

Multi-task Learning of Hierarchical Vision-Language Representation

Duy-Kien Nguyen¹ and Takayuki Okatani^{1,2}

¹Graduate School of Information Sciences, Tohoku University ²RIKEN Center for AIP

{kien, okatani}@vision.is.tohoku.ac.jp

Abstract

It is still challenging to build an AI system that can perform tasks that involve vision and language at human level. So far, researchers have singled out individual tasks separately, for each of which they have designed networks and trained them on its dedicated datasets. Although this approach has seen a certain degree of success, it comes with difficulties of understanding relations among different tasks and transferring the knowledge learned for a task to others. We propose a multi-task learning approach that enables to learn vision-language representation that is shared by many tasks from their diverse datasets. The representation is hierarchical, and prediction for each task is computed from the representation at its corresponding level of the hierarchy. We show through experiments that our method consistently outperforms previous single-task-learning methods on image caption retrieval, visual question answering, and visual grounding. We also analyze the learned hierarchical representation by visualizing attention maps generated in our network.

1. Introduction

Since the recent successes of deep learning on single modality tasks, multi-modal tasks lying on the intersection of vision and language, such as image captioning [20, 38], visual question answering (VQA) [13, 3], visual grounding [29] etc., have attracted increasing attention in the related fields. Despite the fact that each of these tasks has basically been studied independently of others, there must be close connections among them. For instance, each task may occupy a different level in a hierarchy of sub-tasks comprising the cognitive function associated with vision and language. To gain deeper understanding of such hidden relations among these vision-language tasks, we think that multi-task learning of these tasks will be a promising direction of research.

Although we have seen significant progresses in multi-task learning of unimodal tasks of vision [17, 30] or language [24, 1, 33] so far, there has been only a lim-

ited amount of progress in multi-task learning of vision-language tasks. This may be attributable to the diversity of these tasks. In addition to differences in inputs and outputs of the tasks, their level of complexity differs, too. Even though these tasks share some structures in common, it is unclear how to learn them in the framework of multi-task learning.

A solution to this difficulty is to create and use a dataset designed for multi-task learning, where multiple objectives are given to identical inputs. Indeed, recent studies follow this approach, in which they have gained early successes by joint training of different vision-language tasks using multiple objectives, such as answer and question generation for VQA [18], and caption and scene graph generation for image captioning [19]. However, this approach cannot be employed when such dedicated datasets are not available. It may be impossible to create such datasets for arbitrary combinations of vision-language tasks. To do this, we need to limit the range of tasks, which makes it hard for the learned results to generalize to other tasks or datasets.

In this paper, aiming to resolve these issues, we propose a framework for joint learning of multiple vision-language tasks. Our goal is to enable to learn vision-language representation that is shared by many tasks from their diverse data sources. To make this possible, we employ *Dense Co-attention* layers, which were developed for VQA and shown to perform competitively with existing methods [27]. Using a stack of Dense Co-attention layers, we can gradually update the visual and linguistic features at each layer, in which their fine-grained interaction is considered at the level of individual image regions and words. We utilize this property to learn hierarchical vision-language representation such that each individual task takes the learned representation at a different level of the hierarchy corresponding to its complexity. We design a network consisting of an encoder for computing shared hierarchical representation and multiple task-specific decoders for making prediction from the representation; see Fig. 1. This design enables multi-task learning from diverse data sources; to be rigorous, we train the same network alternately on each task/dataset based on a scheduling algorithm.

We evaluate this method on three vision-language tasks, image caption retrieval, visual question answering, and visual grounding, using popular datasets, Flickr30K captions [38], MS-COCO captions [20], VQA 2.0 [13], and Flickr30K-Entities [29]. The results show that our method outperforms previous ones that are trained on individual tasks and datasets. We also visualize the internal behaviours of the task-specific decoders to analyze effects of joint learning of the multiple tasks.

2. Related Work

Vision-language representation learning Recently, studies of multi-modal tasks of vision and language have made significant progress, such as image captioning [2, 23], visual question answering [27, 34], visual grounding [37, 29], image caption retrieval [26, 15], and visual dialog [7]. In the last few years, researchers have demonstrated the effectiveness of learning representations shared by the two modalities in a supervised fashion. However, these studies deal with a single task at a time.

Transfer learning A basic method of transfer learning in deep learning is to train a neural network for a source task and use it in some ways for a target task. This method has been successful in a wide range of problems in computer vision and natural language processing. For multi-modal vision-language problems, early works explored similar approaches that used pretrained models trained on some source tasks. Plummer et al. [29] proposed to use a pretrained network trained on a visual grounding task to enrich the shared representational space of images and captions, improving accuracy of image caption retrieval. Lin et al. [21] proposed to use pretrained models of VQA and CQA (caption Q&A); they compute answer predictions for multiple questions and then treat them as features of an image and a caption, computing relevance between them. In this study, instead of transferring knowledge from a source task to a target task in a single direction (e.g., via pretrained models), we consider a multi-task learning framework in which learning multiple tasks will be mutually beneficial to each individual task. This is made possible by the proposed network and its training methodology; it suffices only to train our network for individual tasks with their loss functions and supervised data.

