Composed Query Image Retrieval Using Locally Bounded Features

Mehrdad Hosseinzadeh and Yang Wang
University of Manitoba, Canada
{mehrdad, ywang}@cs.umanitoba.ca

Abstract

Composed query image retrieval is a new problem where the query consists of an image together with a requested modification expressed via a textual sentence. The goal is then to retrieve the images that are generally similar to the query image, but differ according to the requested modification. Previous methods usually consider the image as a whole. In this paper, we propose a novel method that represents the image using a set of local areas in the image. The relationship between each word in the modification text and each area in the image is then explicitly established, allowing the model to accurately correlate the modification text to parts of the image. We conduct extensive experiments on three benchmark datasets. The results show that our method outperforms other state-of-the-art approaches by a considerable margin.

1. Introduction

Image retrieval [29, 19] 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 [39, 17, 9] and product search [11, 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 paper, we consider the composed query image retrieval problem first introduced in [35]. 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 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. 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 [35, 29] 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 [35] that represents an image as a whole, our

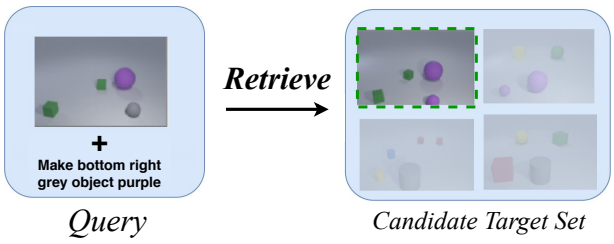

Figure 1: Overview of the composed query image retrieval. The query consists of a reference image and a sentence describing the changes one wants to be applied for retrieved images. The output is a set of images that are most similar to the query with the requested changes.

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.

2. Related Work

In this section, we review previous work in two lines of related research, namely image retrieval and multi-modal learning.

2.1. Image Retrieval

While traditional image retrieval approaches use manually designed features from the image [7], most modern image retrieval approaches use some form of deep learning [36, 13]. Depending on the type of queries, image retrieval falls into a number of 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 [22, 28, 41]. 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 [6] aims to find more optimal hash codes for images by putting a constraint on the quantization error. Deep Cauchy hashing [5] 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 [4, 24], 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. [37] 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. [35] propose the first work on using this type of composed queries for image retrieval.

2.2. Multi-modal Learning

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) [2, 30, 3, 32], image captioning [34, 40]. 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. [40] introduce the notion of attention in the visual domain with application to image captioning. Recently, self-attention [33] has been popular in many computer vision tasks [32, 31, 27]. We also use variations of self-attention in our work to capture a richer representation of images and the modification text.

3. 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. 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 2 depicts the overall architecture of our method.

3.1. 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 [35, 13]. 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 paper, 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 [25] 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 (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:

$$p^i = Linear\Big([N(x^i), N(y^i), N(w^i), N(h^i)]\Big) \quad (1)$$

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 $V(\mathcal{I})$:

$$c^i = Linear(avg(e^i, p^i)) \quad (2)$$

$$\mathcal{C}_1 = \{c^1, c^2, ..., c^K\} \quad (3)$$

$$V(\mathcal{I}) = VEP(\mathcal{C}_1) \quad (4)$$

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 C_{l-1} as the input, then applies a self attention (scaled dot-product attention) [33, 8] and passes it through a linear layer to generate the input for the next layer:

$$\mathcal{C}_{l+1} = Linear(SA(\mathcal{C}_l)), \text{where} \quad (5)$$

$$SA(C_l) = Softmax\left(\frac{\mathcal{C}_l {\mathcal{C}_l}^T}{\sqrt{d_v}}\right)\mathcal{C}_l \quad (6)$$

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:

$$g(\mathcal{I}) = Pool(V(\mathcal{I})) \quad (7)$$

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. 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 [38, 8] 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:

$$\mathcal{W}_{l+1} = Linear(SA(\mathcal{W}_l)), \text{where} \quad (8)$$

$$SA(\mathcal{W}_l) = Softmax\left(\frac{\mathcal{W}_l {\mathcal{W}_l}^T}{\sqrt{d_w}}\right)\mathcal{W}_l \quad (9)$$

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

3.3. 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 [35] 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.

