B|} \sum_{i=1}^{|B|} -\log \Big(\frac{\exp\big(sim(f(\mathcal{I}_i, \mathcal{M}_i), g(\mathcal{I}_{t_i}))\big)}{\sum_{j=1}^{k-1} \exp\big(sim(f(\mathcal{I}_i, \mathcal{M}_i), g(\mathcal{I}_{c_j}))\big)}\Big) \tag{3.21}$$

where B is the batch and i-th sample in B is composed of triplet $(\mathcal{I}_i, \mathcal{M}_i, \mathcal{I}_{t_i})$.

3.2.5 Auxiliary Module

In this section, we propose another auxiliary module to further improve the efficiency and effectiveness of the learning.

Given an image $\mathcal{I}$, after extracting region features using a region proposal network (Sec. 3.2.1), the image can be represented as a 2-d representation in $\mathbb{R}^{K\times d_e}$. We then calculate a compact vector representation of the image via average pooling over K entities (regions), yielding $h(\mathcal{I}) \in \mathbb{R}^{d_e}$ where $h(\mathcal{I})$ is the vector representation of $\mathcal{I}$. We then apply an element-wise soft-sign function on $h(\mathcal{I})$ as:

$$\hat{h}(\mathcal{I}) = Sg\big(h(\mathcal{I}) + 1\big)/2 \tag{3.22}$$

where $Sg(\cdot)$ is the element-wise soft-sign function.

The auxiliary module has only 3 linear layers on top of the main network (see Fig. 3.2). The input to this module is $f(\mathcal{I}, \mathcal{M}) \in \mathbb{R}^{d_v}$. The output of this module is a vector $b(\mathcal{I}, \mathcal{M}) \in \mathbb{R}^{d_e}$, where each element of $b(\mathcal{I}, \mathcal{M})$ is a value between 0 and 1. We can interpret each element of $b(\mathcal{I}, \mathcal{M})$ as an important score used to reweight the corresponding element in $h(\cdot)$.

We then define the following auxiliary loss of this module as:

$$\begin{aligned}\mathcal{L}_B =& L_2(\hat{h}(\mathcal{I}) * b(\mathcal{I}, \mathcal{M}), \hat{h}(\mathcal{I}_t) * b(\mathcal{I}, \mathcal{M})) - \\ & L_2(\hat{h}(\mathcal{I}) * b(\mathcal{I}, \mathcal{M}), \hat{h}(\mathcal{I}_{c_i}) * b(\mathcal{I}, \mathcal{M})) + 1\end{aligned} \tag{3.23}$$

where $\mathcal{I}$ is a query image, $\mathcal{I}_t$ is the ground truth target image, $\mathcal{I}_{c_i} \neq \mathcal{I}_t$ is a random candidate target image in the batch, and $L_2(\cdot)$ denotes the L_2 norm of a vector. Intuitively, this loss function uses the L_2 distance of the feature vector $\hat{h}(\cdot)$ weighed by $b(\mathcal{I}, \mathcal{M})$ to measure the distance between the source and the target images.

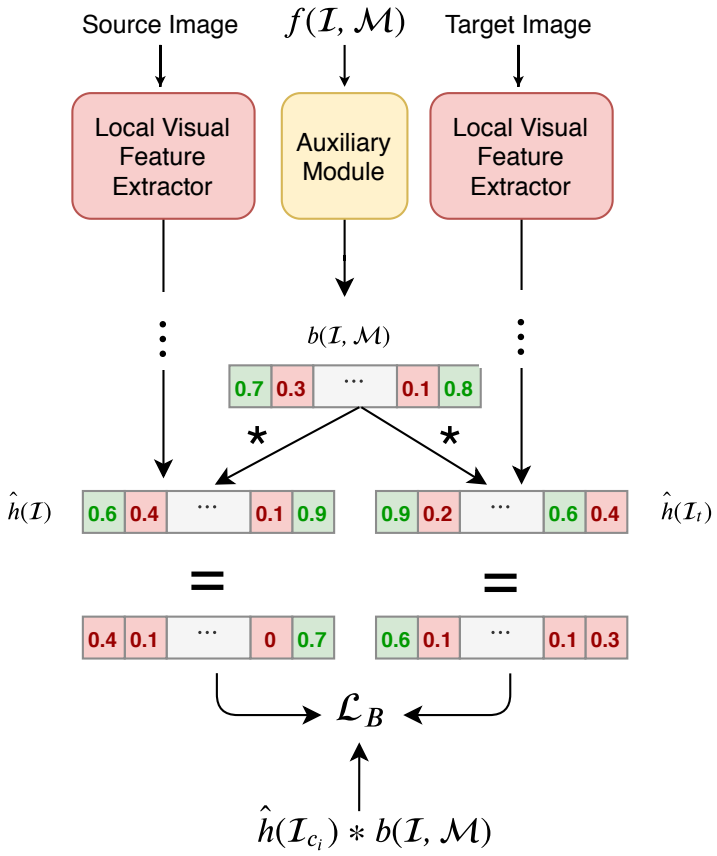


Figure 3.3: During training auxiliary module predicts a weight vector $b(\mathcal{I}, \mathcal{M})$ representing the importance of each feature in the vector representation of query and target images, given the requested modifications according to Eq. 3.23. During inference, this module can be activated to reject the distant candidates for the given query at an early stage (see Sec. 3.2.5).

During inference we first compute $b(\mathcal{I}, \mathcal{M})$. We then calculate $\hat{h}(\mathcal{I})$ and $\hat{h}(\mathcal{I}_{c_i})$ for all the images in set of candidate target images. Note that since $\hat{h}(\mathcal{I}_{c_i})$ requires only a simple average pooling, the computation of $\hat{h}(\mathcal{I}_{c_i})$ is much more efficient that $g(\mathcal{I}_{c_i})$ which requires multi-layer self attentions. We can then use $\hat{h}(\cdot)$ to do a coarse

retrieval and filter out those candidates whose distance with $\hat{h}(\mathcal{I})$ is greater than a defined threshold θ. Those candidates whose distance with $\hat{h}(\mathcal{I})$ is less than θ will be further process by the main network for fine retrieval. The final loss for training our proposed method is then $\mathcal{L}_{TS}$ (or $\mathcal{L}_{BC}$) $+\mathcal{L}_B$. Fig. 3.3 exhibits how our auxiliary module works in training.

3.3 Experimental Setup and Results

We compare the performance of the proposed method with state-of-the-art approaches on three benchmark datasets: Fashion200K [40], MIT States [48], and CSS [117]. Following [117], we use the *Recall@K* metric for comparison. This metric calculates the percentage of test queries for which the ground-truth target image is among the top K retrieved images. We report the performance for $K = \{1, 5, 10, 50\}$. Following prior work in [117], we use the soft triplet loss ($\mathcal{L}_{ST}$) on the MIT States dataset and the CSS dataset, and use the batch classification ($\mathcal{L}_{BC}$) loss on the Fashion200K dataset. We repeat the experiment 5 times on each dataset and report the mean/variance on each dataset.

We use PyTorch to implement our approach. We compare our method with TIRG [117], FiLM [87], Relationship [93], Parameter Hashing [75], Show and Tell [115], Attribute as Operator [72] and the method of [40]. We use the pretrained model provided by [5] with a Faster-RCNN [90] backbone for extracting the visual feature from each region proposal. This pre-model is trained on the MSCOCO dataset [35; 63] consisting of 123K images. For each region proposal, we have a 2048-d feature vector along with a 4-d spatial position encoding vector.

There are 2 layers in each of *VEP*, *TEP*, and cross-modal modules. The auxiliary module has 3 linear layers, where each linear is followed by ReLU. The first layer changes the channel dimension of its input from 768 to 1024, the second layer from 1024 to 2048, and the last layer keep the channel dimension at 2048. The main network is trained using Adam optimizer with linear-decayed learning rate [106] (LR = $1e{-}5$) and Adam optimizer [56] is used to optimize the weights of the auxiliary module (LR = $1e-1$). We run the experiments in two settings. The first setting (denoted as "*big*") extracts 36 region proposals for each image, while the second setting (denoted as "*small*") extracts only 18 region proposals. Note that all the reported numbers use the auxiliary module during training (*i.e.* disabling it in inference). The results of using the auxliary module as a coarse retrieval network are presented as ablation studies in Sec. 3.4.

3.3.1 Results on Fashion200K

The Fashion200K dataset [40] includes about 200K image of clothing images. Each sample is an image of a piece of a dress with accompanying attributes as the description (*e.g.* black leather jacket). This is a very challenging dataset since the visual difference between samples is often subtle. To generate training triplets, we follow [117; 40] and consider two images as the source and the target if they differ in their product description in one word. The modification text is then the different attribute between the source and the target, and is generated on the fly (*e.g.* "change blouse to dress"). Using this setting, there are 172K training triplets and 31K testing triplets.

Method	*Recall@*		
	K=1	*K=10*	*K=50*
Baselines			
Image only [117]	3.5	22.7	43.7
Text only [117]	1.0	12.3	21.8
Concat [117]	$11.9^{\pm 1.0}$	$39.7^{\pm 1.0}$	$62.6^{\pm 0.7}$
SOTA			
Han et al. [40]	6.3	19.9	38.3
Show and Tell [115]	$12.3^{\pm 1.1}$	$40.2^{\pm 1.7}$	$61.8^{\pm 0.9}$
Param. Hash. [75]	$12.2^{\pm 1.1}$	$40.0^{\pm 1.1}$	$61.7^{\pm 0.8}$
Relationship [93]	$13.0^{\pm 0.6}$	$40.5^{\pm 0.7}$	$62.4^{\pm 0.6}$
FiLM [87]	$12.9^{\pm 0.7}$	$39.5^{\pm 2.1}$	$61.9^{\pm 1.9}$
TIRG [117]	$14.1^{\pm 0.6}$	$42.5^{\pm 0.7}$	$63.8^{\pm 0.8}$
Ours (big)	$\mathbf{17.78^{\pm 0.5}}$	$\mathbf{48.35^{\pm 0.6}}$	$\mathbf{68.5^{\pm 0.5}}$
Ours (small)	$\mathbf{16.26^{\pm 0.6}}$	$\mathbf{46.90^{\pm 0.3}}$	$\mathbf{71.73^{\pm 0.6}}$

Table 3.1: Results on the Fashion200K dataset. The numbers of other approaches are adopted from [117]. The proposed method outperforms other state-of-the-art approaches in terms of *Recall@K* metrics. In particular, our proposed method gains a 26% performance boost over the previously best result in terms of *Recall*@1.

Results on this dataset are shown in Table 3.1. Our proposed method outperforms other approaches in all the metrics with a remarkable 26% performance improvement over TIRG [117] in terms of the *Recall@1* metric. We believe that this improvement is due to the fact that our proposed method operates on regions instead of the whole

Method	*Recall@*		
	K=1	*K=5*	*K=10*
Baselines			
Image only [117]	$3.3^{\pm 0.1}$	$12.8^{\pm 0.2}$	$20.9^{\pm 0.1}$
Text only [117]	$7.4^{\pm 0.4}$	$21.5^{\pm 0.9}$	$32.7^{\pm 0.8}$
Concat [117]	$11.8^{\pm 0.2}$	$30.8^{\pm 0.2}$	$42.1^{\pm 0.3}$
SOTA			
Show and Tell [115]	$11.9^{\pm 0.1}$	$31.0^{\pm 0.5}$	$42.0^{\pm 0.8}$
Attribute Op. [72]	$8.8^{\pm 0.1}$	$27.3^{\pm 0.3}$	$39.1^{\pm 0.3}$
Relationship [93]	$12.3^{\pm 0.5}$	$31.9^{\pm 0.7}$	$42.9^{\pm 0.9}$
FiLM [87]	$10.1^{\pm 0.3}$	$27.7^{\pm 0.7}$	$42.9^{\pm 0.9}$
TIRG [117]	$12.2^{\pm 0.4}$	$31.9^{\pm 0.3}$	$41.3^{\pm 0.3}$
Ours (big)	$\mathbf{14.72^{\pm 0.6}}$	$\mathbf{35.30^{\pm 0.7}}$	$\mathbf{46.56^{\pm 0.5}}$
Ours (small)	$\mathbf{14.29^{\pm 0.6}}$	$\mathbf{34.67^{\pm 0.7}}$	$\mathbf{46.06^{\pm 0.6}}$

Table 3.2: Results on the MIT States dataset. The numbers of other approaches are adopted from [117]. Our proposed method outperforms other state-of-the-art approaches in terms of *Recall*@K metrics. In particular, our proposed method gain a 19.67% performance boost in terms of *Recall*@1.

image. This allows our method to more effectively capture the relationship between the modification text and each entity in the image. Moreover, we obtain noticeably better results when $K = 36$ ("*big*").

3.3.2 Results on MIT States dataset

The MITStates dataset [48] contains about 60K images. Each image is annotated with a noun and an adjective. In total, the images are annotated using 245 unique nouns and 115 unique adjectives. Following the standard train and test splits provided by [117], there are about 43K training samples and 80 nouns are used for training. The rest is kept for testing.

Table 3.2 shows the result for the proposed method and other state-of-the-art approaches on this dataset. Our method outperforms others by a large margin. For example, our method achieves **14.72** in *Recall@1* which corresponds to ∼20% performance boost over Relationship [93] and TIRG [117] methods. Again, we observe that the results are better when using 36 region proposals ("*big*").

3.3.3 Results on CSS Dataset

The CSS dataset [117] is a synthetic dataset of images containing several different geometric objects (sphere, cube, etc.) sitting in a variety of layouts. CSS has been produced on top of the CLEVR platform [52]. It contains about 19K training images and 18K testing images, respectively. Modification text for this dataset falls into three categories: adding new objects to the scene, removing objects from the image, and changing the attributes of the current objects in the image. This dataset is especially very interesting. Unlike other datasets that have relatively simple modification text, the CSS dataset contains more complicated modification text with positional words. For instance, a modification text can be "*add a cube to the **right** of the sphere.*". We use the 3D and 2D versions of the dataset in our experiments and report the

Method	*Recall@*	
	$3D \rightarrow 3D$	$2D \rightarrow 3D$
	K=1	*K=1*
Baselines		
Image only [117]	6.3	6.3
Text only [117]	0.1	0.1
Concat [117]	$60.6^{\pm 0.8}$	27.3
SOTA		
Show & Tell [115]	$33.0^{\pm 3.2}$	6.0
Param Hash. [75]	$60.5^{\pm 1.9}$	31.4
Relation. [93]	$62.1^{\pm 1.2}$	30.6
FiLM [87]	$65.6^{\pm 0.5}$	43.7
TIRG [117]	$73.7^{\pm 1.0}$	46.6
Ours (big)	$\mathbf{79.2^{\pm 1.2}}$	$\mathbf{55.69^{\pm 0.9}}$
Ours (small)	$\mathbf{67.26^{\pm 1.1}}$	$\mathbf{50.31^{\pm 0.9}}$

Table 3.3: Results on the CSS dataset. The numbers are adopted from [117]. $3D \rightarrow 3D$ is when the query and target images are both in 3D. $2D \rightarrow 3D$ denotes the setting when the query image is a 2D while the target image set is 3D. Our proposed method outperforms other emphasizing its generalization strength to other domains.

measured *Recall@{1,5}*. The 2D version is more challenging since it corresponds to the situation where the source and target distributions are different. Table 3.3 shows the result of experiment on this dataset. Consistent with other experiments, our

Variation	*Recall@*		*ACRR*
	K=1	*K=5*	
Effect of $\mathcal{L}_B$			
Ours (w/ $\mathcal{L}_B$)	79.2	94.08	
Ours (w/o $\mathcal{L}_B$)	76.1	91.24	
Effect of distance threshold (θ)			
Ours ($\theta = 100$)	**75.92**	**91.03**	**75.24%**
Ours ($\theta = 85$)	70.81	84.07	91.86%

Table 3.4: Ablation studies results on CSS (3D) dataset. First two rows exhibit the the additional loss function role in overall performance: the performance boosts when the additional loss function is added to the main loss function. Last two rows shows average rejection rate and overall performance when AM is used as an early rejection network in inference mode.

proposed method is able to outperform other state-of-the-art on this dataset as well. Again, the results are better when we use the model with more region proposals (i.e. "*big*").