Multi-task learning of vision-language tasks Since its introduction [5], multi-task learning has achieved many successes in several areas including computer vision and natural language processing. However, there have been only a few works that explored joint learning of multiple multi-modal tasks of vision and language. Li et al. [19] proposed a method for learning relations between multiple regions

in the image by jointly refining the features of three different semantic tasks, scene graph generation, object detection, and image/region captioning. Li et al. [18] showed that joint training on VQA and VQG (visual question generation) contributes to improve VQA accuracy and also understanding of interactions among images, questions, and answers. Although these works have demonstrated the potential of multi-task learning for the vision-language tasks, they strongly rely on the availability of the datasets providing supervision over multiple tasks, where an input is shared by all the tasks while a different label is given to it for each task.

3. Learning Vision-Language Interaction

3.1. Problem Formulation

We consider multiple vision-language tasks, in each of which an output O is to be estimated from an input pair of I and S , where I is an image and S is a sentence. The input pair I and S have the same formats for all the tasks (with differences in the interpretation of S for different tasks), whereas the output O will naturally be different for each task. For example, in VQA, O is a set of confident scores of answers to the input question S in a predefined answer set; in image caption retrieval, O is a set of binary values indicating the relevance of the input caption S ; in visual grounding, O is a set of binary variables specifying a set of image regions corresponding to the phrases in the input sentence S .

The input image is represented by a set of region features, which we denote by $I = [i_1, \dots, i_T]$; in our experiments, we use a bag of region features from a pretrained Faster-RCNN [34]. The input sentence is represented by a sequence $S = [s_1, \dots, s_N]$ of word features, which are obtained by first computing GloVe embedding vectors [28] of the input words and then inputting them to a two-layer bidirectional LSTM.

An overview of the proposed network architecture is shown in Fig. 1. It consists of a single encoder shared by all the tasks and multiple task-specific decoders. We will describe these two components below.

3.2. Shared Encoder

To construct the shared encoder, we employ the *Dense Co-attention* layer [27]. We conjecture that different tasks require different levels of vision-language fusion. Thus, we stack multiple Dense Co-attention layers to extract hierarchical, fused features of the input image I and sentence S . We attach a decoder for each task to the layer that is the best fit for the task in terms of the fusion hierarchy, as shown in Fig. 1.

Starting with $S_0 = S$ and $I_0 = I$, the shared encoder incrementally updates the language and vision features at

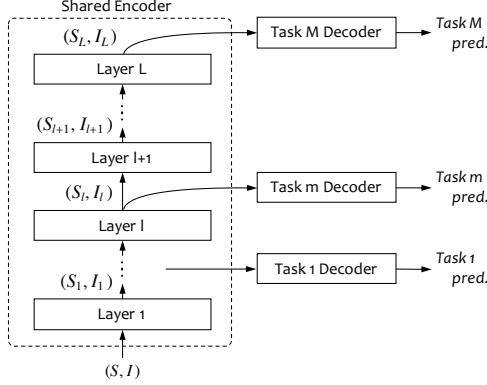


Figure 1: The proposed network consists of a shared encoder and task-specific decoders. The shared encoder is a stack of L Dense Co-attention layers and computes hierarchical representation of the input sentence and image. Each of M task-specific decoders receives one intermediate-layer representation to compute prediction for its task.

each Dense Co-attention layer as

$$(S_l, I_l) = \text{DCL}_l(S_{l-1}, I_{l-1}), \quad (1)$$

where $S_l = [s_{l,1}, \dots, s_{l,N}] \in \mathbb{R}^{d \times N}$, $I_l = [i_{l,1}, \dots, i_{l,T}] \in \mathbb{R}^{d \times T}$, $S_{l-1} = [s_{l-1,1}, \dots, s_{l-1,N}] \in \mathbb{R}^{d \times N}$, and $I_{l-1} = [i_{l-1,1}, \dots, i_{l-1,T}] \in \mathbb{R}^{d \times T}$; DCL_l indicates the input-output function realized by the l -th Dense Co-attention layer. In each Dense Co-attention layer, two attention maps are generated in a symmetric fashion, i.e., the one over image regions conditioned on each sentence word and the other over sentence words conditioned on each image region, where multiplicative attention is employed. The generated attention maps are then applied to I_{l-1} and S_{l-1} to yield \hat{I}_{l-1} and \hat{S}_{l-1} , respectively. Finally, the original and attended features of image and sentence are fused by first concatenating them and then applying a linear transform followed by ReLU. This is done for sentence feature and image feature, respectively, as