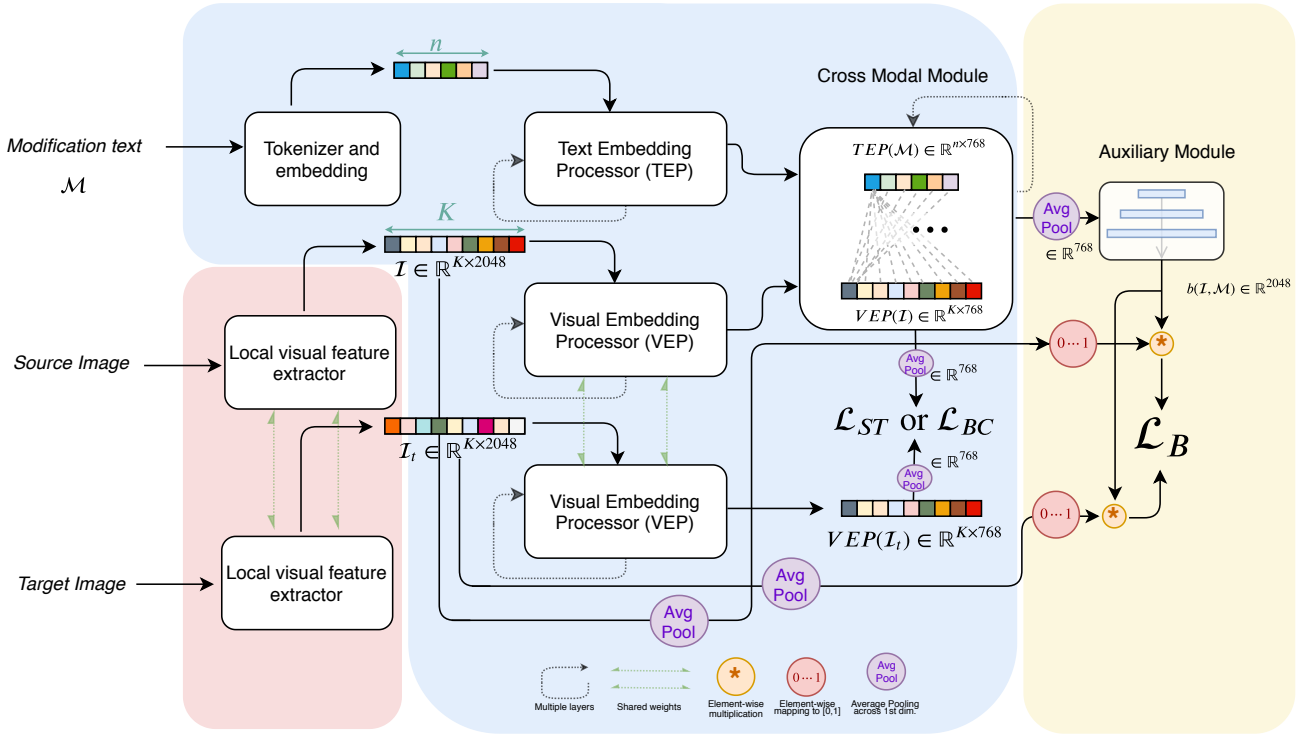

Figure 2: Overview of the proposed method. Light blue area (middle) is the main network, light pink (bottom left) is the pretrained visual feature extractor, and light yellow (right) is the auxiliary module, helping the main network to learn a better representation of query and target image by imposing 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. 3.5). Best viewed in color.

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. 3.1) 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). More specifically in the l-th layer ($l = 0, 1, ..., L-1$) of this module, the linguistically attended visual features are computed as follows:

$$\widehat{V}_l = CA\Big(V_l(\mathcal{I}), T_l(\mathcal{M})\Big), \text{where} \tag{10}$$

$$CA\Big(V_l(\mathcal{I}), T_l(\mathcal{M})\Big) = Softmax\left(\frac{V_l(\mathcal{I})T_l(\mathcal{M})^T}{\sqrt{d_v}}\right)T_l(\mathcal{M}) \tag{11}$$

$$V_{l+1} = Linear\Big(SA\big(\widehat{V}_l\big)\Big), \text{where} \tag{12}$$

$$SA\big(\widehat{V}_l\big) = Softmax\left(\frac{\widehat{V}_l\widehat{V}_l^T}{\sqrt{d_v}}\right)\widehat{V}_l \tag{13}$$