3.4 Ablation Studies and Discussions

In this section, we conduct ablation studies to investigate the effect of various components of the proposed method. We conduct our studies on the CSS (3D) dataset as it provides the most challenging modification text among all datasets.

Effect of $\mathcal{L}_B$: As discussed in Sec. 3.2.5, the auxiliary module is a lightweight

module that can be applied on top of any composed query image retrieval system as long as the system provides a vector representation of the composed query and target image(s). First, we analyze the role of this module on the overall performance. In the first two rows of Table 3.4, we show the results of with and without this module training. Here we do not use this module to filter out any candidate images (i.e. all test images go through the main network). The results show that using $\mathcal{L}_B$ provides additional supervision signal for the training and improves the overall performance.

Next, we analyze the effect of using $\mathcal{L}_B$ to filter out candidate images during testing. In other words, we reject those test candidates that are very dissimilar to the query image, *before* processing them using VEP. To quantify this effect, we define a measure called "Average Candidate Rejection Rate" (ACRR):

$$ACRR = 1 - \frac{\sum_{i=1}^{|TQ|} \mathbb{1}\Big(Dist\big(\mathcal{X}(\mathcal{I}), \mathcal{X}(\mathcal{I}_{c_i})\big) < \theta\Big)}{|TQ| \times |CI|} \tag{3.24}$$

$$s.t. \quad \mathcal{X}(\mathcal{I}) = \hat{h}(\mathcal{I}) * b(\mathcal{I}, \mathcal{M}) \tag{3.25}$$

$$\mathcal{X}(\mathcal{I}_{c_i}) = \hat{h}(\mathcal{I}_{c_i}) * b(\mathcal{I}, \mathcal{M}) \tag{3.26}$$

where TQ and CI are the set of all testing queries and candidate images, respectively. $Dist(\cdot, \cdot)$ and $\mathbb{1}(\cdot)$ are the $L1$ and indicator functions, respectively. Intuitively, ACRR measures the percentage of test candidates that have been filtered on average (a higher ACRR corresponds to more candidate images being filtered out). The last two rows in Table 3.4 show the result with different values of the θ threshold. For example, when setting $\theta = 100$, we can filter out 75.24% of the candidate images during testing. This greatly improves the efficiency of the method without significantly sacrificing the overall accuracy.

Qualitative Examples: Fig. 6.3 shows some qualitative examples of our method.

Each row shows a reference image, the desired changes in terms of a modification text, and the retrieved images from the test set.

Figure 3.4: Some qualitative examples of our method. Each row shows the query image, the modification text, and retrieved images. The examples are from Fashion200K, MITStates, and CSS3D datasets, respectively.

3.5 Conclusion

We have proposed a novel approach for the problem of composed query image retrieval. Our proposed method represents the input image as a set of local regions (entities). We then learn a bidirectional correlation between the words in the modification text and local areas in the image. Besides, we propose an auxiliary module that can be used to effectively filter out candidate images during testing. This can improve the efficiency of the method without sacrificing too much on the accuracy. Through extensive experiments, we demonstrate that our proposed method outperforms other state-of-the-art methods by a large margin.

Chapter 4

Video Captioning of Future Frames

4.1 Introduction

In this chapter, we consider the problem of generating captions for future frames in a video. This problem is related to video captioning. But there are important differences as well. In standard video captioning, all frames in the entire video are observed. But in our problem setting, we consider an online setting where we only have access to the frames observed so far. Our goal is to generate captions for future frames that are not observed yet.

Humans have the amazing ability to anticipate the future based on current events (see Fig. 4.1). Given a short clip of an event happening now, humans can easily anticipate and describe what will most likely happen next. For example, after observing that *"a coach is advising a weight lifting athlete"*, it is easy for us to anticipate that later on *"the athlete will lift the weight again after the advice"* (Fig. 4.1). The goal of this chapter is to enable intelligent agents to have similar capabilities. Generating cap-

tions of future frames also has many important real-world applications. For example, consider the application of assisting visually impaired people. If we can have a system that can automatically generate captions of future events based on the observed visual scenes, the person will be able to anticipate possibly dangerous events in the future and take action appropriately.

Generating captions for future frames is a challenging problem. Since the algorithm does not have access to future frames, it needs to understand the semantic information of the current scene and accurately predict the future. Inspired by recent work on future semantic segmentation [70], we develop our approach by predicting the feature maps of future frames based on observed frames. Then we can use a standard captioning module to generate the captions based on the predicted feature maps of the future frames. we demonstrate superior performance over the baselines on the challenging ActivityNet-Captions dataset [57].

The contributions of this work are manifold. First, we take tackle the problem of video captioning for upcoming future frames in a video in two different settings: general, and conditional captioning. Compared with standard video captioning, this new problem setting is closer to many real-world applications such as assistive technologies for the visually impaired. Second, we propose a novel approach to this problem. Our approach is based on predicting the feature maps of future frames, rather than directly generating pixel values. Finally, our experimental results demonstrate the effectiveness of the proposed approach compared with other baselines.

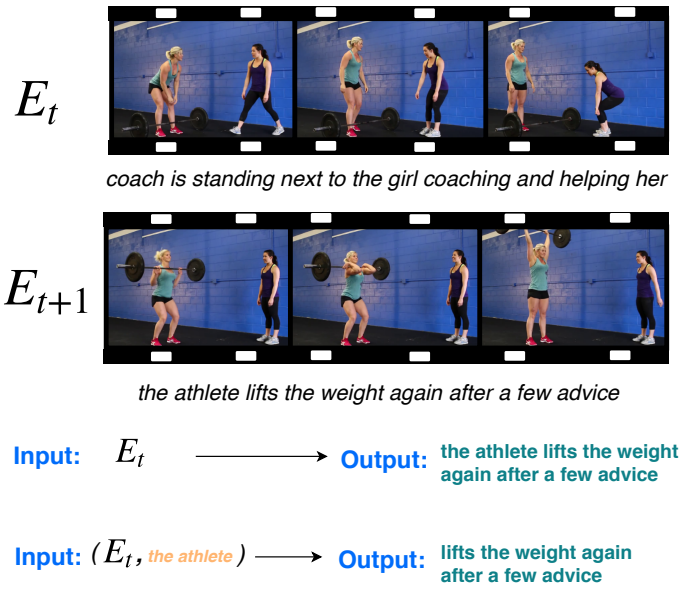


Figure 4.1: Given a sequence of frames of what is happening (E_t), the task is to anticipate what will happen in the next event (future) and describe it using a sentence. First row: the general version of the problem in which the task is to generate all the words in the caption for the next event. Second row: conditional future captioning where the task is to take the current event as well as a noun phrase defining the actor of the next event (*e.g.* the athlete in this case) and describe what the actor will do in the next event.

4.2 Problem Statement

Anticipation is the ability to inference across space, time, causality, etc. [91]. It is considered to be a fundamental capability for an intelligent entity [41]. It allows humans to partially observe a scene and describe what may happen afterward. For instance, given a short clip of "*she opened the hood of the car*", we can describe the

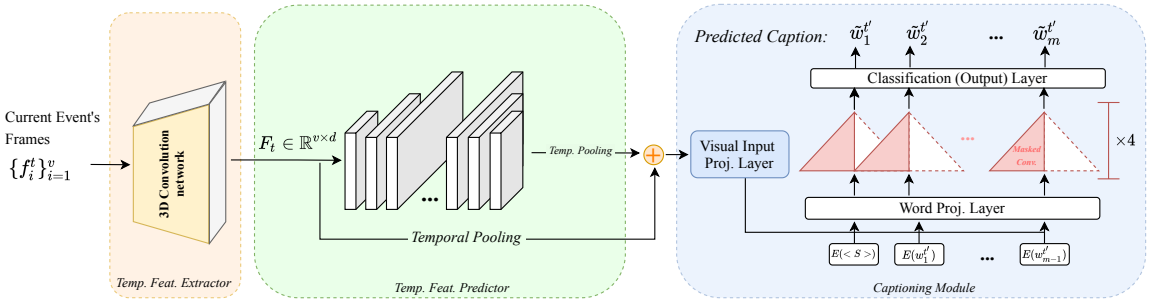


Figure 4.2: After training TFP, it is plugged into the main pipeline for the next stage of training. To train the main pipeline, a pre-trained C3D network computes the convolution features for each event (yellow box). Extracted features are then fed into the TFP to obtain the predicted features for the next event (green box). After temporal pooling, the predicted features are combined with the contextual features coming from the current event ($\oplus$) and are input to the captioning module using a projection layer. The captioning module adds the visual features to the word embedding features for each input word, applies n layers of mask convolution to effectively increase the receptive field for each word. The next word is generated by using softmax on top of the classification layer in the captioning module (blue box).

next possible scene (event) which could be "*she then examined the engine*" [130].

We define an event in a video as a number of consecutive frames in a video clip that capture an action being performed. This notion of an "*event*" is consistent with the definition in [57] as well. Our goal is to generate captions for future events that are not observed yet. We consider two problem settings for future frame captioning. In the first problem setting (which we call the *general case*), we are given a sequence of frames $\{f_i^t\}_{i=1}^v$ representing an event in the video at time-stamp t, where f_i^t denotes

the i-th frame within the t-th event in a video. Note that the number of frames v of an event can be variable for different events. The goal is to generate a sentence $\{w_i^{t'}\}_{i=1}^{m}$ describing what is happening in the next important event in the video at time-stamp t' (where $t < t'$). Here m represents the number of words in the sentence and $w_i^{t'}$ is a word in the sentence. In this setup, the model should solely rely on what is happening at the current moment in the video. It needs to infer what is the *subject* of the upcoming event and describe what the subject would do next.

However, due to the uncertainty of the future, the current event could logically be followed by several different events with different subjects. For example, consider the case where the current event is "*the person blows the leaves from a grass area using the blower*". Potentially, both "***the blower*** *is seen up close.*" and "***the person*** *then walks away from the camera.*" make sense even to humans to be the next possible event in the video.

To take into consideration this uncertainty, we take inspiration from some work in the NLP community [130] and propose the second problem setting. In this problem setting (which we call the *conditional case*), we have access to the visual information for the current event in the video as well as the actor (subject) for the next event. Our goal is to generate the caption for the next time-stamp, given the current visual information and the noun phrase representing the actor of interest for the next event.

4.3 Proposed Approach

One possible way of future frame prediction is to first predict future frames themselves, then apply a captioning model on the predicted frames. However, directly

predicting the pixel values of future frames is challenging. Some recent work in semantic segmentation [69; 70] suggests that predicting convolutional features of future frames is a better choice than predicting raw pixel values. Following these previous works, we also focus on forecasting the convolutional features for the next event and generate captions based on the predicted features. Our proposed method consists of three major modules: a 3D convolution network as a backbone, a feature predictor, and a captioning module (Fig. 6.3).

Temporal Feature Extractor. 3D convolutional networks [57; 29] have been popular in video understanding tasks. In this work, we use the 3D convolutional model proposed by [51] as our feature extractor backbone. Given a sequence of raw frames for the current event $\{f_i^t\}_{i=1}^v$, this network processes the sequence and outputs a $F_t \in \mathbb{R}^{v \times d}$ feature map where v is the number of frames and d is the feature dimension. These features are used as input to the following modules.

Temporal Feature Predictor. Given the feature map F_t from the t-th event, the goal of this module is to predict the feature map $F_{t'}$ of the next event t' $(t < t')$. We first describe this module during the training stage. Each training instance consists of a pair of feature maps $(F_t, F_{t'})$ extracted by the temporal feature extractor for two adjacent events (t, t') from a training video. Note that the temporal dimensions of F_t and $F_{t'}$ can be different, i.e. $F_t \in \mathbb{R}^{v \times d}, F_{t'} \in \mathbb{R}^{v' \times d}$ where $v \neq v'$. The goal of the temporal feature predictor (TFP) module is to predict $F_{t'}$ given F_t.

This module uses a temporal convolution on F_t to produce a new feature map. During training, we learn the parameters of this temporal convolution layer, so that

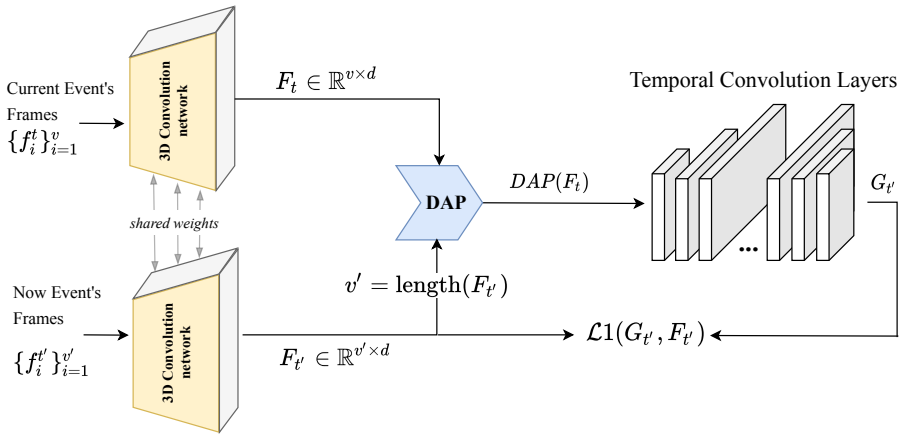


Figure 4.3: Training of TFP module on a pair of consecutive events at t and t' ($t < t'$). For each event, 3D convolution features are computed using a pre-trained C3D network F_t and $F_{t'}$. A dynamic adaptive pooling layer (DAP) takes the first event's features, F_t and the temporal length of the next event, v', and resizes the first event temporally to match the temporal dimension of the next event. The features are fed into a network consisting of several temporal convolution layers (grey blocks) to obtain the predicted features for the next event. The temporal feature predictor network maintains the temporal dimension throughout the layers. Grey dashed arrows indicate weight sharing.

the predicted new feature map matches $F_{t'}$. In order to handle the case where F_t and $F_{t'}$ have different temporal dimensions (i.e. $v \neq v'$), we propose a dynamic adaptive pooling (DAP) layer at the beginning of the TFP pipeline (see Fig. 4.3) which shrinks or enlarges the temporal dimension, depending on the length of the next event. In other words, this module can be written as:

$$G_{t'} = Conv(DAP(F_t)), \text{where } G \in \mathbb{R}^{v' \times d} \tag{4.1}$$

Here $DAP(\cdot)$ is the dynamic adaptive pooling operation that maps the temporal dimension of the input to t', while $Conv(\cdot)$ is the temporal convolution operation. Note that DAP only operates during the training process. During testing, v' is unknown and thus we remove DAP from the pipeline. As a result, the generated features during testing have a temporal length of v. Note that DAP does not have any weights to learn, omitting it from the testing pipeline does not harm the performance of the trained module.

Captioning Module. While it is possible to directly feed the captioning module with the predicted feature vector $G_{t'}$ of the next event, this vector does not carry any information about the context in which the next event should be described by the captioning module. Having the context added to the visual information vector enables this module to describe the next event more accurately. The feature map of the current event can be considered a source of context, so we propose to combine it with the predicted feature map of the next event as follows:

$$G_{t'}^{final} = \lambda \cdot \mathcal{AVG}(F_t) \oplus (1 - \lambda) \cdot \mathcal{AVG}(G_{t'}) \tag{4.2}$$

where $\oplus$ is the element-wise summation, $\mathcal{AVG}(\cdot)$ is the average pooling along the temporal dimension, and $0 \leq \lambda \leq 1$ is a hyper-parameter of the model which controlls the amount of context to be fused into the next event's features. $G_{t'}^{final}$ is considered as the visual information going through the captioning module.

Although LSTM-based models have been widely used in tasks relating to joint vision and natural language processing [126; 88; 136; 13; 121], they fall short in two aspects. First, it is a common issue among the LSTM-based models that as the length

of the sentence becomes longer, the performance of these models drops significantly [110; 6]. Second, LSTMs (and other recurrent units) are not easily parallelizable since the input for each time step needs to be calculated before the unit can move on to the next one. Lately, a number of approaches have been proposed to address these shortcomings by either using only self-attention layers [110; 138] or convolution layers [6] for sequence to sequence tasks involving natural language processing. Using either of these approaches can make the training easily parallelizable.