$$s_{l,n} = \text{ReLU} \left(W_{S_l} \begin{bmatrix} s_{l-1,n} \\ \hat{s}_{l-1,n} \end{bmatrix} + b_{S_l} \right) + s_{l-1,n}, \quad (2)$$

$$i_{l,t} = \text{ReLU} \left(W_{I_l} \begin{bmatrix} i_{l-1,t} \\ \hat{i}_{l-1,t} \end{bmatrix} + b_{I_l} \right) + i_{l-1,t}, \quad (3)$$

where $W_{S_l} \in \mathbb{R}^{d \times 2d}$, $b_{S_l} \in \mathbb{R}^d$, $W_{I_l} \in \mathbb{R}^{d \times 2d}$ and $b_{I_l} \in \mathbb{R}^d$ are learnable parameters.

3.3. Task-specific Decoders

As shown in Fig. 1, we design a task-specific decoder for each task and attach it to the layer of the shared encoder selected for the task. Letting l be the index of this layer, the decoder receives (S_l, I_l) and produces the final output O for

this task. We explain below its design for each of the three tasks considered in this study.

3.3.1 Image Caption Retrieval

In this task, we calculate the relevant score for the input pair (I, S) . The decoder for this task consists of two summary networks and a scoring layer. Let l_R be the index of the layer of the shared encoder to which this decoder is attached. The first summary network computes a vector $v_{I_{l_R}} \in \mathbb{R}^d$ that summarizes the image features $I_{l_R} = [i_{l_R,1}, \dots, i_{l_R,T}]$ of T regions. The second summary network computes $v_{S_{l_R}} \in \mathbb{R}^d$ that summarizes the sentence features $S_{l_R} = [s_{l_R,1}, \dots, s_{l_R,N}]$ of N words.

The two summary networks have the same architecture. Let us take the image summary network for explanation. It consists of a two-layer feedforward network that yields attention maps over the T image regions and a mechanism that applies the attention maps to I_{l_R} to obtain the summary vector $v_{I_{l_R}}$.

The feedforward network has d hidden units with ReLU non-linearity, which receives the image feature of a single region and outputs K scores. To be specific, denoting the feedforward network by MLP_I , it maps the feature vector of each region $t (= 1, \dots, T)$ to K scores as

$$c_t^I = [c_{1,t}^I, \dots, c_{K,t}^I] = \text{MLP}_I(i_{l_R,t}), \quad t = 1, \dots, T. \quad (4)$$

These scores are then normalized by softmax across the T regions to obtain K parallel attention maps over the T regions, which are averaged to produce the final attention map $[\alpha_1^I, \dots, \alpha_T^I]$; more specifically,

$$\alpha_t^I = \frac{1}{K} \sum_{k=1}^K \frac{\exp(c_{k,t}^I)}{\sum_{t=1}^T \exp(c_{k,t}^I)}, \quad t = 1, \dots, T. \quad (5)$$

The summary vector $v_{I_{l_R}}$ is the weighted sum of the image feature vectors using this attention weights, i.e.,

$$v_{I_{l_R}} = \sum_{t=1}^T \alpha_t^I i_{l_R,t}, \quad (6)$$

As mentioned above, we generate K parallel attention maps and average them to obtain a single attention map. This is to capture more diverse attention distribution.

We follow the same procedure to compute the summary vector $v_{S_{l_R}}$, where a two-layer feedforward network MLP_S generating K parallel attention maps over N word features $S_{l_R} = [s_{l_R,1}, \dots, s_{l_R,N}]$ is used. Using the two summary vectors $v_{I_{l_R}}$ and $v_{S_{l_R}} \in \mathbb{R}^d$ thus obtained, the scoring layer computes the relevant score of an image-caption pair (I, S) as

$$\text{score}(I, S) = \sigma(v_{I_{l_R}}^\top W v_{S_{l_R}}), \quad (7)$$

where σ is the logistic function and $W \in \mathbb{R}^{d \times d}$ is a learnable weight matrix.

3.3.2 Visual Question Answering

In this task, we compute the scores of a set of predefined answers for the input image-question pair. Let l_Q be the index of the layer to which the decoder is attached. We employ the same architectural design as in the decoder for image caption retrieval to compute the summary vectors $v_{I_{l_Q}}$ and $v_{S_{l_Q}}$ from the input features, i.e., $S_{l_Q} = [s_{l_Q,1}, \dots, s_{l_Q,N}]$ and $I_{l_Q} = [i_{l_Q,1}, \dots, i_{l_Q,T}]$. To obtain these summary vectors, two summary networks, each of which is two-layer feedforward network with d hidden units and ReLU nonlinearity, are used to compute K attention maps, and then they are applied to the input features.