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:

$$\widehat{T}_l = CA\Big(T_l(\mathcal{M}), V_l(\mathcal{I})\Big), \text{where} \tag{14}$$

$$CA\Big(T_l(\mathcal{M}), V_l(\mathcal{I})\Big) = Softmax\left(\frac{T_l(\mathcal{I})V_l(\mathcal{M})^T}{\sqrt{d_w}}\right)V_l(\mathcal{M}) \tag{15}$$

$$T_{l+1} = Linear\Big(SA\big(\widehat{T}_l\big)\Big), \text{where} \tag{16}$$

$$SA\big(\widehat{T}_l\big) = Softmax\left(\frac{\widehat{T}_l\widehat{T}_l^T}{\sqrt{d_w}}\right)\widehat{T}_l \tag{17}$$

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

$$f(\mathcal{I}, \mathcal{M}) = Pool(V_L) \tag{18}$$

where $f(\mathcal{I}, \mathcal{M}) \in \mathbb{R}^{d_v}$.

3.4. 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:

$$sim(f(\mathcal{I}, \mathcal{M}), g(\mathcal{I}_t)) \gg sim(f(\mathcal{I}, \mathcal{M}), g(\mathcal{I}_{c_i})) \tag{19}$$

where $i = 1, 2, \cdots, k-1$. Here $sim(\cdot, \cdot)$ can be any similarity function. Similar to [35], we use the dot product as the similarity function $sim(\cdot, \cdot)$.

Following [35], 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:

$$\mathcal{L}_{ST} = \sum_{i=1}^{k-1} \log\left(1 + \frac{\exp(sim(f(\mathcal{I}, \mathcal{M}), g(\mathcal{I}_t)))}{\exp(sim(f(\mathcal{I}, \mathcal{M}), g(\mathcal{I}_{c_i})))}\right) \tag{20}$$

$\mathcal{L}_{ST}$ 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:

$$\mathcal{L}_{BC} = \frac{1}{|B|}\sum_{i=1}^{|B|} -\log\left(\frac{\exp\big(sim(f(\mathcal{I}_i, \mathcal{M}_i), g(\mathcal{I}_{t_i}))\big)}{\sum_{j=1}^{k-1}\exp\big(sim(f(\mathcal{I}_i, \mathcal{M}_i), g(\mathcal{I}_{c_j}))\big)}\right) \tag{21}$$

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

3.5. Auxiliary Module

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

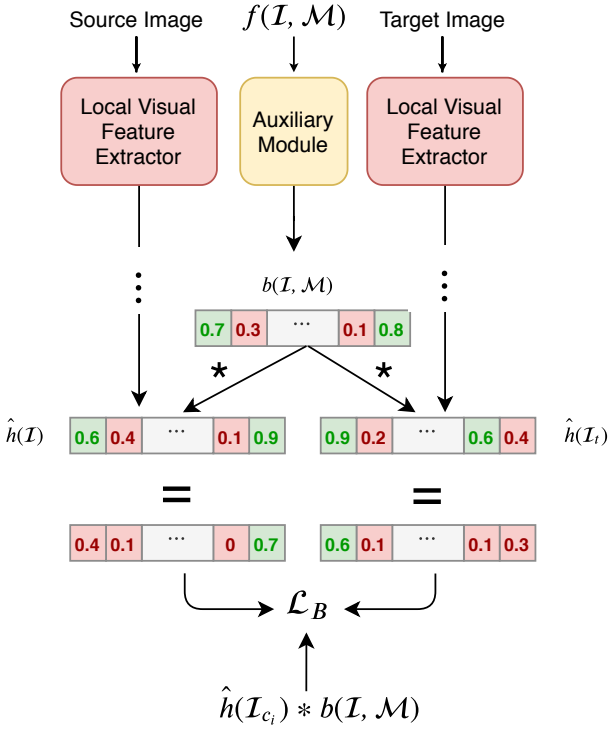

Figure 3: During training auxiliary module predicts a weight vector $b(\mathcal{I}, \mathcal{M})$ representing the importance of each feature in the vector representation of query and target images, given the requested modifications according to Eq. 23. During inference, this module can be activated to reject the distant candidates for the given query at an early stage (see Sec. 3.5).