For the captioning module, we take advantage of the convolutional captioning module proposed by [6] because of its great performance on the image captioning task. Note that the proposed method is not limited by any specific captioning architecture. This module used in this work can be easily substituted with any existing alternative. We adopt a similar architecture that has been used in [6] with some modifications. In [6], the visual input to the captioning module is a vector of 4096 features coming from the FC7 layer of a VGG-16 network [103]. But in our case, we take the feature G from the temporal feature predictor and perform an average pooling over the dimension to obtain an d-dimensional feature vector. This feature vector is then used as the input to the captioning module.

Training and Inference. To train our model, we first extract convolution features for each event in a training video offline using a pre-trained 3D model. This results in a $v \times d$ feature representation for each event where we use $v = T/16$, $d = 500$ in the experiments. Here T is the number of frames in the video clip corresponding to the event.

Then the feature extractor model is trained using the offline-computed features from the 3D network. As mentioned earlier, this network receives a $v \times d$ representation of the current event. It temporally resizes it to $v' \times d$ on the fly and generates the predicted features of the next event, $G_{t'} \in v' \times d$. To train this network we use a $\mathcal{L}1$ loss function between the predicted features of the next event and the pre-extracted ones:

$$\mathcal{L}_{TFP} = \sum_{i=1}^{v'} \sum_{j=1}^{d} |G_{t'}(i,j) - F_{t'}(i,j)| \tag{4.3}$$

Once this module is trained, it is plugged into the proposed network and the entire network is trained solely using a standard cross-entropy captioning loss. Note that when adding the temporal feature predictor module into the pipeline, the dynamic adaptive pooling layer (DAP) is detached and is no longer used.

The captioning module is first trained on the MSCOCO [63] dataset which has a total number of 9.2K words in its vocabulary. The weights of the pre-trained model are then used to initialize the weights of the captioning module in our case. Since the vocabulary size is 6K in our problem, we have to replace the first and last layers of the captioning module, *i.e.* the word embedding and word outputting layers in Fig. 6.3. Therefore for the new outputting layer (of size 6K), weights are initialized with a normal distribution where the mean and standard deviation are calculated based on the learned weights on MSCOCO. For the embedding layer, on the other hand, the weights are initialized randomly sampled from a normal distribution with *mean=0* and *std=0.1*.

Putting everything together, now the entire network is trained using a cross-entropy loss defined on the probability of predicted words in the sentence. At each

position i in the sentence and for each word w, the probability is defined as $p(\tilde{w}_i|w < i, I)$ where $y < i$ are the ground truth (GT) words in positions before i and I is the projected visual features.

Since during training we have access to the GT word at each position, it is feasible to train the network in parallel. Nonetheless, inference happens sequentially since the prediction of each word depends on the previous predicted words, $p(\tilde{w}_i|\tilde{w} < i, I)$. Sentence generation starts with feeding a special token $< S >$ indicating the beginning of the sentence and is continued until the end-of-sentence token $< EOS >$ is generated (or reaching the maximum number of generated words).

4.3.1 Implementation Details

The feature extractor module consists of 7 temporal dilated convolution layers (except the first one), where each convolution layer is followed by a ReLU layer. The first 3 layers convert the feature depth to 1024 while preserving the temporal length. The next 4 layers return the feature depth to 500 again while retaining the temporal dimension of input (Fig. 4.3). Feature extractor is trained with an SGD optimizer with an initial learning rate of 0.001. The captioning module uses 3 layers of masked convolution with a kernel size of 5. Word projection and image projection layers map their input into a 300-d and 512-d space, respectively. RSMProp optimizer has been used with an initial learning rate of 5×10^{-5}, while the feature predictor module learns with $LR = 10^{-6}$ after plugging into the main pipeline. We experiment with the proposed approach with different λ and the results are reported in Table 4.3.

Method	B2	B3	B4	CIDEr	METEOR	ROUGE-L
Baseline	7.87	3.08	1.32	12.77	7.24	17.76
Proposed method ($\lambda = 0$)	8.25	3.33	1.52	13.49	7.80	18.46
Ours ($\lambda = 0.50$)	**8.55**	**3.56**	**1.60**	**15.28**	**7.82**	**18.62**
oracle	*8.70*	*3.80*	*1.90*	*17.05*	*8.05*	*18.87*

Table 4.1: Performance of the proposed method compared to the baseline and oracle method on the first problem setting (i.e. general case). The hyperparameter λ controls the amount of visual contextual information added to the predicted features for the next event. Adding context substantially boosts the performance of the proposed method, especially CIDEr metric.

4.4 Experiments and Results

In this section, we first introduce the datasets in Sec. 4.4.1 and the evaluation metrics in Sec. 4.4.2. We then introduce methods used for comparison in Sec. 4.4.3. We present the experimental results in Sec. 4.4.4. Finally, We then more precisely examine our approach through ablation study. Finally, we perform ablation studies in Sec. 4.4.5.

4.4.1 Datasets

We use the ActivityNet-Captions dataset [57] and the SWAG-AF dataset [130] in the experiments. The ActivityNet dataset [14] is a large-scale benchmark for video understanding. Krishna et al. [57] have expanded the dataset by providing temporal

Method	B2	B3	B4	CIDEr	METEOR	ROUGE-L
Baseline	9.04	4.23	1.95	20.81	**7.17**	**18.57**
Ours ($\lambda = 0.5$)	**9.09**	**5.01**	**2.76**	**26.55**	7.02	18.16
oracle	*9.38*	*4.6*	*2.30*	*27.17*	*7.45*	*19.08*

Table 4.2: Performance of the proposed method on the conditional future captioning task compared to the baseline and oracle methods. The hyperparameter λ controls the amount of visual contextual information added to the predicted features for the next event.

Method	B2	B3	B4	CIDEr	METEOR	ROUGE-L
Ours ($\lambda = 0$)	8.25	3.33	1.52	13.49	7.80	18.46
Ours ($\lambda = 0.35$)	8.47	3.51	1.57	14.80	7.93	18.62
Ours ($\lambda = 0.50$)	**8.55**	**3.56**	**1.60**	**15.28**	7.82	18.62
Ours ($\lambda = 0.65$)	8.51	3.50	1.53	15.05	**7.96**	**18.83**

Table 4.3: Performance of the proposed method in the first problem setting (i.e. general case) using different values of λ. By increasing λ and therefore injecting more contextual information, we obtain better results. However when λ becomes bigger than 0.50, the performance starts to drop.

annotations and descriptions for the events in the video to form the ActivityNet-Captions dataset. There are about 20K Youtube videos in the dataset with an average of 3.65 events per video. Videos cover a broad spectrum of human activities. There

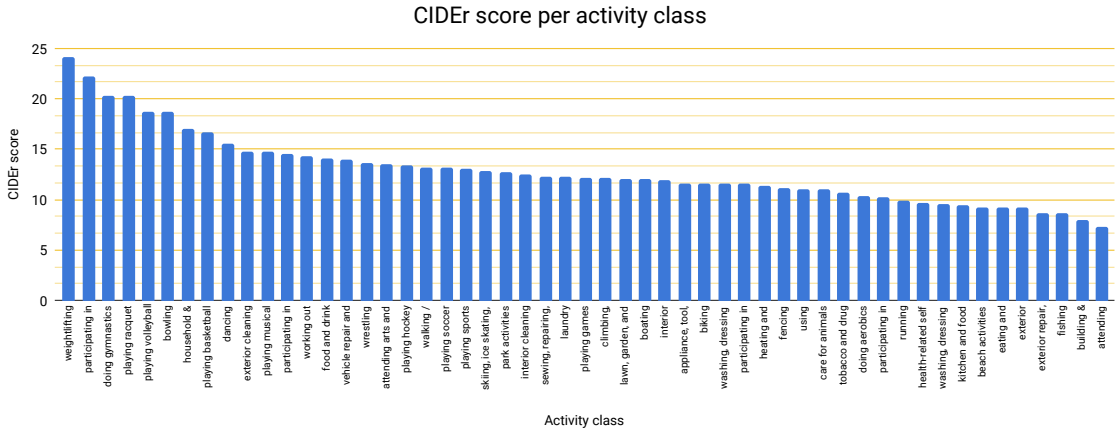


Figure 4.4: CIDEr score for different activity classes. While our proposed method works generally well, we found that it works best in the events relating to the sports and has moderate performance in more complex environments such as "*building & repairing furniture*", "*exterior repair, improvements, & decoration*", and "*exterior maintenance, repair, & decoration*".

are about 100,000 events. Each event is temporally annotated and described using a sentence. We use the standard train/val split in [57] for the ActivityNet-Captions dataset.

The SWAG-AF dataset [130] has been recently introduced for the NLI and entailment task. To test the performance on future conditional captioning tasks, we use SWAG-AF to extract the noun phrase representing the actor in the next event. The noun phrase is used as the initial words for the caption of the next event. Note that SWAG-AF contains the samples of the ActivityNet-Caption dataset. Thus, for the conditional case, we report the results on the intersection of ActivityNet-Captions and SWAG-AF. Moreover, since some of the events in the ActivityNet-Captions are

highly overlapped with each other, we use the SWAG-AF dataset to filter out those events –as such events are not included in SWAG-AF.

4.4.2 Evaluation Metrics

We evaluate the performance of our approach using several evaluation metrics, including BLEU@N, METEOR, ROUGE-L, and CIDEr. BLEU@N [81] is a family of methods that computes the precision of the generated caption using N-gram matching. METEOR [11] has more focus on the recall accuracy of the generated caption. ROUGE-L [62] measures the quality of the generated caption using the longest common subset between the predicted and the ground-truth sentence. CIDEr [111] is a newly-introduced metric and is reported to be very consistent with human judgments.

4.4.3 Compared Methods

We set up a baseline and an oracle method to compare the performance of the proposed method. For the baseline, we extracted the visual features for the current event using the feature extracted network (3D network). The features are then averaged along the temporal dimension and are fed into the captioning module. In other words, the caption decoder in this baseline has to caption the next event solely based on the features of the current event. To train the baseline, each event is accompanied by the consequent event's caption as the ground-truth.

We also set up an oracle method that has access to the next event and generates captions for future events based on the corresponding visual features. To do so, the oracle receives the extracted features for the next events, averages them across the

frames to obtain a $500 - d$ vector which is then fed into the captioning module as the visual input. This oracle represents an upper bound for this task. We want the performance of our method to be as close as possible to this oracle. For the sake of fair comparison, we use the same backbone feature extractor and captioning modules in all methods (baseline, oracle, and ours). However, the proposed method is by no means constrained to these backbones. It can be used in conjunction with any video feature extractor and/or captioning module available in the literature.

4.4.4 Results

We first analyze the performance of our method in the first problem setting where the model is asked to generate the caption without having access to information about the next event's actor. Table 4.1 reports the performance on this task. The second row ($\lambda = 0$) is the case where the method does not take advantage of the context of the video. As seen in Table 4.1, the proposed method outperforms the baseline. But there is still a noticeable gap with the oracle's performance. When adding the visual context information to the predicted feature vector of the next event (third row in Table 4.1), the performance of the proposed method increases substantially.

Furthermore, we examine the performance of the proposed method against the baseline and oracle in the conditional future captioning task. In this task, during inference for each event e, the models have access to a ground-truth noun phrase for the next event $N_e = \{w_i\}_{i=1}^{n_e}$ which indicates the actor in the next event. Although at first glance, this task seems to be easier than the general case, Table 4.2 shows that it is still a challenging task. For this task, the proposed method outperforms the

baseline by a large margin in terms of BLEU@3, BLEU@4, and more significantly in terms of CIDEr. Using more information about the future event, i.e. the next event's actor entity tends to boost almost every metric in Table 4.2 (compared with Table 4.1) except METEOR. For METEOR, it slightly decreases. We believe this is due to the nature of the METEOR score that sometimes fails to capture the similarity between two sentences as it is not originally introduced for captioning tasks [55]. The CIDEr score, on the other hand, is specifically designed to evaluate the captioning-related tasks and one can see a significant boost on the CIDEr score in Table 4.1 compared with Table 4.2. Fig. 6.3 provides some qualitative examples of our method on the ActivityNet-Caption validation set. The proposed method works well on the sport-related events and works moderately in complex scenes as it does in the third example.

4.4.5 Ablation Study

The hyperparameter λ in our method controls the relative contribution of the context information from the current event and the predicted features of the next event. We analyze the importance of λ and find the most suitable value for it through an ablation study. We split the original *training* set of ActivityNet-Captions dataset into two new disjoint sets of training and evaluation. The new training set has 80% of the original training samples and the new evaluation set inherits the remaining 20% samples. We then run our method on the new training set and measure the performance using the new evaluation set. Once we found the λ value that works best on the evaluation set (in this case $\lambda = 0.50$), the proposed method is trained

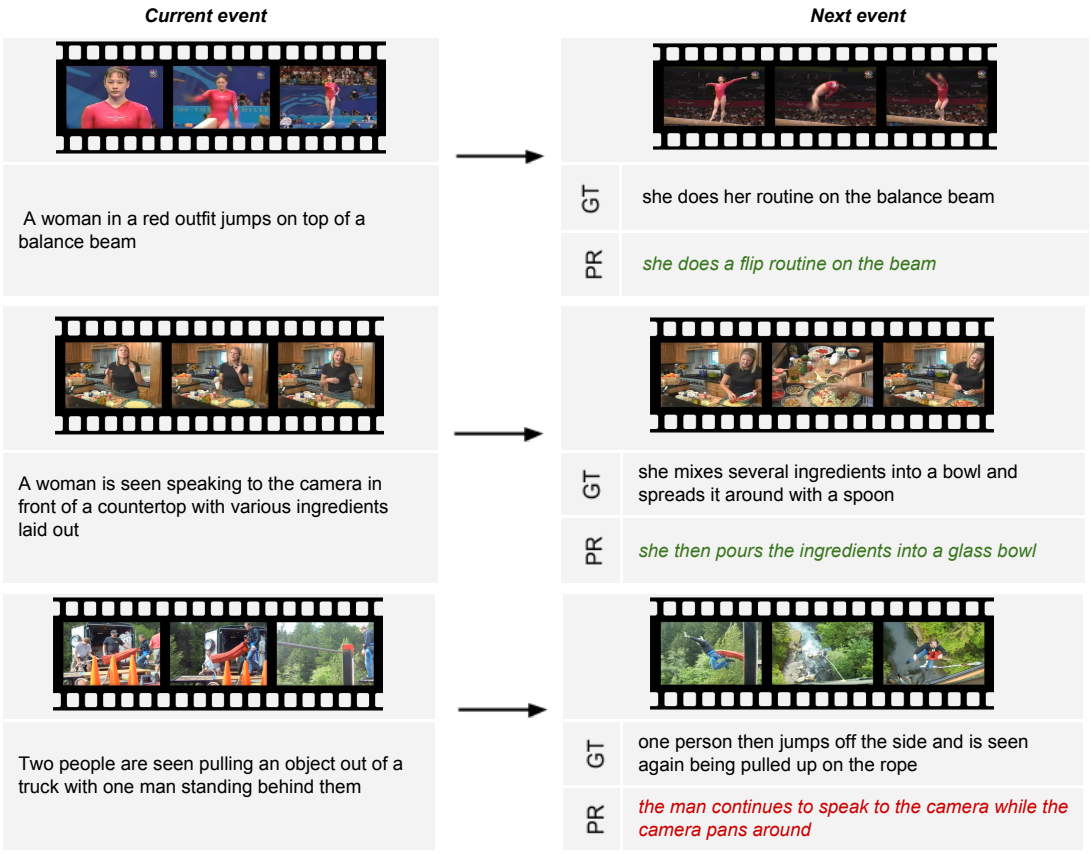


Figure 4.5: Qualitative examples. GT and PR are the ground-truth and predicted caption for the next event. In the first and second examples, the proposed method accurately captions the next event. But it fails to describe properly in the third example.

again on the entire *original* training set and tested on the original validation set to obtain the results in Table 4.1 and Table 4.3.