Following [27], we compute scores for a set of the predefined answers by using a two-layer feedforward network having d hidden units with ReLU non-linearity and output units for the scores; the output units employ the logistic function for their activation function. Denoting the network by MLP, the scores are calculated as

$$(\text{scores of answers}) = \sigma \left(\text{MLP} \left(\begin{bmatrix} v_{I_{l_Q}} \\ v_{S_{l_Q}} \end{bmatrix} \right) \right). \quad (8)$$

3.3.3 Visual Grounding

This is a task in which given an image and a phrase (usually one contained in a caption describing the image), we want to identify the image region corresponding to the phrase. Previously proposed approaches attempt to learn to score each region-phrase pair separately; any context in the caption is not taken into account, or any joint inference about global interaction between all phrases in the caption is not performed. We believe that context is important for understanding a local phrase in a sentence, needless to mention its necessity for higher-level tasks.

Let l_G be the index of the layer of the shared encoder connecting to the decoder for this task. Given $P = [(b_1, e_1), \dots, (b_H, e_H)]$ where (b_h, e_h) indicates the start and end indexes of the h -th phrase in the input N word caption ($1 \leq b_h \leq e_h \leq N$), we compute the feature $p_h \in \mathbb{R}^d$ for the h -th phrase by pooling the word features in the index range of $[b_h : e_h]$ as

$$p_h = \text{AvgPooling}(S_{l_G}[b_h : e_h]). \quad (9)$$

Here we use average pooling to produce a fixed-size vector representation of a phrase p_h . This can also be seen as computing an attended feature using an attention map with equal weights on words in the phrase and zero weights on other words.

We then compute the score for a pair of a phrase $p_h \in \mathbb{R}^d$ and an image region $i_{l_G,t} \in \mathbb{R}^d$ as

$$\text{score}(p_h, i_t) = \sigma \left(p_h^\top W i_{l_G,t} \right), \quad (10)$$

where σ is the logistic function and $W \in \mathbb{R}^{d \times d}$ is a learnable weight matrix.

4. Training on Multiple Tasks

We train the proposed network on the above multiple tasks. Considering their diversity, we use a strategy to train it on a single selected task at a time and iterate this by switching between the tasks. (Note that we cannot simultaneously train the network on these tasks by minimizing the sum of their losses, because the inputs differ among tasks.)

4.1. Task-switching schedule

It is essential for which task and how many times we update the parameters of the network. In this study, we employ two strategies. One is a curriculum learning approach that starts from a single task and increases the number of tasks one by one, i.e., training first on single tasks, then on pairs of tasks, and finally on all tasks. The other is a scheduling method when training more than one task in this curriculum. To be specific, we employ the strategy of periodical task switching as in [8] but with different iterations of parameter updates for each task. Following [24], we update the network parameters for i -th task for $C\alpha_i$ iterations before switching to a new task, where C is the number of iterations in an updating cycle that we specify; α_i is determined as explained below. Algorithm 1 shows the entire procedure. More details are given in the supplementary material.

4.2. Choosing Layers Best Fit for Tasks

We need to decide which layer $l(i)$ of the shared encoder is the best fit for each task i . We pose it as a hyperparameter search, in which we also determine other parameters for training each task i , i.e., $\# \text{step}_i$ (step size for learning rate decay), the number of iteration $\# \text{iter}_i$ (used to determine α_i), and the batch size bs_i . To choose them, we conduct a grid search by training the network on each individual task. After that, these hyperparameters are used in joint learning of the tasks. Denoting the number of tasks to be learned by $M' (= 1, 2 \text{ or } 3)$, the step size of training is given by $\# \text{step} = \sum_{i=1}^{M'} \# \text{step}_i$; the total number of iterations is $\# \text{iter} = \sum_{i=1}^{M'} \# \text{iter}_i$; and α_i is determined as $\alpha_i = \# \text{iter}_i / \# \text{iter}$. The batch size bs_i and layer $l(i)$ determined as above are fixed in all the subsequent training processes.

5. Experiments

We conducted a series of experiments to test the effectiveness of the proposed approach.

Algorithm 1: Training the proposed network on M' tasks. \mathbf{E}_l represents a sub-network of the shared encoder up to l -th layer ($l = 1, \dots, L$); $\mathbf{D}_1, \dots, \mathbf{D}_{M'}$ are M' task-specific decoders; θ indicates their parameters. $l(i)$ is the index of the layer to which the decoder for i -th task is attached. We represent the output of this layer for an input \mathbf{x} as $\mathbf{E}_{l(i)}$.

```

1 num_cycle =  $\lfloor \frac{\# \text{iter}}{C} \rfloor$ 
2  $S = ([1] * C\alpha_1 + \dots + [M'] * C\alpha_{M'}) * \text{num\_cycle}$ 
3 # Array operation in Python style:
4  $[1] * 3 + [2] * 2 = [1, 1, 1, 2, 2]$ 
5 for task  $i$  in  $S$  do
6   1: Sample pairs of an input and output:  $\mathbf{x}, \mathbf{y} \sim \mathbb{P}_i$ 
7   2:  $\mathbf{h}_{i\mathbf{x}} \leftarrow \mathbf{E}_{l(i)}(\mathbf{x})$ 
8   3: Output prediction  $\hat{\mathbf{y}} \leftarrow \mathbf{D}_i(\mathbf{h}_{i\mathbf{x}})$ 
9   4:  $\theta \leftarrow \text{Adam}(\nabla_\theta L(\mathbf{y}, \hat{\mathbf{y}}))$ 
10 end