Given an image $\mathcal{I}$, after extracting region features using a region proposal network (Sec. 3.1), the image can be represented as a 2-d representation in $\mathbb{R}^{K \times d_p}$. We then calculate a compact vector representation of the image via average pooling over K entities (regions), yielding $h(\mathcal{I}) \in \mathbb{R}^{d_v}$ 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:

$$\hat{h}(\mathcal{I}) = Sg\big(h(\mathcal{I}) + 1\big)/2 \tag{22}$$

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. 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 $b(\mathcal{I}, \mathcal{M}) \in \mathbb{R}^{d_v}$, 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 mod-

ule as:

$$\mathcal{L}_B = L_2(\hat{h}(\mathcal{I}) * b(\mathcal{I},\mathcal{M}), \hat{h}(\mathcal{I}_t) * b(\mathcal{I},\mathcal{M})) - L_2(\hat{h}(\mathcal{I}) * b(\mathcal{I},\mathcal{M}), \hat{h}(\mathcal{I}_{c_i}) * b(\mathcal{I},\mathcal{M})) + 1 \quad (23)$$

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.

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 exhibits how our auxiliary module works in training.

4. Experimental Setup and Results

We compare the performance of the proposed method with state-of-the-art approaches on three benchmark datasets: Fashion200K [12], MIT States [14], and CSS [35]. Following [35], 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 [35], 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 [35], FiLM [23], Relationship [26], Parameter Hashing [21], Show and Tell [34], Attribute as Operator [20] and the method of [12]. We use the pre-trained model provided by [2] with a Faster-RCNN [25] backbone for extracting the visual feature from each region proposal. This pre-model is trained on the MSCOCO dataset [10, 18] 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

Method	*Recall@*		
	K=1	*K=10*	*K=50*
Baselines			
Image only [35]	3.5	22.7	43.7
Text only [35]	1.0	12.3	21.8
Concat [35]	$11.9^{\pm 1.0}$	$39.7^{\pm 1.0}$	$62.6^{\pm 0.7}$
SOTA			
Han et al. [12]	6.3	19.9	38.3
Show and Tell [34]	$12.3^{\pm 1.1}$	$40.2^{\pm 1.7}$	$61.8^{\pm 0.9}$
Param. Hash. [21]	$12.2^{\pm 1.1}$	$40.0^{\pm 1.1}$	$61.7^{\pm 0.8}$
Relationship [26]	$13.0^{\pm 0.6}$	$40.5^{\pm 0.7}$	$62.4^{\pm 0.6}$
FiLM [23]	$12.9^{\pm 0.7}$	$39.5^{\pm 2.1}$	$61.9^{\pm 1.9}$
TIRG [35]	$14.1^{\pm 0.6}$	$42.5^{\pm 0.7}$	$63.8^{\pm 0.8}$
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 1: Results on the Fashion200K dataset. The numbers of other approaches are adopted from [35]. 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 $Recall@1$.

Adam optimizer with linear-decayed learning rate [32] (LR = $1e-5$) and Adam optimizer [16] 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. 5.

4.1. Results on Fashion200K

The Fashion200K dataset [12] 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 [35, 12] 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.

Results on this dataset are shown in Table 1. Our proposed method outperforms other approaches in all the metrics with a remarkable 26% performance improvement over TIRG [35] 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 image. This allows our method to more effectively capture the rela-