Setting λ to 0.50 means that half of the visual information is coming from the current event while the other half is from the predicted features using the TFP module. In other words, we equally rely on our TFP module and the information that we have at hand about the current event. Table 4.3 clearly shows that this strategy

yields the most appealing results. Interestingly adding more context information (e.g. $\lambda = 0.65$) is detrimental to the general performance of the method. Having $\lambda = 0.65$ causes the model to lose to the case where the λ is set to 0.50 in 5 out of 7 metrics, namely in BLEU@{2,3,4}, CIDEr, and METEOR metrics.

To further analyze the proposed method when captioning different activity types, we present the CIDEr score for each class of activity. In total there are 200 unique activity labels available on the ActivityNet V1.3 dataset. Based on their semantics, these activities are merged together to form 53 super activity groups. Following this taxonomy, we merge the event in the validation set according to their supergroup labels as well. We then use the best version of our method, *i.e.* $\lambda = 0.5$, for each of those groups. Finally, we compute the CIDEr score for each supergroup individually. Fig. 4.4 depicts the obtained result for this analysis.

We have found that our method is most effective for activities related to sports. The top 5 most accurately activity types are "*weightlifting*", "*participating in rodeo competitions*", "*doing gymnastics*", "*playing racquet*", and "*playing volleyball*". On the other hand, the least accurately activity types are "*building & repairing furniture*", "*fishing*", "*exterior repair, improvements, & decoration*", and "*exterior maintenance, repair, & decoration*". We believe this is due to the fact that these activities tend to have large variations in their environment and scene.

4.5 Conclusion

The ability to anticipate and describe what might happen next is a fundamental capability of human beings. In this chapter, we have tackled the problem of caption-

ing future frames in a video given the currently observed frames. We have proposed an architecture that first predicts the convolutional features for the next event, then fuses the features with the context features coming from the current event. Finally, it uses the fused feature to generate the caption. Our experimental results demonstrate that the proposed approach outperforms the baseline.

Chapter 5

Image Change Captioning by Learning from an Auxiliary Task

5.1 Introduction

Change is an inevitable part of a dynamic environment. There has been much attention in the community for a variety of change detection tasks [37; 85; 92; 49; 77; 78; 100]. While localizing the change has been the cornerstone of these works, it requires a deeper level of understanding to be able to semantically refer to the change. From a user's perspective, describing (captioning) the change between two images provides a more meaningful way of understanding the difference between images. The task of change captioning aims to describe the change between two images by generating a detailed sentence about the change of objects in these images. Note that in this task, we are only interested in changes at the object level (*e.g.* changes in terms of object color, position, etc), *i.e.* we do not want to generate captions for

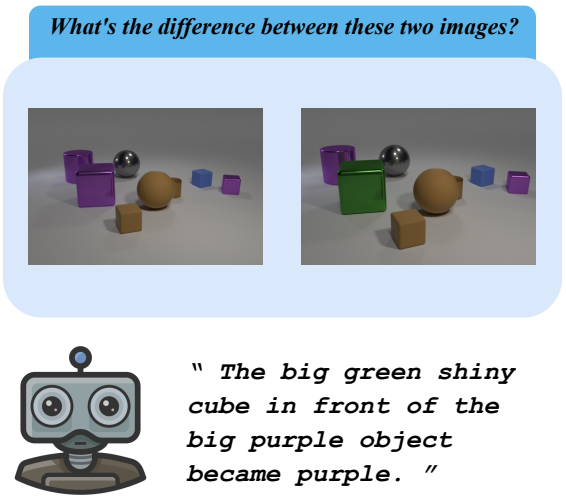


Figure 5.1: (Best viewed in color) Given two very similar images, the goal of change captioning is to describe the subtle difference between these two images. The difference can be in terms of objects' color, texture, position, addition or removal, etc.

changes in terms of viewpoints, light condition, etc. Some early works [49; 77; 78] in this area assume that there is always an object change between a pair of input images. This assumption does not always hold in a real-world application. In many cases, no object has been changed between the two images. Instead, it is only the viewpoint that is different (*e.g.* a moving robot can see a scene from two different viewpoints and should be able to distinguish that it is the same scene observed from different angles/lighting conditions). To address this limitation, recent work [82] proposes the task of robust changing captioning where not all pairs of images exhibit a change – some pairs can be distractors with similar scenes from different viewpoints.

In this chapter, we also consider the problem of change captioning with distractors. Our goal is to detect a change between a pair of images and generate a sentence

describing the change. While previous works mainly focus on proposing new network architectures to better tackle the problem [100; 82], we instead focus on improved learning strategies via multi-task learning.

Our proposed method consists of networks for two complementary tasks, namely the primary task and the auxiliary task. Composed query image retrieval discussed in Chapter 3, is a viable candidate to be used as auxiliary task due to the natural similarity with the image change captioning task.

The *primary task* in our formulation is the task of change captioning. Given two images, the goal of the primary task is to describe the difference between these two images. The input images are similar to each other and only differ in very subtle ways (*e.g.* an object's color, shape, or position is changed).

The *auxiliary task* in our formulation is the composed query image retrieval [117; 46; 27; 19]. This is an extension of the image retrieval task. The input to this task consists of a reference image and a sentence that defines the user's desired modification on the reference image. The model should then pick a candidate among a set of images. The candidate should look like the reference image, but differ from it according to the desired modification.

We argue that the aforementioned two tasks are naturally complimentary to each other. As a result, we can use the auxiliary task to help the primary task during learning. We propose a joint learning scheme for these two tasks. Inspired by the works on cycle-consistency [139; 97; 29; 45], our proposed learning scheme sequentially performs the primary and the auxiliary tasks in a way that the output of one task is the input to the other task. These two tasks can reinforce each other during training.

For example, the learning scheme forces the primary task to generate a good caption to be used as the input to the auxiliary network. If the caption generated by the primary network is not reliable, the auxiliary network would fail to retrieve the correct image. This will produce an auxiliary loss which is then used to further train the primary network.

The contributions of this chapter are manifold.

- First, we propose a new learning scheme for the task of change image captioning that involves using an auxiliary task to improve the performance of the primary task.
- Second, we further improve our proposed learning scheme by defining a new strategy for selecting hard negative samples for the auxiliary task of composed query image retrieval.
- Finally, we show that our proposed method can improve the performance of a change captioning task by performing empirical experiments on the CLEVR-Change dataset and the Spot-the-diff dataset.

5.2 Background

In this section, we provide some background on the image change captioning task and the composed query image retrieval task. Our proposed approach uses the composed query image retrieval as an auxiliary task to improve the performance of the primary image change captioning task.

Image Change Captioning: As mentioned in Sec. 5.1, the task of image change captioning is to generate a caption that describes the subtle but important change between two very similar images. Formally, given a pair of images (A, B), a model generates a caption describing what has been changed between A and B:

$$f(A, B; \theta_{\mathcal{P}}) \rightarrow \hat{C} \tag{5.1}$$

where $\theta_{\mathcal{P}}$ denotes the model parameters of the change captioning network and $\hat{C}$ represents the generated caption.

We use the Dual Dynamic Attention (DUDA) model [82] as the image change captioning network in our work. Although some recent work [100] has reported better performance, the code is not available yet. So we choose to build our approach based on DUDA. Here we briefly describe the network. DUDA consists of 3 major modules: a feature extraction module, a dual attention visual module, and a caption generator. The feature extraction module is a ResNet model trained offline and its weights are frozen. The feature extractor takes two images A and B as the input. It then produces 2D feature maps, $A_f, B_f \in \mathbb{R}^{d_v \times H \times W}$. A dual attention module is then applied on (A_f, B_f) and produces three feature maps capturing the visual information of the two images (A, B), and the difference between A and B. An LSTM-based captioning module then generates the words in the caption based on these three feature maps.

Composed Query Image Retrieval: This task can be seen as the opposite of the image change captioning task. Given an image and a caption describing some desired modification, the goal of this task is to retrieve the result of the modification among a set of image candidates [117; 46]. More formally, given an image A and a text sentence C describing the desired modification to be applied on A, the goal is to

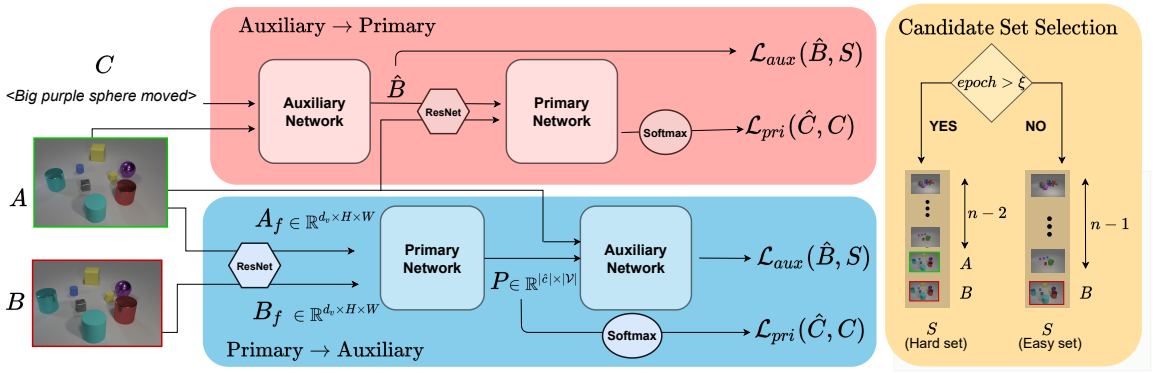


Figure 5.2: Overview of our approach. Given a triplet (A, B, C) from the training set, where (A, B) are image pairs and C is the caption describing their difference, our method involves jointly training two networks for the primary and the auxiliary tasks. The training involves two stages. In the first stage ("Primary → Auxiliary"), we feed (A, B) to the primary network to generate a caption $\hat{C}$, then feed $(A, \hat{C})$ as the input to the auxiliary network. In the second stage, we feed (A, C) to the auxiliary network to retrieve $\hat{B}$ from a set of S candidate images, then feed $(A, \hat{B})$ to the primary network. These two stages form a cycle consistency. The candidate set S is constructed differently depending on the training epoch. See the main text for details.

retrieve an image $\hat{B}$ from a candidate set S.

$$g(A, C|S; \theta_{\mathcal{A}}) \rightarrow \hat{B} \tag{5.2}$$

Let B be the resulting image of applying the modification C on the image A. Ideally, B and $\hat{B}$ should be the same image. We use a modified version of TIRG [117] model as our network for the composed query image retrieval. TIRG encodes the input image (A) and the modification text (C) using CNN and LSTM, respectively. Next,

the textual feature is added to the visual feature to generate a single feature vector representing the composed query. The feature vector for the composed query and the visual feature vector of each candidate image $I \in S$ are then projected onto a common feature space. Using a nearest-neighbor strategy, the closest candidate image to the composed query is selected and retrieved.

5.3 Our Approach

Our proposed approach is based on the following key observation. Image change captioning and composed query image retrieval are two closely related problems. In this chapter, we call them the primary task and the auxiliary task, respectively. If we have good models for both tasks, these two models should have the following cycle consistency [139]. Let (A, B, C) be a sample triplet in the training set where A and B are two images that are almost identical to each other but differ in very subtle details, and C is a sentence describing the subtle difference. Suppose we feed (A, B) to the primary network (image change captioning) and produce $\hat{C}$ as the output. We then feed $(A, \hat{C})$ to the auxiliary network (composed query image retrieval) and produces $\hat{B}$. We would expect B and $\hat{B}$ to be close. Similarly, if we first feed (A, C) to the auxiliary network to produce $\hat{B}$, then feed $(A, \hat{B})$ to obtain $\hat{C}$, we would expect C and $\hat{C}$ to be close (Fig. 5.2). Based on this observation, we jointly train these two networks. Note that the training dataset for the primary task can be easily re-purposed for the auxiliary task, so we do not need extra training data for the auxiliary task.

5.3.1 Joint Primary and Auxiliary Networks

The main novelty of this work is that we propose a new approach to train the primary network by coupling it with the auxiliary network. We use the DUDA and TIRG (see Sec. 5.2) as our primary and auxiliary networks, respectively. But our proposed method is not limited to these specific choices. It can be applied with any other network architectures for image chance captioning or composed query retrieval, respectively.

It's worth noting that for the auxiliary task, we do not use the method proposed in Chapter 3. This is on purpose and due to the fact that we are interested here to measure the effect of the new learning scheme on image change captioning. Using a strong network may hinder this evaluation. Therefore, we use a weaker network for composed query retrieval (i.e TIRG network) to assure the gained performance boost is indeed because of the new learning scheme.

Based on recent advances in using auxiliary task learning to improve the primary task [65; 108; 76; 99], we propose to couple the primary and the auxiliary tasks, then train them jointly. Given a (A, B, C) triplet representing two images (A, B) and a caption C describing the difference between these two images, our proposed method involves two stages for training.

Primary $\rightarrow$ Auxiliary: The first stage involves feeding the image input (A, B) into the primary network and using the generated caption along with one of the images A as the input to the auxiliary network. More specifically, consider $\theta_{\mathcal{P}}$ and $\theta_{\mathcal{A}}$ to be the parameters of primary and auxiliary networks, respectively. We first input A and B to the primary network. The primary network generates a caption $\hat{C} = \{\hat{w}^i\}_{i=0}^{|\hat{C}|}$

where $\hat{w}^i$ is the i-th word in the caption. Each word $\hat{w}^i$ is chosen from a vocabulary $\mathcal{V}$. We use P to denote the softmax scores for each word at each time step as the output of the primary network:

$$P = f(A, B; \theta_{\mathcal{P}}) \tag{5.3}$$

where $P \in \mathbb{R}^{|\hat{C}| \times |\mathcal{V}|}$ is a matrix representing the probability score of each word in the vocabulary being selected as the word at each time step in the generated caption. In other words, $\hat{C}$ is obtained by picking the word with the highest probability at each time step according to P.

Let $\mathcal{E}_{\hat{C}}$ be a matrix representing the embedding of each word in the caption $\hat{C}$. The input to the auxiliary network is then $(A, \mathcal{E}_{\hat{C}})$. The auxiliary network has a visual-textual module $k(A, \mathcal{E}_{\hat{C}}; \theta_{\mathcal{K}})$ which extracts a feature vector of $(A, \mathcal{E}_{\hat{C}})$. For a candidate image B for retrieval, the auxiliary network uses an image encoder $g(B; \theta_{\mathcal{G}})$ to extract a feature vector of B. Note that $k(\cdot, \cdot; \theta_{\mathcal{K}})$ and $g(\cdot; \theta_{\mathcal{G}})$ have the same dimensions. The similarity between $k(A, \mathcal{E}_{\hat{C}}; \theta_{\mathcal{K}})$ and $g(B; \theta_{\mathcal{G}})$ is used to measure how good B is as the retrieved result.

Note that if we naively feed discrete words in $\hat{C}$ to an LSTM module to get the embedding vector $\mathcal{E}_{\hat{C}}$, this will make the pipeline non-differentiable. As a result, the learning signal will not be able to propagate from the auxiliary network to the primary network. To address this issue, we propose to use a soft word selection based on their softmax scores (i.e. P) instead of words themselves. Let $W_{emb} \in \mathbb{R}^{|\mathcal{V}| \times d_e}$ be the word embedding matrix for the auxiliary network where d_e is the dimension of the word embedding, we define a soft word embedding layer as follows:

$$\mathcal{E}_{\hat{C}} = P \times W_{emb} \tag{5.4}$$

where $\mathcal{E}_{\hat{C}} \in \mathbb{R}^{|\hat{C}| \times d_e}$ is the soft embedding of words in the generated caption.

We then use the image A and the generated caption $\hat{C}$ (represented as $\mathcal{E}_{\hat{C}}$) as the input to the auxiliary network. The auxliary network also has access to a set of n candidate images $S = \{I_j\}_{j=1}^{n-1} \cup B$. The goal of the auxiliary network is to retrieve one image from S that best matches $(A, \hat{C})$. This can be achieved using a batch-based classification [117]. Let $\theta_{\mathcal{A}} = \{\theta_{\mathcal{G}}, \theta_{\mathcal{K}}\}$ be the parameters of the auxiliary network. We are given a batch β of training examples. Each sample i in the batch has the form $(A_i, \mathcal{E}_i, B_i)$ where

$$\mathcal{E}_i = f(A_i, B_i; \theta_{\mathcal{P}}) \times W_{emb}, \quad i = 1, 2, ..., \beta \tag{5.5}$$

We can define the following batch-based classification loss:

$$\mathcal{L}_{aux} = \frac{1}{\beta} \sum_{i=1}^{\beta} -\log\Big\{ \frac{e^{\left\langle k(A_i, \mathcal{E}_i; \theta_{\mathcal{K}}), g(B_i; \theta_{\mathcal{G}}) \right\rangle}}{\sum_{j=1}^{\beta} e^{\left\langle k(A_i, \mathcal{E}_i; \theta_{\mathcal{K}}), g(B_j; \theta_{\mathcal{G}}) \right\rangle}} \Big\} \tag{5.6}$$

where $< \cdot, \cdot >$ is the dot-product of two vectors as the similarity measure.