```

5.1. Datasets and Evaluation Methods

Image Caption Retrieval We use two datasets for this task, MS-COCO and Flickr30k. MS-COCO consists of 82,783 *train* and 40,504 *val* images. Following the standard procedure [16], we use the 1,000 *val* images and the 1,000 or 5,000 *test* images, which are selected from the original 40,504 *val* images. We use all of the 82,783 *train* images for training. Flickr30k consists of 31,783 images collected from Flickr. Following the standard procedure [16], we split them into *train*, *val*, and *test* sets; *val* and *test* contains 1,000 images for each and *train* contains all the others. We report Recall@ K ($K = 1, 5, 10$) (i.e., recall rates at the top 1, 5, and 10 results).

Visual Question Answering We use VQA 2.0 [13], which is the most popular and the largest (as of now) dataset for this task. It contains questions and answers for images of MS-COCO. There are 443,757 *train*, 214,354 *val*, and 447,793 *test* questions, respectively. The *train* questions are for *train* images of MS-COCO and *val* and *test* questions are for *val* and *test* images of MS-COCO respectively. Following the standard approach [34], we choose correct answers appearing more than 8 times to form the predefined answer pool. We use the accuracy metric presented in the original paper [3] in all the experiments.

Visual Grounding For visual grounding task, we evaluate our approach on Flickr30k Entities [29], which contains 244,035 annotations to the image-caption pairs (31,783 images and 158,915 captions) of Flickr30k. It provides correspondence between phrases in a sentence and boxes in an image that represent the same entities. The *train*, *val* and

test are splitted as in the ICR task. We use 1,000 images for *val* and *test* splits each and the rest for *train* split following [29]. The task is to localize the corresponding box(es) to each of the given phrases in a sentence. As proposed in [29], we consider a predicted region to be a correct match with a phrase if it has IOU ≥ 0.5 with the ground truth bounding box for that phrase. By treating the phrase as the query to retrieve the regions from the input image, we report Recall@ K ($K = 1, 5, 10$) similar to image caption retrieval (the percentage of queries for which a correct match has rank of at most K).

Avoiding Contamination of Training Samples As we train the network by alternately switching the tasks, we need to make sure that there is no contamination between training and testing sets for all the tasks. To make a fair comparison with previous studies of VQA, we need to train the network using both *train* and *val* questions of VQA 2.0, as was done in the previous studies. However, if we use *val* questions in our joint learning framework, our network (i.e., the shared encoder) can see the *val* set of MS-COCO, resulting in contamination of training samples. To avoid this, we use the following procedure: i) we first train the network using all the *train* sets for the three tasks and test it on the *test* sets for ICR and VG; ii) we then train the network (from scratch) using the *train* sets for ICR and VG and *train+val* sets for VQA and test it on the *test* sets for VQA. This procedure was employed in the experiments of Sec. 5.4, but not employed in the experiments of Sec. 5.3, because evaluation was done only on *val* sets for all the tasks.

5.2. Optimal Layers and Training Parameters

As explained in Sec. 4.2, we first train our network on each individual task to find the layers fit for each task along with other training parameters. The results are: $l_R = 3$ (image caption retrieval), $l_Q = 5$ (VQA), and $l_G = 2$ (visual grounding). The training parameters were determined accordingly; see the supplementary material for details. We freeze all these parameters throughout all the experiments.

We note here the training method used in all the experiments. We used the Adam optimizer with the parameters $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.99$, and α decay = 0.5. We employed a simple training schedule; we halve the learning rate by “ α decay” after each “# step” or step size, which are determined above. All the weights in our network were initialized by the method of Glorot et al. [12]. Dropout is applied with probability of 0.3 and 0.1 over FC layers and LSTM, respectively. The dimension d of the feature space is set to 1024.