Method	*Recall@*		
	K=1	*K=5*	*K=10*
Baselines			
Image only [35]	$3.3^{\pm 0.1}$	$12.8^{\pm 0.2}$	$20.9^{\pm 0.1}$
Text only [35]	$7.4^{\pm 0.4}$	$21.5^{\pm 0.9}$	$32.7^{\pm 0.8}$
Concat [35]	$11.8^{\pm 0.2}$	$30.8^{\pm 0.2}$	$42.1^{\pm 0.3}$
SOTA			
Show and Tell [34]	$11.9^{\pm 0.1}$	$31.0^{\pm 0.5}$	$42.0^{\pm 0.8}$
Attribute Op. [20]	$8.8^{\pm 0.1}$	$27.3^{\pm 0.3}$	$39.1^{\pm 0.3}$
Relationship [26]	$12.3^{\pm 0.5}$	$31.9^{\pm 0.7}$	$42.9^{\pm 0.9}$
FiLM [23]	$10.1^{\pm 0.3}$	$27.7^{\pm 0.7}$	$42.9^{\pm 0.9}$
TIRG [35]	$12.2^{\pm 0.4}$	$31.9^{\pm 0.3}$	$41.3^{\pm 0.3}$
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 2: Results on the MIT States dataset. The numbers of other approaches are adopted from [35]. 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 $Recall@1$.

tionship between the modification text and each entity in the image. Moreover, we obtain noticeably better results when $K = 36$ ("*big*").

4.2. Results on MIT States dataset

The MITStates dataset [14] 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 [35], there are about 43K training samples and 80 nouns are used for training. The rest is kept for testing.

Table 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 [26] and TIRG [35] methods. Again, we observe that the results are better when using 36 region proposals ("*big*").

4.3. Results on CSS Dataset

The CSS dataset [35] 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 [15]. 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

Method	*Recall@*	
	$3D \rightarrow 3D$ *K=1*	$2D \rightarrow 3D$ *K=1*
Baselines		
Image only [35]	6.3	6.3
Text only [35]	0.1	0.1
Concat [35]	$60.6^{\pm 0.8}$	27.3
SOTA		
Show & Tell [34]	$33.0^{\pm 3.2}$	6.0
Param Hash. [21]	$60.5^{\pm 1.9}$	31.4
Relation. [26]	$62.1^{\pm 1.2}$	30.6
FiLM [23]	$65.6^{\pm 0.5}$	43.7
TIRG [35]	$73.7^{\pm 1.0}$	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: Results on the CSS dataset. The numbers are adopted from [35]. $3D \rightarrow 3D$ 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.

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 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 shows the result of experiment on this dataset. Consistent with other experiments, our 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*").

5. 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. 3.5, 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 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 addi-

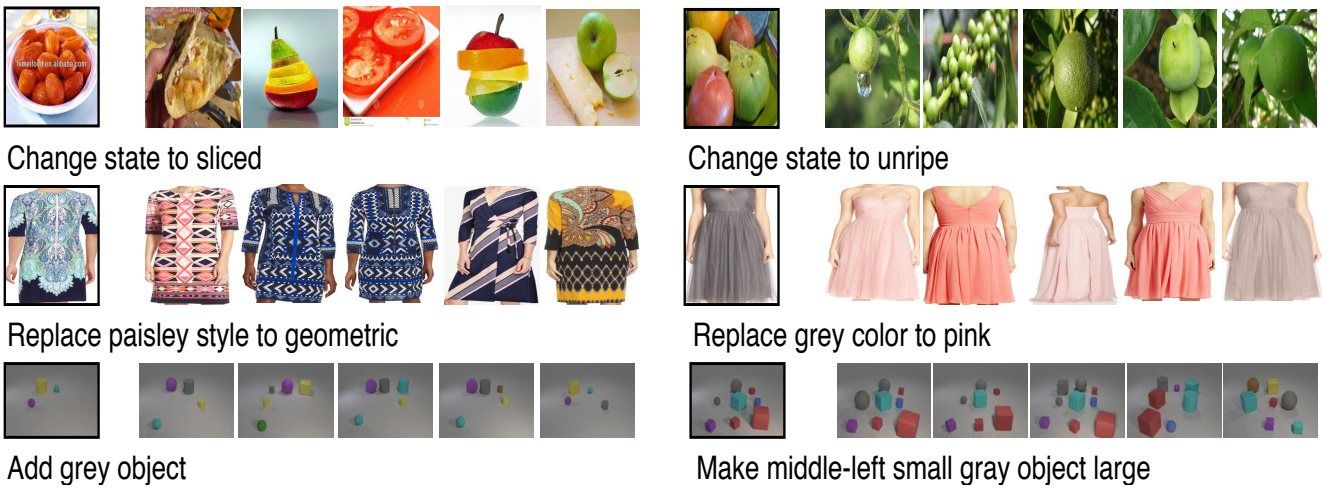

Figure 4: Some qualitative examples of our method. Each row shows the query image, the modification text, and retrieved images. The examples are from Fashion200K, MITStates, and CSS3D datasets, respectively.