Auxiliary → Primary: The second stage for the training starts with the auxiliary network. Given (A, C) where C is the ground-truth caption describing the difference between A and B, the auxiliary network tries to pick B among a set of candidate images $S = \{I_j\}_{j=1}^{n-1} \cup B$. However, since the hard selection operation is not differentiable, we adopt a soft selection strategy as follows. We first compute the joint representation of (A, C) using the multi-modal module in the auxiliary network:

$$R = k(A, \mathcal{E}_C; \theta_{\mathcal{K}}) \tag{5.7}$$

where $\mathcal{E}_C$ is a matrix representing the embedding of words in the caption C by applying an embedding layer with weight W_{emb}, and $R \in \mathbb{R}^{d_r}$ is the joint representation

vector. We also encode each $I_j \in S$ using the visual module of the auxiliary network to obtain $\tilde{I}_j$:

$$\tilde{I}_j = g(I_j; \theta_{\mathcal{G}}) \quad \forall I_j \in S \tag{5.8}$$

where $\hat{I}_j \in \mathbb{R}^{d_r}$. We define $\tilde{S}$ as the matrix representing encoded images in the candidate set, i.e. $\tilde{S} = \{\tilde{I}\}_{j=1}^{n} \in \mathbb{R}^{n \times d_r}$. The soft selection is calculated by first computing a set of n weights denoted by ω:

$$\omega = Softmax(\tilde{S} \cdot R) \tag{5.9}$$

where $\omega \in \mathbb{R}^n$. We now softly select (generate) from the candidate set using weights calculated above:

$$\hat{B} = \sum_{j=1}^{n} \omega_j \tilde{I}_j \tag{5.10}$$

We now feed $(A, \hat{B})$ as input to the primary network. The output of the primary network at each time step i is a $|\mathcal{V}|$-sized vector p_i representing the Softmax scores for each word in the vocabulary. To generate the predicted caption, we can take the word that maximizes the score as the predicted word in i-th time step according to p_i. To calculate the primary network's loss function during training, we use the negative likelihood of the words in the ground-truth caption according to the Softmax output:

$$\mathcal{L}_{pri} = -\sum_{w^i \in C} log(p_i(w^i)) \tag{5.11}$$

$$C = \{w^1, \cdots, w^{C_n}\} \quad C_n = |C| \tag{5.12}$$

5.3.2 Model Training

We jointly train primary and auxiliary networks end-to-end for 60 epochs using the Adam optimizer [56]. The learning rate is set to $5e-3$. Following [82], we also

use a pre-trained ResNet 101 trained on ImageNet [25] as our feature extractor for the primary network. Other modules are trained from scratch. We alternate between the two stages from one batch to another.

Negative Sample Selection: Selecting the right candidate set for the auxiliary network is crucial since it has a direct effect on the primary network. Selecting an easy-to-pick set of candidates will cause the auxiliary network to converge quickly. This will cause $\mathcal{L}_{aux}$ to diminish fast and provide little gradient for the primary network to train. On the other hand, a very hard set of candidates at the beginning of the training prevents the network from converging since $\mathcal{L}_{aux}$ will be high.

To circumvent this issue, we propose the following curriculum learning strategy. In the early epochs of training, we provide the auxiliary network with a relatively easy set of candidates to choose from. In the later epochs, we switch to a harder set of images. More specifically, during early epochs, we use $n-1$ random images plus B to form the candidate set S. This helps the model to first learn to distinguish between B and other images in S, *i.e.* $S = \{I_j\}_{j=1}^{n-1} \cup B$. Once the model has learned this task ($\mathcal{L}_{aux}$ becomes small), we construct S differently. We randomly select $n-2$ and add to them both A and B to form S, *i.e.* $S = \{I_j\}_{j=1}^{n-2} \cup \{A, B\}$. So we have:

$$\begin{cases} S = \{I_j\}_{j=1}^{n-1} \cup B & epoch < \xi \\ \\ S = \{I_j\}_{j=1}^{n-2} \cup \{A, B\} & epoch \geq \xi \end{cases} \tag{5.13}$$

where *epoch* denotes the current training epoch and ξ is a predefined threshold which defines the epoch at which we start using the hard negative sampling strategy.

From the captioning perspective, the easy set helps the model to learn to produce

generally good captions, while the hard samples push the model to focus on subtle details in the images and generate fine-detailed captions.

Putting everything together, the network is trained using a weighted loss function:

$$\mathcal{L}_{final} = (1 - \gamma)\mathcal{L}_{pri} + \gamma\mathcal{L}_{aux} \tag{5.14}$$

where $0 \leq \gamma \leq 1$ determines the weight for the auxiliary loss function.

Method	B4	C	M	R	S
Capt-Pix-Diff [82]	30.2	75.9	23.7	-	17.1
Capt-Rep-Diff [82]	33.5	87.9	26.7	-	19.0
Capt-At [82]	42.7	106.4	32.1	-	23.2
Capt-Dual-Att [82]	43.5	108.5	32.7	-	23.4
DUDA [82]	47.3	112.3	33.9	-	24.5
VAM [100]	50.3	114.9	37.0	69.7	30.5
Ours	**51.2**	**115.4**	**37.7**	**70.5**	**31.1**

Table 5.1: Performance of the proposed method on the entire CLEVR-Change dataset. Metrics indicated by "-" are not reported by the authors. Numbers are taken from respective works. Our proposed method improves the performance of DUDA which uses the same base network. B4, C, M, and R, S are BLEU-4, CIDEr, METEOR, and ROUGE-L, Spice, respectively.

Method	B4	C	M	R	S
Changed Pairs Only					
DUDA	42.9	94.6	29.7	-	19.9
Ours	**49.9**	**101.3**	**34.3**	**65.4**	**27.9**
Distractor Pairs Only					
DUDA	59.8	110.8	45.2	-	29.1
Ours	**62.4**	**116.3**	**50.5**	**53.9**	**35.0**

Table 5.2: Performance of the proposed method evaluated only on the changed pairs (top) vs. the performance evaluated only on the distractor pairs (bottom) on the CLEVR-Change dataset. B4, C, M, and R, S are BLEU-4, CIDEr, METEOR, and ROUGE-L, Spice, respectively.

Method	B4	C	M	R
DDLA [49]	0.081	0.340	0.115	0.283
DUDA	0.081	0.325	0.118	0.291
Ours	0.081	0.345	0.125	0.299

Table 5.3: Performance of our method against DUDA and DDLA on the Spot-the-diff dataset. B4, C, M, and R are BLEU-4, CIDEr, METEOR, and ROUGE-L, respectively.

Method	B4	C	M	R	S
Ours (easy set only)	51.0	115.2	37.3	70.4	30.8
Ours (easy+hard set)	**51.2**	**115.4**	**37.7**	**70.5**	**31.1**

Table 5.4: Performance of the proposed method when only providing the auxiliary network with the easy set of candidates (first row) vs. the performance when using a dynamic strategy and switching to hard sample sets after a certain epochs. Other settings remain identical in both cases. Our method benefits from the dynamic strategy and the result has been improved. B4, C, M, and R, S are BLEU-4, CIDEr, METEOR, and ROUGE-L, Spice, respectively.

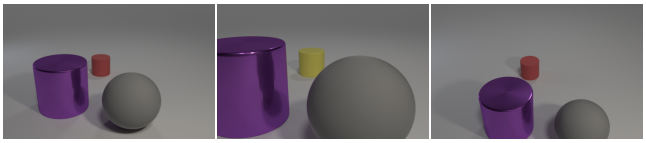
5.4 Experimental Results

We perform empirical experiments to evaluate the performance of the proposed method on the change captioning task.

5.4.1 Dataset and Setting

We use the CLEVR-Change [82] dataset to evaluate the performance of our approach. CLEVR-Change is a synthetic dataset that is generated using the CLEVR engine [52]. Due to its flexibility in generating different scenarios, CLEVR has become a standard tool to create diagnostic datasets for a variety of vision-language applications.

CLEVR-Change has 67660, 3976, and 7970 samples for training, validation, and test splits, respectively. The image pairs are categorized into two scenarios: distractor



(a) (b) (c)

Figure 5.3: (Best viewed in color) Examples of two types of pairs in the dataset. (a) and (b) form a "changed pair" since there is a change at the object level (red → yellow). (a) and (c) form a "distractor" pair since there is only viewpoint change.

pairs, and changed pairs. Distractor pairs are those in which no object has been changed between two images. However, the camera view has changed from one image to another. A successful model should predict that there has been no change for distractor pairs. Changed pairs are those samples in which there is a change at the object level from one image to the other. Changes can be 1) changing an object's color; 2) changing an object's texture; 3) adding a new object to the scene; 4) removing one object from the image; and 5) moving an object to a new position. Each image pair in the dataset is accompanied by a sentence describing the change or expressing the absence of any changes depending on the scenario. Fig. 5.3 shows a sample of each of these two scenarios.

To compare with other methods, we also report the performance of our method on Spot-the-difference dataset [49]. This dataset contains 13,192 images of mostly parking lots. Each pair of images differ in a subtle change. There are two key differences between Spot-the-diff and CLEVR-change: 1) the camera in Spot-the-diff is fixed while in CLEVR-change the viewing changes from one image to the other in the pair, and 2) there are no distrators in Spot-the-diff, i.e. one can assume that

there is always a change between two images. These differences make CLEVR-change a more challenging benchmark for the task of image change captioning.

We use PyTorch [84] for implementation and jointly train the primary and auxiliary networks end-to-end for 60 epochs using Adam optimizer [56]. The learning rate is set to $5e-3$. We report our experimental results in terms of BLEU@1, BLEU@2, BLEU@3, BLEU@4, CIDEr, SPICE, METEOR, Rouge-L [82].

5.4.2 Results

Results on CLEVR-Change: We present the results of our method on the entire CLEVER-Change dataset in Table 6.1. We compare our method against other state-of-the-art methods. Our approach outperforms DUDA [82] which uses the same base network. The performance of our method is also on par with VAM [100] which uses a different network architecture. Since the code of VAM is not released yet, we cannot build our approach based on VAM. Also, note that [100] has reported improved results using reinforcement learning as postprocessing. In order to keep the comparison fair, we report the results of the VAM version without this extra postprocessing in Table 6.1. Note that the focus of this work is not on proposing a new architecture for change captioning, but proposing a training scheme that can improve the performance of any given change captioning network including VAM.

Results on Changed Pairs: Table 5.2 (top rows) provides the results of evaluating our proposed method only on pairs of images that have a changed object. Our method outperforms the DUDA method which uses the same change captioning network as our primary network.

Results on Distractor Pairs: Finally we present our result on evaluating only on distractor pairs in Table 5.2 (bottom rows). These image pairs only have the camera angle/scene lighting change. Again we see a similar trend. Our method significantly outperforms DUDA.

Results on Spot-the-diff: We report the performance of our method on the Spot-the-diff dataset in Table 5.3. Again, our method outperforms other alternative approaches.

Qualitative Examples: We present some qualitative examples in Fig. 5.4. The proposed method generates captions that are semantically similar to the ground-truth captions. Note that in the last example, there is no change at the object level between the two input images. Instead, these two images only have a viewpoint change. The caption generated by our method correctly indicates that there is no change between these two images.

5.4.3 Ablation Study

We perform additional ablation studies to further analyze various aspects of the proposed approach. Specifically, we are interested to measure the effect of dynamic negative set selection on the overall performance. Also, we provide a break-down performance of our method on various types of change for the semantically changed pairs. All ablation studies are performed on the CLEVR-Change dataset.

Effect of Using Hard Negative Samples: To identify the effect of easy vs. hard candidate set, we perform two experiments and report the result in Table 5.4. For

the first experiment, we train the networks using only the easy candidate sets, *i.e.* S only contains one image from (A, B). In the second experiment, we start our training by constructing S as an easy set. After certain epochs (30 in our case), we start to provide the hard set for the auxiliary network, namely, put both A and B among the n candidate images. Table 5.4 clearly demonstrates that the later strategy is superior to the former strategy which only uses easy candidate sets.

The result of this experiment make intuitive sense. Consider the case where the generated caption from the primary network ($\hat{C}$) along with image A is the input to the auxiliary network. When the easy set is solely used to train the auxiliary network, eventually the auxiliary network learns to distinguish the right image from the candidate set even if the generated caption $\hat{C}$ is not very accurate. This is because A and B are very similar to each other and only differ in one change. So if only easy candidate sets are provided to the auxiliary network, it eventually learns to ignore the input caption and picks the image in the candidate set that is most similar to the input image. This causes the auxiliary loss to become extremely low and does not provide much gradient flow for the primary network supervision.

To avoid this issue, it is essential to dynamically increase the task difficulty for the auxiliary task so that it does not converge prematurely or learn to ignore $\hat{C}$. Using this dynamic method, the auxiliary loss provides a much more reasonable gradient flow throughout the training process and improves the performance of the primary network as seen in Table 5.4.

Result per Change Category: We also provide the break-down results of our

	CIDEr					METEOR					SPICE				
Method	C	T	A	D	M	C	T	A	D	M	C	T	A	D	M
CaptPixDiff	4.2	16.1	30.1	27.1	18.0	7.4	16.0	24.4	20.9	18.2	1.3	6.8	11.4	10.6	9.2
CaptRepDiff	44.5	21.9	50.1	49.7	26.5	19.2	18.2	25.7	23.5	18.9	8.2	8.8	12.1	12.0	9.6
CaptAt	112.1	75.9	91.5	98.4	49.6	30.5	25.4	30.2	31.2	22.2	17.9	16.3	19.0	22.3	14.5
CaptDualAtt	115.8	82.7	85.7	103.0	52.6	32.1	26.7	29.5	31.7	22.4	19.8	17.6	16.9	21.9	14.7
DUDA	120.4	86.7	108.2	103.4	56.4	32.8	27.3	33.4	31.4	23.5	21.2	18.3	22.4	22.2	15.4
Ours	**120.8**	**89.9**	**119.8**	**123.4**	**62.1**	**36.1**	**30.4**	**37.8**	**36.7**	**27.0**	**29.7**	**27.4**	**31.4**	**30.8**	**23.5**

Table 5.5: Performance of the proposed method compared with other state-of-the-art approaches on each category of change. The changes categories are : Color Change (C), Texture Change (T), Adding an object (A), Deleting an Object (D), and Moving an Object (M). Numbers for other methods are taken from [82]

method for different change categories in Table 5.5. The proposed method effectively improves the performance of the DUDA network by a large margin in almost every category.

5.5 Conclusion

We have proposed a new training scheme for the task of image change captioning. Our proposed scheme uses the composed query image retrieval as an auxiliary task to improve the primary task of image change captioning. The two networks of these tasks are jointly trained in a sequential fashion. Our learning scheme enables the auxiliary task to provide an extra level of supervision for the primary task. This scheme along with a proposed candidate set selection for the image retrieval task proves to

be effective for improving the performance of primary network. Our experimental results demonstrate the effectiveness of our proposed approach.

Ours: *the tiny green metal block that is behind the big purple matte thing is no longer there*

GT: the small green metallic block that is behind the large red metallic object has disappeared

Ours: *the small yellow shiny cylinder that is behind the small cyan rubber thing turned cyan*

GT: the small yellow shiny cylinder that is to the left of the small brown shiny thing became cyan

Ours: *hte large brown metallic sphere that is behind the big cyan matte thing turned cyan*

GT: the big brown metal sphere behind the yellow metallic thing turned cyan

Ours: *the scene remains the same*

GT: there is no change in the scene

Figure 5.4: Qualitative examples. The first three rows depict the cases where there is a change at the object level between the two input images. The last example shows a case in which there is no semantic change between two images.