5.3. Effects of Joint Learning of Multiple Tasks

To evaluate the effectiveness of joint learning, we first trained the model on all possible combinations out of the

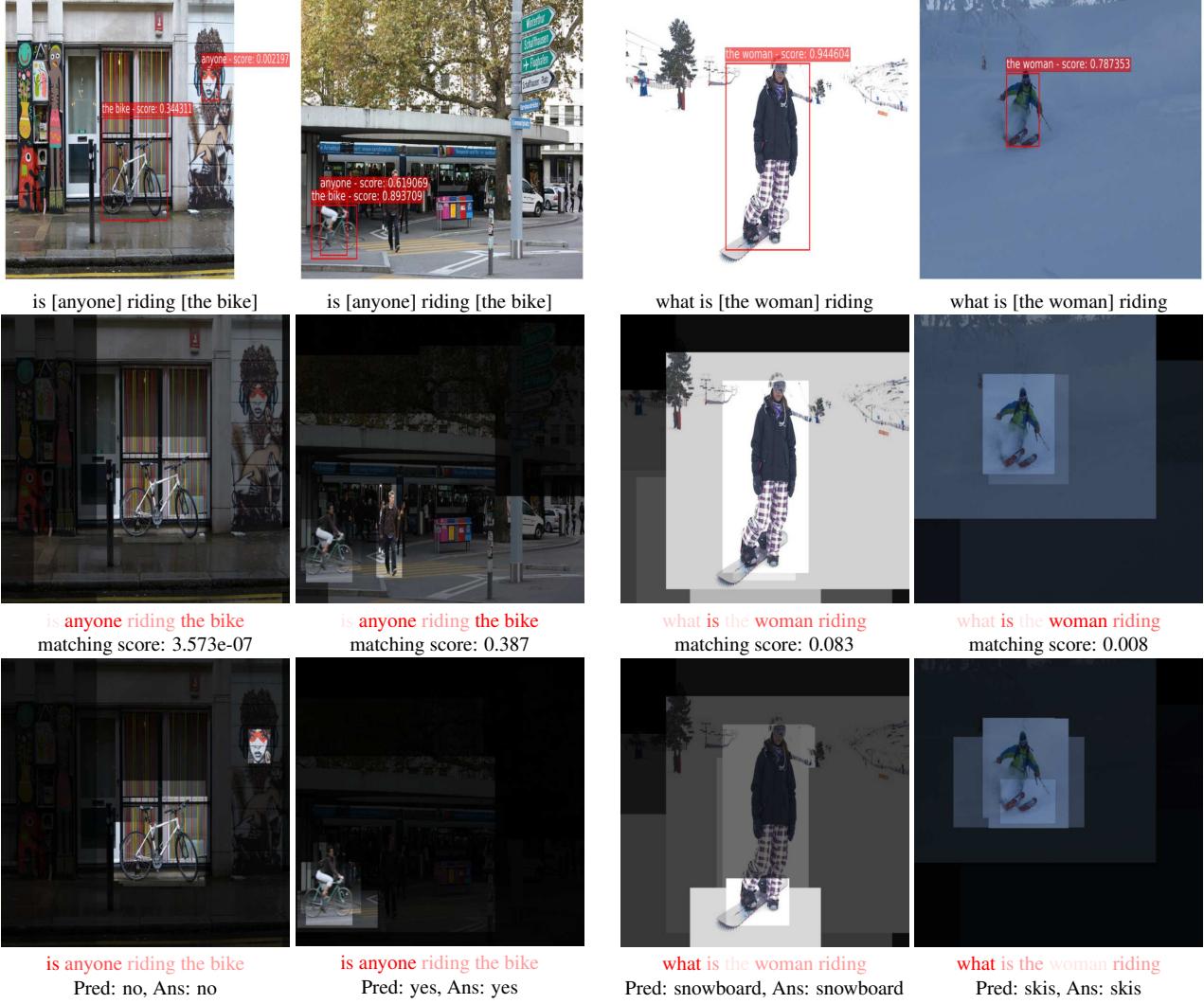


Figure 2: Example visualizations of behaviours of our network for two complementary image-question pairs (i.e., samples with the same question but different images and answers) from VQA 2.0 dataset. The three rows (from top to bottom) show the behaviours of the VG, ICR, and VQA decoders, respectively. For VG, top-1 regions corresponding to the entities (i.e., NP chunks) in the questions are shown. For ICR and VQA, the attention maps generated in their decoders are shown; the brightness of image pixels and the redness of words indicate the attention weights.

three tasks and evaluated their performances. To be specific, for each combination of tasks, we trained our model on their *train* split(s) and calculating its performance for each of the trained tasks on its *val* split. When training on two or more tasks, we used the method explained in Sec. 4.1.

Table 1 shows the results. It is observed that the joint learning of two tasks achieves better or comparable performances than the learning on a single task; and that the joint learning of all the three tasks yields the best performance. These confirm the effectiveness of our method for multi-task learning.

For ICR, we use two datasets, MS-COCO and Flickr30k;

the former is about three times larger than the latter. We evaluated how performances vary between when using the former and when using the latter. Table 2 shows the results. It is observed that the joint learning with VQA and VG is more beneficial for the smaller dataset (Flickr30k) than the larger one (MS-COCO), e.g., from 67.16 to 72.07 vs. from 69.05 to 70.43 (ICR: image annotation). On the other hand, the improvements of the other tasks (VQA and VG) due to the joint training with ICR are smaller for Flickr30k than for MS-COCO, e.g., from 65.50 to 66.09 vs. 65.50 to 66.35 (VQA).