Variation	*Recall@* K=1	*Recall@* K=5	*ACRR*
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 (θ = 100)	**75.92**	**91.03**	**75.24**%
Ours (θ = 85)	70.81	84.07	91.86%

Table 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.

tional 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):

$$ACRR = 1 - \frac{\sum_{i=1}^{|TQ|} \mathbb{1}\Big(Dist\big(\mathcal{X}(\mathcal{I}), \mathcal{X}(\mathcal{I}_{c_i})\big) < \theta\Big)}{|TQ| \times |CI|} \quad (24)$$

$$s.t. \quad \mathcal{X}(\mathcal{I}) = \hat{h}(\mathcal{I}) * b(\mathcal{I}, \mathcal{M}) \quad (25)$$

$$\mathcal{X}(\mathcal{I}_{c_i}) = \hat{h}(\mathcal{I}_{c_i}) * b(\mathcal{I}, \mathcal{M}) \quad (26)$$

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 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. 4 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.

6. 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.

Acknowledgement

This work was supported by NSERC. We thank NVIDIA for donating some of the GPUs used in this work.

References

[1] Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1096–1104, 2016. 1

[2] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6077–6086, 2018. 2, 6

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 2425–2433, 2015. 2

[4] Sreyasee Das Bhattacharjee, Junsong Yuan, Weixiang Hong, and Xiang Ruan. Query adaptive instance search using object sketches. In *Proceedings of the ACM international conference on Multimedia (ACMMM)*, pages 1306–1315. ACM, 2016. 2

[5] 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)*, June 2018. 2

[6] 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. 2

[7] Sumit Chopra, Raia Hadsell, Yann LeCun, et al. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 539–546, 2005. 2

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. 3

[9] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8295–8302, 2019. 1

[10] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6904–6913, 2017. 6

[11] 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. 1

[12] 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 *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1463–1471, 2017. 6

[13] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015. 2, 3

[14] Phillip Isola, Joseph J Lim, and Edward H Adelson. Discovering states and transformations in image collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1383–1391, 2015. 6, 7

[15] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2901–2910, 2017. 7

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[17] 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. 1

[18] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755, 2014. 6

[19] Bin Liu, Yue Cao, Mingsheng Long, Jianmin Wang, and Jingdong Wang. Deep triplet quantization. In *Proceedings of the ACM International Conference on Multimedia (ACMMM)*, 2018. 1

[20] Tushar Nagarajan and Kristen Grauman. Attributes as operators: factorizing unseen attribute-object compositions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 169–185, 2018. 6, 7

[21] 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. 6, 7

[22] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. In *British Machine Vision Conference*, 2015. 2

[23] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 6, 7

[24] Filip Radenovic, Giorgos Tolias, and Ondrej Chum. Deep shape matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 751–767, 2018. 2

[25] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 91–99, 2015. 3, 6

[26] 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. 6, 7

[27] Nikolaos Sarafianos, Xiang Xu, and Ioannis A Kakadiaris. Adversarial representation learning for text-to-image matching. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 5814–5824, 2019. 2

[28] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015. 2

[29] Rishab Sharma and Anirudha Vishvakarma. Retrieving similar e-commerce images using deep learning. *arXiv preprint arXiv:1901.03546*, 2019. 1

[30] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8317–8326, 2019. 2

[31] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. *arXiv preprint arXiv:1904.01766*, 2019. 2

[32] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5103–5114, 2019. 2, 6

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017. 2, 3

[34] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2015. 2, 6, 7

[35] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6439–6448, 2019. 1, 2, 3, 5, 6, 7

[36] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[37] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5005–5013, 2016. 2

[38] 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 preprint arXiv:1609.08144*, 2016. 3

[39] Jing Xu, Rui Zhao, Feng Zhu, Huaming Wang, and Wanli Ouyang. Attention-aware compositional network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2119–2128, 2018. 1

[40] 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 *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2048–2057, 2015. 2

[41] Xiao Zhang, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10823–10832, 2019. 2