Chapter 6

Few-Shot Personality-Specific Image Captioning via Meta-Learning

6.1 Introduction

Image captioning has been an area of active research in computer vision. Most existing work [125; 115; 6; 50; 134; 79; 137] focuses on learning generic image captioning models. While being popular, these models tend to produce captions stating obvious things about an image. This is not surprising since the datasets on which these models are trained are collected objectively, trying to ignore annotator-specific style. As a result, these captions are not very engaging to users. To generate truly engaging captions, we need to incorporate the personality traits of users.

For instance, a birthday scene as shown in Fig. 6.1 would be described as *women*

Figure 6.1: Most existing image captioning models produce generic captions stating obvious things in an image, e.g. "a women is cutting the cake" in this case. These captions are not very engaging for humans. In this work, we consider the personality-specific image captioning. We assume that we know the personality trait (e.g. "sweet", "anxious") of the user. Our goal is to produce image captions with a style that is consistent with the personality of the user. These personality-specific captions are more engaging to the user.

is cutting a cake. by a generic caption generator. While a person who has a "*sweet*" personality would probably prefer to describe the same image as *I love small birthday parties, like this.*. Or a person who has an "*anxious*" personality trait would describe it as *Dealing with a birthday party like this makes me panic.* This difference in users' preference requires an image captioning system that takes into account each user's personality.

There has been an effort in proposing captioning systems that generate more appealing descriptions. For instance, Guo et al. [38] propose a stylistic caption generation system that takes a specific style (e.g. romance, humor) and generates the captions according to that style. Most work [18; 38; 3; 33] in stylistic captioning is limited to a limited number of styles (less than 5).

Recently, Shuster et al. [101] propose the problem of personality-specific captioning. In this work, the personality trait is used as an additional input to the image captioning model. The model is learned to incorporate this personality trait and generate captions specific to this personality. The work in [101] considers more than 200 personality traits.

However, a limitation of personality-specific captioning as proposed in [101] is that it is a purely supervised approach. To train an image captioning model for a particular personality trait requires having access to enough labeled training data for this personality trait. This is an expensive process and is not scalable.

In this chapter, we plan to study a more challenging problem called the *few-shot personality-specific image captioning*. During training, we have access to training data for a set of personality traits. During testing, we are given a new personality trait. We only have a small number of training data for this new personality trait. Our goal is to get a captioning model specific to this new personality trait from these few-shot examples.

Our solution will be based on meta-learning, in particular the model-agnostic meta-learning (MAML) [31]. We will consider each personality trait as a "*task*" in meta-learning.

6.2 Problem Setup

Given a specific personality trait t, our goal is to obtain a image captioning model $f_t : \mathcal{I} \rightarrow \mathcal{S}$ that takes an image $\mathcal{I}$ as its input and generates a caption $\mathcal{S}$. Here we would like the model f_t to be an image captioning model specifically tuned to the particular personality trait t instead of being a generic image captioning model. A naive solution to this problem is to collect a large amount of training data for each personality trait and train a separate image captioning model using the corresponding training data. However, this approach is not scalable since it is expensive to collect enough training data for all possible personality traits.

In this chapter, we consider the few-shot setting for the personalized image captioning. During training, we have training data for a set of personalized traits. During testing, we are given one or more *new* personality trait(s) that did not appear during training. For each new personality trait, we assume that we have a small amount of training data. Our goal is to effectively obtain a new image captioning model for this new personality trait using only a small amount of training data.

6.3 Our Approach

We propose to solve the few-shot personalized image captioning using a meta-learning framework, in particular the MAML approach [31]. In the following, we describe how to formulate the few-shot personalized image captioning using the MAML framework.

In the classic setting of image captioning, we are given a training dataset D^{train}

and a test dataset D^{test}. Each dataset consists of pairs of images and their ground-truth captions. The goal is to learn a model $f : \mathcal{I} \rightarrow \mathcal{S}$ that maps the input image $\mathcal{I}$ to a caption $\mathcal{S}$. We learn the parameters of the model f by optimizing a loss function defined on D^{train}. Once the model is trained, we evaluate its performance on the test dataset D^{test}.

In the few-shot setting of personalized image captioning, the model is trained during a meta-training where the training dataset (also called meta-train dataset) D^{tr} consists of a collection of N datasets N, i.e. $D^{tr} = \{D_t^{tr}\}_{t=1}^{N}$. Each D_t^{tr} is a dataset corresponding to a personality trait t. The model is trained during a meta-training stage on a set of "tasks", where each task is a few-shot personalized image captioning problem. Each task has its training and validation sets that are constructed from D^{tr}. The model is learned in a way to enable effective adaption to new personality traits using a few examples.

6.3.1 Model Architecture

We use convolutional image captioning network (ConvCap) [6] as the base network in this work, however, the framework is not bound to any particular captioning network. ConvCap consists of three main sub-modules: a visual feature extraction network, a visual attention module, and a captioning module. The image feature extractor is a VGG16 [103] network pretrained on imagenet. We take FC-7 features of VGG-16 as visual features and FC-5 features of VGG16 as attention module input. Therefore, the visual features for each image are a 4096-d vector, and the visual attentions are 7×7 attention value applied on $7 \times 7 \times 512$ output of the FC-5 layer.

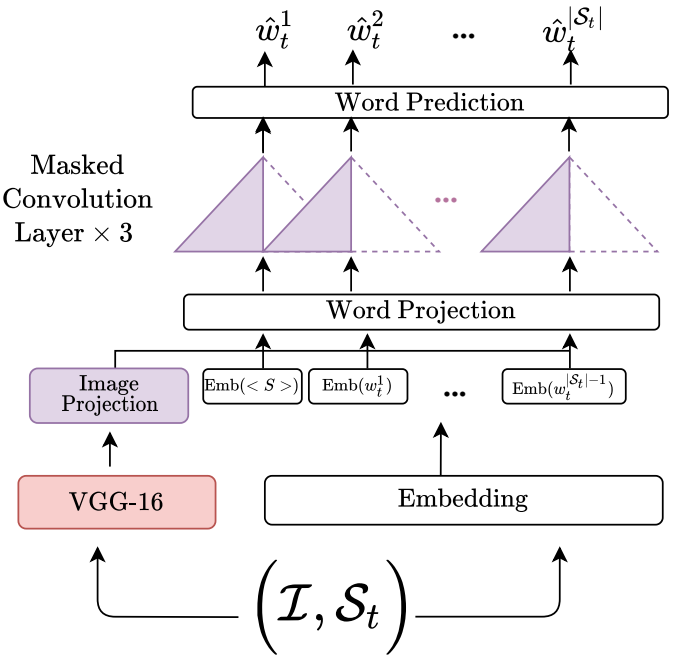


Figure 6.2: Model Architecture for ConvCap network. ConvCap uses VGG as its visual feature extractor, and replaces recurrent network for caption generation with several masked convolution layers. Refer to [6] for more details.

The caption generation network is a fully convolutional module. It has two embedding layers. One is used to encode the words in the sentence and the second embedding layer is used to project the visual features into a 512-d space. Each embedded word ($\texttt{Emb}(w) \in \mathbb{R}^{512}$) is further added to the projected visual features ($\in \mathbb{R}^{512}$) and is input to the first masked convolution layer.

The attended visual features (i.e. output of attention module) are added to the output of the first convolution layer. Finally, the output of the last masked convolution layer is used to predict the next word using an FC layer. Fig. 6.2 shows the diagram of the ConvCap model.

6.3.2 Meta-Training

During meta-training, we have a meta-train dataset $\mathcal{D}^{tr}$ consisting of N smaller datasets.

$$\mathcal{D}^{tr} = \{\mathcal{D}_t^{tr}\}_{t=1}^{N} \tag{6.1}$$

Here D_t^{tr} is the dataset for the personality trait t. It consists of N_t (image, caption) pairs:

$$\mathcal{D}_t^{tr} = \{(\mathcal{I}^i, \mathcal{S}_t^i)\}_{i=1}^{N_t} \tag{6.2}$$

where $\mathcal{S}_t^i$ is a caption describing image $\mathcal{I}^i$ by someone who has a personality trait t (e.g. optimistic, pessimistic, ...).

We parameterize the image feature extractor and captioning module by ϕ and θ, respectively. During each iteration of the meta-training, we sample a set of tasks as follows. With $\mathcal{B}$ being the size of meta-batch, a subset of $\mathcal{B}$ personality traits $\{t_1, \cdots, t_{\mathcal{B}}\}$ from N personality traits in $\mathcal{D}^{tr}$ are selected. From each trait t_i, K and $L \geq K$ samples are randomly picked to build the training set $\mathcal{D}_{\mathcal{T}_i}^{tr}$ and the test set $\mathcal{D}_{\mathcal{T}_i}^{ts}$ for task $\mathcal{T}_i$. Note that K is chosen to be a small number to emulate the few-shot scenario that we will encounter during testing.

For an image $\mathcal{I}$, the image features are extracted using as $f_\phi(\mathcal{I})$ using the pre-trained image feature network with parameters ϕ. In our formulation, weights of this network are frozen and only the captioning module's weights are updated. Given the extracted image features, the captioning module generates the output captions using another network $g_\theta(\cdot)$ with parameters θ, i.e. the generation caption is $g_\theta(f_\phi(\mathcal{I}))$). Using a cross-entropy loss function and through back-propagation, weights of the

captioning module are updated $\theta \rightarrow \theta'_i$ w.r.t to the personality trait t_i:

$$\mathcal{L}_i^{tr}(\theta) = \sum_{(\mathcal{I},\mathcal{S})\in\mathcal{D}_{\mathcal{T}_i}^{tr}} \ell_{CE}(g_\theta(f_\phi(\mathcal{I})), \mathcal{S}) \tag{6.3}$$

$$\theta'_i \leftarrow \theta - \alpha\nabla_\theta\mathcal{L}_i^{tr}(\theta) \tag{6.4}$$

where $\ell_{CE}(\cdot,\cdot)$ is a word-wise cross entropy function and α is the adaption (i.e. inner loop) learning rate. We use $(\mathcal{I},\mathcal{S})$ to denote an (image, caption) pair in $\mathcal{D}_{\mathcal{T}_i}^{tr}$Note that we use $\mathcal{L}_i^{tr}(\theta)$ to make it explicit that $\mathcal{L}_i^{tr}$ is a function of θ. The parameters θ can be interpret as a "global model", while θ'_i can be interpreted as the model tuned to the personalized trait in the task $\mathcal{T}_i$. In MAML, we obtain θ'_i by taking a few gradient updates from θ (Eq. 6.4). We can then calculate the loss of θ'_i on the corresponding test set $\mathcal{D}_i^{ts}$ of this task as:

$$\mathcal{L}_i^{ts}(\theta'_i) = \sum_{(\mathcal{I},\mathcal{S})\in\mathcal{D}_{\mathcal{T}_i}^{ts}} \ell_{CE}(g_{\theta'_i}(f_\phi(\mathcal{I})), \mathcal{S}) \tag{6.5}$$

Note that here we use $\mathcal{L}_i^{ts}(\theta'_i)$ to emphasize that $\mathcal{L}_i^{ts}$ is a function of θ'_i.

Finally, a gradient step is taken on θ by minimizing the following objective:

$$\theta \leftarrow \theta - \beta\sum_{i=1}^{\mathcal{B}} \nabla_\theta\mathcal{L}_i^{ts}(\theta'_i) \tag{6.6}$$

where β is a outer loop learning rate in MAML. Note that in Eq. 6.6, $\mathcal{L}_i^{ts}(\theta'_i)$ is a function of θ'_i, but the gradient is taken with respect to θ.

6.3.3 Few-Shot Adaptation to New Personality Traits

After meta-training, we have obtained a global model θ. Note that θ is trained in a way such that it can be quickly adapt to a new personality trait with only a few

Algorithm 1: Meta-training algorithm for few-shot personality-specific image captioning

Input : $\mathcal{D}^{tr}$ containing N different personality traits and N_i pairs of (image,caption) for each personality trait t_i

Input : α and β, inner and outer loop learning rates

Input : ϕ, θ, image feature extraction, and captioning module initial parameters

while *not converged* **do**

Sample a batch of $\mathcal{B}$ personality traits from $\mathcal{D}^{tr}$;

for *each personality trait* t_i **do**

Randomly select K and L pairs of (image,caption) from t_i;

$\mathcal{D}_i^{tr} = \{(\mathcal{I}^j, \mathcal{S}_i^j)\}_{j=1}^{K}$;

$\mathcal{D}_i^{ts} = \{(\mathcal{I}^j, \mathcal{S}_i^j)\}_{j=1}^{L}$;

Form task $\mathcal{T}_i = \{\mathcal{D}_i^{tr}, \mathcal{D}_i^{tr}\}$;

Calculate $\mathcal{L}_i^{tr}(\theta)$ using K samples of $\mathcal{D}_i^{tr}$;

Update $\theta' \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_i^{tr}(\theta)$;

Evaluate θ' on $\mathcal{D}_i^{ts}$ and calculate $\mathcal{L}_i^{ts}(\theta')$;

Accumulate gradient of $\mathcal{L}_i^{ts}(\theta')$ w.r.t. to θ ;

end

Update $\theta \leftarrow \theta - \beta \sum_{i=1}^{\mathcal{B}} \nabla_\theta \mathcal{L}_i^{ts}(\theta_i')$ using the accumulated gradients over tasks ;

end

examples. During testing, we are given a new personality trait that does not appear during the meta-training stage. For this new personality trait, we have K pairs of (image, caption) pairs $\{\mathcal{I}^j, S^j\}_{j=1}^K$, where K is a small number. Our goal is to quickly adapt the image captioning model to this new personality trait using these few-shot K examples. We simply perform new gradient updates on θ using the loss defined on these K examples.

$$\mathcal{L}^{tr} = \sum_{j=1}^{K} \ell_{CE}(g_\theta(f_\phi(\mathcal{I}^j)), \mathcal{S}^j) \tag{6.7}$$

$$\theta' \leftarrow \theta - \alpha \nabla_{\theta_e} \mathcal{L}^{tr} \tag{6.8}$$

where θ' is the adapted captioning module for the new personality trait. We can now use $\{\phi, \theta'\}$ to generate captions with this personality trait for any test images.

6.4 Experiments

We first describe the details of the dataset and implementation details. We then introduce several baseline methods for comparison. Later, we present the experiment results and ablation studies.

6.4.1 Dataset and Implementation

Datasets: Since this is the first work on few-shot personality-specific image captioning, there are no existing datasets that we can directly use. Instead, we build our dataset by repurposing the Personality-Captions dataset in [101]. The Personality-Captions dataset is a collection of triplets $(\mathcal{I}, \mathcal{S}_t, t)$ where each image $\mathcal{I}$ is accompanied

Method	B1	B2	B3	B4	M	R	C	S
PRETRAINED	0.12	0.01	0.00	0.00	0.55	0.13	0.01	0.21
PRETRAINED+FINETUNE	9.66	3.18	1.34	0.66	4.40	10.59	8.82	3.72
PRETRAINED+METATEST	0.12	0.00	0.00	0.00	0.69	0.13	0.09	0.001
OURS W/O ADAPTATION	9.78	3.06	1.12	0.54	4.30	10.47	8.45	3.26
OURS	**10.65**	**3.45**	**1.53**	**0.64**	**4.45**	**10.64**	**9.10**	**3.62**

Table 6.1: Performance of the proposed method on the Personality Caption dataset using our proposed data split protocol. The proposed method outperform the baselines for all the metrics while only using 10 labeled samples for adaptation.

with a sentence caption $\mathcal{S}_t$ expressing the image when a person with a personality trait t describes the given image. In total, the Personality-Captions dataset has about 242K image-caption pairs that are originally divided into a train set of size 186,858 pairs, in addition to a validation and test set of 5,000 and 50,000 pairs, respectively. Each sentence on average has 11.1 words in it. There are more than 200 different personality traits on this dataset.