Table 1: Performances for different combinations of the three tasks, VQA, VG(visual grounding), and ICR(image caption retrieval). Accuracy (Acc) is reported for VQA, and Recall@1 (R@1) is reported for VG and ICR; two numbers of ICR are image annotation (upper) and image retrieval (lower), respectively. MS-COCO dataset is used for ICR.

Task	VQA (Acc)	ICR (R@1)	VG (R@1)
VQA	65.50	-	-
ICR	-	69.05 56.47	-
VG	-	-	58.09
VQA + ICR	66.24	69.52 56.74	-
VQA + VG	65.85	-	58.07
ICR + VG	-	69.23 57.40	58.28
VQA + ICR + VG	66.35	70.43 57.50	58.26

Table 2: Effects of joint training when using different image caption retrieval datasets, MS-COCO and Flickr30k. *Single* means that each task is learned individually.

Task	Single (w/o ICR)	MS-COCO		Flickr30k	
		Single	+VQA+VG	Single	+VQA+VG
VQA (Acc)	65.50	-	66.35	-	66.09
ICR (R@1)	-	69.05 56.47	70.43 57.50	67.16 53.17	72.07 56.42
VG (R@1)	58.09	-	58.26	-	58.03

5.4. Full Results on Test Sets

We next show the performance of our method on *test* sets for the three tasks. We employ the procedure for avoiding training data contamination explained in the last paragraph of Sec. 5.1. We show below comparisons of our method with previous methods on each task. Note that our method alone performs joint learning of the three tasks and others are trained only on each individual task.

Image Caption Retrieval Table 3 shows the performances of previous methods and our method on Flickr30k and MS-COCO (The numbers for MS-COCO are performance on the 1,000 testing images. In the supplementary material, we report the performance on the 5,000 testing images of MS-COCO). It is seen that our method is comparable with the state-of-the-art method (S-E Model) on MS-COCO. For Flickr30k, which is three times smaller than MS-COCO, our method outperforms the best published result (S-E Model) by a large margin (about 9.5% in average) on all six evaluation criteria, showing the effectiveness of

our method. This demonstrates that our method can leverage the joint learning with other tasks to cover insufficient amount of training data for ICR.

Visual Question Answering Table 4 shows comparisons of our method to previous published results on VQA 2.0 in both test-dev and test-standard sets. It is observed in Table 4 that our method outperforms the state-of-the-art method (DCN [27]) by a noticeable margin of $\sim 0.7\%$ on the two test sets. It is noted that the improvements are seen in all the question types of test-standard set (*Other* with 0.5%, *Number* with 0.2%, and *Yes/No* with 0.9%). Notably, its accuracy for counting questions (*Number*) is on par with the Counting Module, which is designed to improve accuracy of this question type.

Visual Grounding Table 5 shows comparisons of our method with previous methods on the Flickr30k Entities dataset. Although our method shows lower performance than RTP [29] on the R@5 and R@10 evaluation metrics, it achieves a much better result on the hardest metric R@1. It should be noted that our method uses only phrase-box correspondences provided in the training dataset, and does not use any other information, such as box size, color, segmentation, or pose-estimation, which are used in previous studies [29, 37].

5.5. Qualitative Evaluation

To analyze effects of joint learning of multiple tasks, we visualize behaviours of our network. We use complementary image-question pairs contained in VQA 2.0 [13] for better analyses. Figure 2 shows two examples of such visualization, each for a complementary image-question pair. For visualization of VG, we extract NP chunks from the input question and treat them as entities. We then compute the score between each entity and all of the image regions, as described in Sec. 3.3.3. The first row of Fig. 2 shows the correspondences between a few entities found in the questions and their top-1 image regions. For ICR and VQA, we visualize attention maps generated in their decoders, which are shown in the second and third rows of Fig. 2.

It can be seen from the first row of Fig. 2 that the VG decoder correctly aligns each entity to its corresponding image region. From the second row of Fig. 2 (i.e., the attention maps of the ICR decoder) we can observe that the ICR decoder is looking at the same entities as those found in the VG decoder but with wider attention in the image and sentence, implying that not only the relevant entities but their relations are captured in the ICR decoder. It is then seen from the third row of Fig. 2 (i.e., the attention weights on image regions and question words of the VQA decoder) that it narrows down its attention on the image regions and question words that are relevant to properly answer the input

Table 3: Results of image annotation and retrieval on the Flickr30K and MSCOCO (1000 testing) datasets.