Since the original train, validation, and test splits are not disjoint with respect to personality traits, we cannot use the original splits of [101] for our few-shot formulation. Instead, we propose a new split of the dataset into meta-training and meta-testing sets, so that these two sets have disjoint personality traits. In particular, the meta-training dataset contains 106 personality traits and their associated image-caption pairs. The remaining 108 personality traits (and their respective pairs) are used as the meta-testing set. One personality trait in the dataset had only a single

Method	B1	B2	B3	B4	M	R	C	S
First-Order vs. Hybrid-Order Gradients								
Ours w/ Hybrid	**10.65**	**3.45**	**1.53**	**0.64**	**4.45**	**10.64**	**9.10**	**3.62**
Ours w/ First-order	9.27	2.87	1.16	0.60	4.30	9.91	7.88	3.24
GT VS. Predicted Words in Training								
Using Predicted Words	6.12	1.20	0.56	0.09	3.06	10.98	7.10	1.27
Using GT Words	**10.65**	**3.45**	**1.53**	**0.64**	**4.45**	**10.64**	**9.10**	**3.62**
Convergence Speed								
Pre+FT (1000 Steps)	6.45	1.74	0.52	0.07	2.71	9.47	2.00	0.01
Pre+FT (2000 Steps)	8.01	2.32	0.89	0.41	3.39	9.82	3.18	0.19
Pre+FT (40000 Steps	9.66	3.18	1.34	0.66	4.40	10.59	8.82	3.72
Ours (1000 Steps)	8.21	2.40	0.85	0.35	4.22	10.20	4.20	0.06
Ours (2000 Steps)	8.80	2.91	1.12	0.50	3.94	10.19	5.2	1.13
Ours (12000 Steps)	10.22	3.26	1.28	0.62	4.28	10.57	7.81	2.84

Table 6.2: Evaluating different aspect of our method via ablation studies.

image-caption pair and thus is removed from the final set, making the total number of used personality traits to 214 (instead of 215 in [101]).

The complete list of personality traits used for meta-training and meta-testing is available in the supplementary material.

Implementation details: We use PyTorch [84] and the Learn2Learn library [10] for implementation. We remove words in the vocabulary that occur less than 10 times in the captions. The final size of our vocabulary is 7075. During fast adaptation, we

use SGD as our optimizer with an initial learning rate of $5e-3$. The outer loop is optimized using RSMProp [36] with an initial learning rate of $5e-4$.

In our experiments, we use a meta batch size of 64. In all of our experiments, K is set to 10 (i.e. 10-shot) for adaptation, and L is set to 20 samples for inner loop evaluation and meta-update. During meta-training and meta-testing, we take two adaptation steps on the base network. We replace the word embedding layer in ConvCap with GloVE weights [86] and replace the word prediction layer (i.e. last FC layer) according to our vocabulary size.

We report the performance of our method (and other baselines) on the meta-test set using the following metrics. BLEU@1, BLEU@2, BLEU@3, BLUE@4, METEOR [11], ROUGE-L [61], CIDEr [111], and SPICE [4].

Method	B1	B2	B3	B4	M	R	C	S
Ours (Explicit)	10.33	3.30	1.30	0.52	4.35	10.46	6.70	2.90
Ours (Implicit)	10.65	3.45	1.53	0.64	4.45	10.64	9.10	3.62

Table 6.3: Performance of our method when using explicit and implicit personality trait information. Using only implicit information is adequate for the model to pick up on specific clues associated with each personality trait caption style, and learn them using a few provided adaptation samples.

6.4.2 Experimental Results and Comparisons

Since this work deals with a new problem setting, there is no previous work that we can directly compare with. Nevertheless, we define several baselines in this section

Method	B1	B2	B3	B4	M	R	C	S
OURS ($K = 1$)	8.51	2.64	1.07	0.49	3.91	9.72	7.48	2.81
OURS ($K = 5$)	9.22	3.01	1.25	0.60	4.25	10.31	8.55	3.25
OURS ($K = 10$)	**10.65**	**3.45**	**1.53**	**0.64**	**4.45**	**10.64**	**9.10**	**3.62**

Table 6.4: Comparing the performance of our model using different K values. The best performance is achieved using $K = 10$. However, the performance does not degrade severely when we use 5 samples for adaptation instead of 10. In contrast, one can see a significant drop in performance when only one sample is used for adaptation.

for comparison. Training of our method and other baselines begins from the same pretrained ConvCap model.

- PRETAINED: this baseline directly uses the pretrained ConvCap model on test data without any adaptation.

- PRETRAINED + FINETUNE: this baseline fine-tunes a pretrained ConvCap model on the meta-train set. Afterward, its performance on the meta-test set is evaluated. Since the captions and personality traits in the meta-test set are different from those of meta-train, this baseline illustrates the domain shift and its effect on the trained model.

- PRETRAINED + META-TEST: This is a baseline where a pretrained model is taken and only meta-testing is performed during inference. By comparing our method against this baseline, we can have a clearer picture of how important our meta-training phase is.

- OURS W/O ADAPTATION: this baseline performs the meta-training starting from the pretrained ConvCap model. But it directly uses the model obtained from meta-training during testing without adaptation.

Table 6.1 compares our method against these baselines. Our proposed method significantly outperforms all the baselines. The last two rows of Table 6.1 show that when using the adaptation samples available in testing time, the performance increases especially in the case of BLEU-1 from 9.78 to 10.65, from 8.45 to 9.10 for CIDEr, and from 3.26 to 3.62 for SPICE method.

Nonetheless, we observe is that meta-training has a much more significant effect than meta-testing. As seen on the first and third rows of Table 6.1, if only meta-testing was performed on a pretrained model (i.e. taking advantage of few available labeled samples in testing time), the performance would not improve significantly. This highlights that meta-training has led the model to learn a new personality trait caption style fast and efficiently. Figure 6.3 shows qualitative results for our method.

6.4.3 Ablation Studies

We perform additional ablation studies to gain further insights into our proposed method.

First-order vs second-order MAML: Training a MAML-based system involves using second-order gradient calculation which is expensive. Therefore, most few-shot learning systems such as MAML[31], and REPTILE [74] propose to use the first-order approximation method to overcome the computational expense. MAML++ [9] uses a hybrid approach. At the beginning of meta-training, the first-order gradient

is used up to a point. After that, it switches to second-order gradient calculation for better generalization. MAML++ shows that this hybrid method improves the performance in the image classification task. To test if this hybrid method improves the performance of our problem, we conduct an ablation study.

As shown in the top section of Table 6.2, we notice no significant difference between using first-order gradient only and the hybrid method. However, consistent with [74], we observe a more stable meta-training phase when using the first-order gradient at the beginning of the meta-training loop (i.e. hybrid-order).

Using ground-truth vs predicted words in training: Caption generation is a sequential process. Regardless of how a captioning system is trained, we need to generate words sequentially in the inference/testing phase, since the generation of each word depends on what has been generated so far in the output sentence.

However, during training, we have access to the ground truth captions. Therefore, at each time step t, we can use the ground truth words in previous time steps $(w_1, \cdots, w_{t-1})$ as the input, not the actual predicted words $(\hat{w}_1, \cdots, \hat{w}_{t-1})$. This allows models such as ConvCap or BERT-based systems to speed up training by being able to run in parallel.

Using the ground-truth words during training creates a discrepancy between training and testing. In this ablation study, we analyze the performance when using the ground-truth words versus predicted words. As shown in the middle section of Table 6.2, using predicted words degrades the performance to some extent. This is probably because in this case, the back-propagation needs to unfold across the time steps. This could potentially cause gradient vanishing problems.

Convergence: Finally, we compare the convergence of PRETRAINED+FINETUNE with that of our method according to the number of gradient steps needed for convergence. The bottom part of Table 6.2 summarize the results for this comparison. We can see that our method is far more efficient in terms of gradient steps needed to converge. In other words, our proposed method can outperform the baseline in terms of convergence speed.

Personality trait embedding: Our method does not explicitly use the semantic information of personality traits. For example, it does not exploit the fact that "happy" and "cheerful" are similar personality traits.

Nonetheless, we examine an explicit setting as well where the personality trait is explicitly given to the model. To capture the semantic information of personality traits, we embed each personality trait as its word vector and incorporate this information in the captioning module:

$$\texttt{Emb}(w^l) \leftarrow \texttt{Emb}(w^l) + \texttt{Emb}(t_i) \quad \forall l = 1, \cdots, |\mathcal{S}_i| \tag{6.9}$$

where $\texttt{Emb}(\cdot)$ is an embedding layer and w^l is the input word at l-th position and t_i is the word (e.g. "optimistic") of the personality trait. Also, $\mathcal{S}_i$ denotes a caption, expressed in personality trait t_i. By explicitly providing the semantic information in the form of word vectors of personality traits, we can help the model to take advantage of the semantic relationship between different personality traits for training.

In this experiment, we seek to answer the question "*would it help the model perform better if it had access to explicit information about the personality traits?*". To answer this question, we compare the performance of our formulation with and without

using personality information. Other than using/ignoring personality information, other experimental settings remain the same. Table 6.3 summarizes this comparison. There is no significant improvement in the model performance using explicitly fed personality information. Based on this experiment, we believe that the captions, on their own, are adequately informative for the model to learn the new style and it may not need to use explicit information.

Number of adaptation samples: We performed the experiments reported in Table 6.1 using $K = 10$ samples as our adaptation set. The number of adaptation samples used for each task has a direct and significant effect on the overall performance of the model. To quantify this effect, we repeat the experiments for various sizes of adaptation sets. As expected and shown in Table 6.4, reducing the number of adaptation samples from 10 to 5 hinders the performance, however, the significant decrease in performance occurs when K is set to 1. That is, only a single sample per task is used for fast adaptation. The best result obtained when using 10 labeled samples for adaptation. This is still a very small number when compared to a fully-supervised method where the number of samples needed per personality trait is far beyond 10 samples.

(a) 

Trait: "Bizarre"

PR: I would love to run here.

GT: It'd be so cool to run down that hill!

(b) 

Trait: "Complex"

PR: This man looks like he is rocking out to the rhythm.

GT: The rhythms are so intricate.

(c) 

Trait: "Tense"

PR: I would be scared to be in that mountains in winter.

GT: Is snow on those mountains an indication of winter to come?

(d) 

Trait: "Fickle"

PR: What a cute little girl

GT: That is either a very cute sweater, or a very ugly sweater!

Figure 6.3: Qualitative examples of our method. (a) and (b) are cases where our method was successful in describing the image using the given trait. (c) shows an example where our method has described the image partially correct. (d) depicts a case where our method failed to generate a correct caption. GT and PR are ground-truth and predicted captions, respectively. Best viewed in color.

Meta-training		Meta-testing	
Freethinking	High-spirited	Cultured	Fanciful
Mystical	Passive	Extreme	Gloomy
Businesslike	Absentminded	Neutral	Sentimental
Cold	Confident	Rigid	Discouraging
Brilliant	Coarse	Clever	Tough
Dramatic	Enigmatic	Confused	Grim
Malicious	Offhand	Airy	Open
Aggressive	Barbaric	Objective	Curious
Cute	Complex	Neurotic	Artificial
Uncreative	Ordinary	Excitable	Aloof
Patriotic	Wise	Rational	Rustic
Resentful	Contemplative	Witty	Cowardly
Energetic	Scornful	Extraordinary	Cynical
Youthful	Articulate	Idealistic	Eloquent
Devious	Reflective	Breezy	Boyish
Silly	Peaceful	Hostile	Sensitive
Frivolous	Narcissistic	Formal	Cerebral
Outrageous	Sweet	Casual	Vague
Conservative	Contradictory	Elegant	Respectful
Deep	Miserable	Quirky	Happy
Melancholic	Bizarre	Dull	Exciting
Fearful	Argumentative	Courageous	Solemn
Scholarly	Rowdy	Sophisticated	Dreamy
Dry	Morbid	Calm	Irrational
Fiery	Ridiculous	Spirited	Whimsical
Gentle	Arrogant	Boisterous	Playful
Vivacious	Escapist	Maternal	Anxious
Assertive	Profound	Stiff	Opinionated
Abrasive	Vacuous	Moody	Fatalistic
Warm	Foolish	Intense	Honest
Fanatical	Blunt	Obnoxious	Old-fashioned
Unrealistic	Wishful	Cheerful	Mellow
Kind	Sympathetic	Money-minded	Creative
Colorful	Haughty	Passionate	Contemptible

Meta-training		Meta-testing	
Stylish	Imaginative	Impersonal	Disturbing
Romantic	Stoic	Fun-loving	Grand
Extravagant	Hateful	Fawning	Envious
Glamorous	Amusing	Unimaginative	Attractive
Conceited	Egocentric	Adventurous	Meticulous
Obsessive	Tense	Monstrous	Erratic
Destructive	Overimaginative	Angry	Pretentious
Charming	Apathetic	Shy	Bland
Critical	Bewildered	Nihilistic	Logical
Zany	Empathetic	Practical	Spontaneous
Paranoid	Humble	Serious	Caring
Suave	Considerate	Relaxed	Humorous
Captivating	Crazy	Artful	Irritable
Skeptical	Compassionate	Knowledgeable	Cruel
Earnest	Appreciative	Frightening	Daring
Pompous	Observant	Insightful	Lazy
Realistic	Fickle	Enthusiastic	Perceptive
Childish	Sarcastic	Emotional	Stupid
Optimistic	Odd	Questioning	Simple
Intelligent	Sensual	Provocative	

Table 6.5: Personality traits used in meta-training and meta-testing.

6.5 Conclusion

In this chapter, we tackle the challenging problem of few-shot caption generation for specific personality traits. We establish strong baselines and a data processing protocol to use in a few-shot setting. Through empirical studies, we show the challenging nature of this problem and propose a solution that can outperform the strong baselines in a variety of settings. Through ablation studies, we also further study the challenges associated with this challenging problem which shed light on future research direction.

Chapter 7

Conclusion and Future Work

In this thesis, we have introduced several works for a variety of vision-language problems aiming to increase the interactivity of computer vision systems for users. We tackle the problem of composed query image retrieval where the query consists of a reference image and a modification text in Chapter 3. We argue that the key element for such a system to perform better is to establish an explicit relationship between different modalities. Our cross-modal and self-attention modules connect each word in the modification text with each region of the input image. This design enables the model to precisely ground the desired modifications in local regions of the image. By formulating the cross-modal relationship explicitly, we are able to more precisely apply the desired modifications to the retrieved images.

In Chapter 4, we propose and tackle the challenging task of future video captioning. In this setting, the goal is to describe the most likely event to follow, given a limited video stream of what is happening now. Since predicting the future frame pixel values are almost impossible due to spatial and temporal variation, we pro-

pose to predict the semantic features (i.e. convolutional features) of the next frames, instead. These predicted features combined with features of the current event as context features and are input to a convolutional captioning module to generate the description.

We explore the problem of describing the change between a pair of images in Chapter 5. We present a novel training approach that ties the training of our primary task (i.e. image change captioning) with the training of the auxiliary task of composed query image retrieval. In the first part of our proposed training, the primary network generates a caption describing the change between a pair of input images, while the auxiliary network uses the generated caption and one of the images as input and retrieves the second image. In the second part, these networks switch places. The co-training of both tasks in a pseudo cycle consistency fashion forces the primary network to generate more informative captions.

Inspired by the recent advances in meta-learning, we propose a few-shot personalized image captioning task. Given that every user has a unique personality trait, our proposed method generates image captions that are tailored to the user's personality trait. Collecting a training dataset to train such a network in a fully supervised manner is a non-trivial task. Therefore, we design the few-shot setting. During training, the model learns to quickly adapt the parameters to generate image captions in the style of a specific personality trait, given only a few pairs of annotated samples. Therefore, in testing, the model utilizes a few annotated samples of a new personality trait and performs a few gradient descent steps for quick adaption to the new personality trait.

This thesis tackles several problems at the intersection of computer vision and natural language processing including composed query image retrieval, future video captioning, image change captioning, and few-shot personalized image captioning. Moving forward, we would like to further explore the following related directions:

- Linguistic commands are arguably the most natural form of interaction between humans and intelligent systems. The textual modifications as used in preceding chapters for image retrieval in a powerful tool in other applications and real-world production systems. For example, image manipulation or image editing can benefit leverage linguistic modifications from users to alter and/or edit specific parts of images. One can use textual commands for personalized video highlight creation. For instance, generating a recap for a certain part of a TV series or movie according to a given command by user: "*Show me a recap of action scenes for the past season.*"

- Embodied AI [23; 122; 34; 95] is a general term coined for agents that interact with a virtual or physical environment. Executing linguistic commands in an environment is a popular research area for embodied AI. Due to its real-life applications in-home robots and personal assistants, this field is on the verge to extend beyond laboratory research and enter the product level. A personal assistant or home robot usually ought to work in a visually complex environment. By using embodied AI, these agents can learn to take linguistic commands from the user and perform complex tasks (e.g. "*go to the second bedroom and put the book on the desk*"). A deep understanding of linguistics is a cornerstone in being able to perform the given tasks.

Bibliography

[1] Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] Sandra Aigner and Marco Körner. Futuregan: Anticipating the future frames of video sequences using spatio-temporal 3d convolutions in progressively growing autoencoder gans. *arXiv preprint arXiv:1810.01325*, 2018.

[3] Malihe Alikhani, Piyush Sharma, Shengjie Li, Radu Soricut, and Matthew Stone. Cross-modal coherence modeling for caption generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6525–6535, Online, July 2020. Association for Computational Linguistics.

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

[5] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[6] Jyoti Aneja, Aditya Deshpande, and Alexander G Schwing. Convolutional image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5561–5570. IEEE, 2018.

[7] Jyoti Aneja, Aditya Deshpande, and Alexander G Schwing. Convolutional image captioning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[9] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. *arXiv preprint arXiv:1810.09502*, 2018.

[10] Sébastien M R Arnold, Praateek Mahajan, Debajyoti Datta, Ian Bunner, and Konstantinos Saitas Zarkias. learn2learn: A library for Meta-Learning research. Aug. 2020.

[11] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72. Association for Computational Linguistics, 2005.

[12] Sreyasee Das Bhattacharjee, Junsong Yuan, Weixiang Hong, and Xiang Ruan. Query adaptive instance search using object sketches. In *ACM International Conference on Multimedia (ACMMM)*, 2016.

[13] Marc Bolaños, Álvaro Peris, Francisco Casacuberta, Sergi Soler, and Petia Radeva. Egocentric video description based on temporally-linked sequences.

Journal of Visual Communication and Image Representation, 50:205–216, 2018. Elsevier.

[14] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970. IEEE, 2015.

[15] Yue Cao, Mingsheng Long, Bin Liu, and Jianmin Wang. Deep cauchy hashing for hamming space retrieval. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[16] Yue Cao, Mingsheng Long, Jianmin Wang, Han Zhu, and Qingfu Wen. Deep quantization network for efficient image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

[17] Pew Research Center. Disabled americans are less likely to use technology. https://www.pewresearch.org/fact-tank/2017/04/07/disabled-americans-are-less-likely-to-use-technology/. Accessed: 2021-07-27.

[18] Tianlang Chen, Zhongping Zhang, Quanzeng You, Chen Fang, Zhaowen Wang, Hailin Jin, and Jiebo Luo. "factual"or"emotional": Stylized image captioning with adaptive learning and attention. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 519–535, 2018.

[19] Yanbei Chen and Loris Bazzani. Learning joint visual semantic matching embeddings for language-guided retrieval. In *European Conference on Computer Vision (ECCV)*, 2020.

[20] Sumit Chopra, Raia Hadsell, Yann LeCun, et al. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conference on*

Computer Vision and Pattern Recognition (CVPR), 2005.

[21] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[22] Tech Crunch. Facebook and instagram's ai-generated image captions now offer far more details. https://techcrunch.com/2021/01/19/facebook-and-instagrams-ai-generated-image-captions-now-offer-far-more-details/. Accessed: 2021-07-27.

[23] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, et al. Robothor: An open simulation-to-real embodied ai platform. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3164–3174, 2020.

[24] Riccardo Del Chiaro, Bartłomiej Twardowski, Andrew Bagdanov, and Joost van de Weijer. Ratt: Recurrent attention to transient tasks for continual image captioning. *Advances in Neural Information Processing Systems*, 2020.

[25] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[26] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2019.

[27] Eric Dodds, Jack Culpepper, Simao Herdade, Yang Zhang, and Kofi Boakye.

Modality-agnostic attention fusion for visual search with text feedback. *arXiv:2007.00145*, 2020.

[28] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[29] Xuguang Duan, Wenbing Huang, Chuang Gan, Jingdong Wang, Wenwu Zhu, and Junzhou Huang. Weakly supervised dense event captioning in videos. In *Advances in Neural Information Processing Systems*, 2018.

[30] Panna Felsen, Pulkit Agrawal, and Jitendra Malik. What will happen next? forecasting player moves in sports videos. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 3342–3351. IEEE, 2017.

[31] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135, 2017.

[32] Yang Fu, Yunchao Wei, Yuqian Zhou, Honghui Shi, Gao Huang, Xinchao Wang, Zhiqiang Yao, and Thomas Huang. Horizontal pyramid matching for person re-identification. In *AAAI Conference on Artificial Intelligence*, 2019.

[33] Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. Stylenet: Generating attractive visual captions with styles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3137–3146, 2017.

[34] Chuang Gan, Yiwei Zhang, Jiajun Wu, Boqing Gong, and Joshua B Tenenbaum. Look, listen, and act: Towards audio-visual embodied navigation. In

2020 IEEE International Conference on Robotics and Automation (ICRA), pages 9701–9707. IEEE, 2020.

[35] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[36] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[37] Lionel Gueguen and Raffay Hamid. Large-scale damage detection using satellite imagery. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[38] Longteng Guo, Jing Liu, Peng Yao, Jiangwei Li, and Hanqing Lu. Mscap: Multi-style image captioning with unpaired stylized text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4204–4213, 2019.

[39] Yangyang Guo, Zhiyong Cheng, Liqiang Nie, Yinglong Wang, Jun Ma, and Mohan Kankanhalli. Attentive long short-term preference modeling for personalized product search. *ACM Transactions on Information Systems (TOIS)*, 37(2):19, 2019.

[40] Xintong Han, Zuxuan Wu, Phoenix X Huang, Xiao Zhang, Menglong Zhu, Yuan Li, Yang Zhao, and Larry S Davis. Automatic spatially-aware fashion concept discovery. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[41] Jeff Hawkins and Sandra Blakeslee. *On intelligence: How a new understanding*

of the brain will lead to the creation of truly intelligent machines. Macmillan, 2007.

[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. corr abs/1512.03385 (2015), 2015.

[43] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. MIT Press.

[44] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, 2015.

[45] Sina Honari, Pavlo Molchanov, Stephen Tyree, Pascal Vincent, Christopher Pal, and Jan Kautz. Improving landmark localization with semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1546–1555, 2018.

[46] Mehrdad Hosseinzadeh and Yang Wang. Composed query image retrieval using locally bounded features. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[47] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[48] Phillip Isola, Joseph J Lim, and Edward H Adelson. Discovering states and transformations in image collections. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

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

[50] Junzhong Ji, Zhuoran Du, and Xiaodan Zhang. Divergent-convergent attention

for image captioning. *Pattern Recognition*, page 107928, 2021.

[51] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013. IEEE.

[52] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[53] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137. IEEE, 2015.

[54] Salman H Khan, Xuming He, Fatih Porikli, and Mohammed Bennamoun. Forest change detection in incomplete satellite images with deep neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(9):5407–5423, 2017.

[55] Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. Re-evaluating automatic metrics for image captioning. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 199–209, 2017.

[56] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[57] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 706–715. IEEE, 2017.

[58] Guang Li, Linchao Zhu, Ping Liu, and Yi Yang. Entangled transformer for im-

age captioning. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[59] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2285–2294, 2018.

[60] Zhuowan Li, Quan Tran, Long Mai, Zhe Lin, and Alan L Yuille. Context-aware group captioning via self-attention and contrastive features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3440–3450, 2020.

[61] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[62] Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 605–612. Association for Computational Linguistics, 2004.

[63] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

[64] Bin Liu, Yue Cao, Mingsheng Long, Jianmin Wang, and Jingdong Wang. Deep triplet quantization. In *ACM International Conference on Multimedia (ACMMM)*, 2018.

[65] Shikun Liu, Andrew Davison, and Edward Johns. Self-supervised generalisation with meta auxiliary learning. In *Advances in Neural Information Processing Systems*, 2019.

[66] Zhunga Liu, Gang Li, Gregoire Mercier, You He, and Quan Pan. Change detection in heterogenous remote sensing images via homogeneous pixel transformation. *IEEE Transactions on Image Processing*, 27(4):1822–1834, 2017.

[67] J Long, E Shelhamer, T Darrell, and UC Berkeley. Fully convolutional networks for semantic segmentation. arxiv 2014. *arXiv preprint arXiv:1411.4038*, 2014.

[68] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition (CVPR)*, pages 375–383. IEEE, 2017.

[69] Pauline Luc, Camille Couprie, Yann Lecun, and Jakob Verbeek. Predicting future instance segmentation by forecasting convolutional features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 584–599. Springer, 2018.

[70] Pauline Luc, Natalia Neverova, Camille Couprie, Jakob Verbeek, and Yann LeCun. Predicting deeper into the future of semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 648–657. IEEE, 2017.

[71] Moz. How to rank in google image search. https://moz.com/blog/seo-photos-visuals-graphics-whiteboard-friday. Accessed: 2021-07-27.

[72] Tushar Nagarajan and Kristen Grauman. Attributes as operators: factorizing unseen attribute-object compositions. In *European Conference on Computer Vision (ECCV)*, 2018.

[73] Pinterest Newsroom. Celebrating one year of pinterest lens. https://newsroom.pinterest.com/en/post/celebrating-one-year-of-pinterest-

lens. Accessed: 2021-07-27.

[74] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[75] Hyeonwoo Noh, Paul Hongsuck Seo, and Bohyung Han. Image question answering using convolutional neural network with dynamic parameter prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 30–38, 2016.

[76] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in neural information processing systems*, pages 3235–3246, 2018.

[77] Ariyo Oluwasanmi, Muhammad Umar Aftab, Eatedal Alabdulkreem, Bulbula Kumeda, Edward Y Baagyere, and Zhiquang Qin. Captionnet: Automatic end-to-end siamese difference captioning model with attention. *IEEE Access*, 7:106773–106783, 2019.

[78] Ariyo Oluwasanmi, Enoch Frimpong, Muhammad Umar Aftab, Edward Y Baagyere, Zhiguang Qin, and Kifayat Ullah. Fully convolutional captionnet: Siamese difference captioning attention model. *IEEE Access*, 7:175929–175939, 2019.

[79] Yingwei Pan, Ting Yao, Yehao Li, and Tao Mei. X-linear attention networks for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10971–10980, 2020.

[80] Yanwei Pang, Yazhao Li, Jianbing Shen, and Ling Shao. Towards bridging semantic gap to improve semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4230–4239, 2019.

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

[82] Dong Huk Park, Trevor Darrell, and Anna Rohrbach. Robust change captioning. In *IEEE International Conference on Computer Vision*, 2019.

[83] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. In *British Machine Vision Conference (BMVC)*, 2015.

[84] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

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

[86] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics, 2014.

[87] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence*, 2018.

[88] Álvaro Peris, Marc Bolaños, Petia Radeva, and Francisco Casacuberta. Video description using bidirectional recurrent neural networks. In *Proceedings of the*

International Conference on Artificial Neural Networks, pages 3–11. Springer, 2016.

[89] Filip Radenovic, Giorgos Tolias, and Ondrej Chum. Deep shape matching. In *European Conference on Computer Vision (ECCV)*, 2018.

[90] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[91] Alexander Riegler. The role of anticipation in cognition. In *Proceedings of the AIP Conference*, volume 573, pages 534–541. American Institute of Physics, 2001.

[92] Ken Sakurada and Takayuki Okatani. Change detection from a street image pair using cnn features and superpixel segmentation. In *British Machine Vision Conference*, 2015.

[93] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 4967–4976, 2017.

[94] Nikolaos Sarafianos, Xiang Xu, and Ioannis A Kakadiaris. Adversarial representation learning for text-to-image matching. In *IEEE International Conference on Computer Vision (CVPR)*, 2019.

[95] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347,

2019.

[96] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[97] Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-consistency for robust visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[98] Rishab Sharma and Anirudha Vishvakarma. Retrieving similar e-commerce images using deep learning. *arXiv:1901.03546*, 2019.

[99] Baifeng Shi, Judy Hoffman, Kate Saenko, Trevor Darrell, and Huijuan Xu. Auxiliary task reweighting for minimum-data learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[100] Xiangxi Shi, Xu Yang, Jiuxiang Gu, Shafiq Joty, and Jianfei Cai. Finding it at another side: A viewpoint-adapted matching encoder for change captioning. 2020.

[101] Kurt Shuster, Samuel Humeau, Hexiang Hu, Antoine Bordes, and Jason Weston. Engaging image captioning via personality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12516–12526, 2019.

[102] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[103] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2015.

[104] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[105] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. 2019.

[106] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

[107] Andrew Tao, Karan Sapra, and Bryan Catanzaro. Hierarchical multi-scale attention for semantic segmentation. *arXiv preprint arXiv:2005.10821*, 2020.

[108] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.

[109] Jiaojiao Tian, Shiyong Cui, and Peter Reinartz. Building change detection based on satellite stereo imagery and digital surface models. *IEEE Transactions on Geoscience and Remote Sensing*, 52(1):406–417, 2013.

[110] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[111] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), pages 4566–4575. IEEE, 2015.

[112] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4534–4542. IEEE, 2015.

[113] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1494–1504. Association for Computational Linguistics, 2015.

[114] The Verge. Microsoft's new image-captioning ai will help accessibility in word, outlook, and beyond. https://www.theverge.com/2020/10/14/21514405/image-captioning-seeing-ai-microsoft-algorithm-word-powerpoint-outlook. Accessed: 2021-07-27.

[115] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[116] ViSense. How visual search has transformed the modern shopping experience. https://www.visenze.com/blog/how-visual-search-has-transformed-the-modern-shopping-experience/. Accessed: 2021-07-27.

[117] Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. Composing text and image for image retrieval-an empirical odyssey. In

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[118] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[119] Junbo Wang, Wei Wang, Liang Wang, Zhiyong Wang, David Dagan Feng, and Tieniu Tan. Learning visual relationship and context-aware attention for image captioning. *Pattern Recognition*, 98:107075, 2020.

[120] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[121] Spencer Whitehead, Heng Ji, Mohit Bansal, Shih-Fu Chang, and Clare Voss. Incorporating background knowledge into video description generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3992–4001. Association for Computational Linguistics, 2018.

[122] Erik Wijmans, Samyak Datta, Oleksandr Maksymets, Abhishek Das, Georgia Gkioxari, Stefan Lee, Irfan Essa, Devi Parikh, and Dhruv Batra. Embodied question answering in photorealistic environments with point cloud perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6659–6668, 2019.

[123] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human

and machine translation. *arXiv:1609.08144*, 2016.

[124] Jing Xu, Rui Zhao, Feng Zhu, Huaming Wang, and Wanli Ouyang. Attention-aware compositional network for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[125] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning (ICML)*, 2015.

[126] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4507–4515. IEEE, 2015.

[127] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[128] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4584–4593. IEEE, 2016.

[129] Massimo Zanetti and Lorenzo Bruzzone. A generalized statistical model for binary change detection in multispectral images. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3378–3381, 2016.

[130] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings*

of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 93–104. Association for Computational Linguistics, 2018.

[131] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020.

[132] Ke Zhang, Yurong Guo, Xinsheng Wang, Jinsha Yuan, and Qiaolin Ding. Multiple feature reweight densenet for image classification. *IEEE Access*, 7:9872–9880, 2019.

[133] Xiao Zhang, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[134] Yu Zhang, Xinyu Shi, Siya Mi, and Xu Yang. Image captioning with transformer and knowledge graph. *Pattern Recognition Letters*, 143:43–49, 2021.

[135] Qi Zheng, Chaoyue Wang, and Dacheng Tao. Syntax-aware action targeting for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[136] Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. Grounded video description. *arXiv preprint arXiv:1812.06587*, 2018.

[137] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13041–13049, 2020.

[138] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong.

End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8739–8748. IEEE, 2018.

[139] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[140] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.