Method	Flickr30k dataset						MSCOCO dataset					
	Image Annotation			Image Retrieval			Image Annotation			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
VQA [21]	33.9	62.5	74.5	24.9	52.6	64.8	50.5	80.1	89.7	37.0	70.9	82.9
RTP [29]	37.4	63.1	74.3	26.0	56.0	69.3	-	-	-	-	-	-
DSPE [35]	40.3	68.9	79.9	29.7	60.1	72.1	50.1	79.7	89.2	39.6	75.2	86.9
sm-LSTM [14]	42.5	71.9	81.5	30.2	60.4	72.3	53.2	83.1	91.5	40.7	75.8	87.4
RRF [22]	47.6	77.4	87.1	35.4	68.3	79.9	56.4	85.3	91.5	43.9	78.1	88.6
2WayNet [9]	49.8	67.5	-	36.0	55.6	-	55.8	75.2	-	39.7	63.3	-
DAN [26]	55.0	81.8	89.0	39.4	69.2	79.1	-	-	-	-	-	-
VSE++ [10]	52.9	79.1	87.2	39.6	69.6	79.5	64.6	89.1	95.7	52.0	83.1	92.0
S-E Model [15]	55.5	82.0	89.3	41.1	70.5	80.1	69.9	92.9	97.5	56.7	87.5	94.8
Ours	71.6	84.6	90.8	56.1	82.9	89.4	70.2	89.2	95.9	57.4	88.4	95.6

Table 4: Results of the proposed method along with published results of others on VQA 2.0 with single model.

Method	Feature	Test-dev				Test-standard			
		Overall	Other	Number	Yes/No	Overall	Other	Number	Yes/No
MCB [11] reported in [13]	Resnet	-	-	-	-	62.27	53.36	38.28	78.82
MF-SIG-T3 [6]		64.73	55.55	42.99	81.29	-	-	-	-
Adelaide-Teney-MSR [34]		62.07	52.62	39.46	79.20	62.27	52.59	39.77	79.32
DCN [27]		66.72	56.77	46.65	83.70	67.04	56.95	47.19	83.85
Memory-Augmented Net [25]		-	-	-	-	62.10	52.60	39.50	79.20
VKMN [32]		-	-	-	-	64.36	57.79	37.90	83.70
Adelaide-Teney-MSR [34]	Faster RCNN	65.32	56.05	44.21	81.82	65.67	56.26	43.90	82.20
DCN [27] in our experiments		68.60	58.76	50.85	84.83	68.94	58.78	51.23	85.27
Counting Module [39]		68.09	58.97	51.62	83.14	68.41	59.11	51.39	83.56
MLB + DA-NTN [4]		67.56	57.92	47.14	84.29	67.94	58.20	47.13	84.60
Ours		69.28	59.17	51.54	85.80	69.57	59.27	51.46	86.17

Table 5: Comparison of our method and previous ones on the visual grounding task using Flickr30k Entities in the same condition.

Method	R@1	R@5	R@10
Structured Matching [36]	42.08	-	-
DSPE [35]	43.89	64.46	68.66
GroundeR [31]	48.38	-	-
MCB [11]	48.69	-	-
RTP [29]	50.89	71.09	75.73
GOP [37]	53.97	-	-
Ours	57.39	69.37	71.03

questions, e.g., the bikes in the images and the phrase “*is anyone*” in the questions; and the snowboard and the skis in the images and the phrase “*what is*”.

Other observations can be made for the results in Fig. 2. For instance, the ICR decoder gives a very low score (3.573×10^{-7}) for the pair of the first image and the question “*is anyone riding the bike*” and a high score (0.387) for the second image and the same question. Considering the word attention focusing only on the phrase “*anyone riding the bike*”, we may think that the ICR decoder correctly judges the (in)consistency between the contents of the im-

ages and the phrase. These agree well with their correct answers in VQA (i.e., “*No*” and “*Yes*”), implying the interaction between ICR and VQA. Further analyses will be provided in supplementary material.

6. Summary and Conclusion

In this paper, we have presented a multi-task learning framework for vision-language tasks. The key component is the proposed network consisting of the representation encoder that learns to fuse visual and linguistic representations in a hierarchical fashion, and task-specific decoders that utilize the learned representation at their corresponding levels in the hierarchy to make prediction. We have shown the effectiveness of our approach through a series of experiments on three major tasks and their datasets. The shared hierarchical representation learned by the encoder has been shown to generalize well across the tasks.

Acknowledgement

This work was partly supported by JSPS KAKENHI Grant Number JP15H05919 and JST CREST Grant Number JPMJCR14D1.

References

- [1] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. Multi-task learning for document ranking and query suggestion. In *International Conference on Learning Representations (ICLR)*, 2018. 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 *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2
- [3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015. 1, 5
- [4] Yalong Bai, Jianlong Fu, Tiejun Zhao, and Tao Mei. Deep attention neural tensor network for visual question answering. In *European Conference on Computer Vision (ECCV)*, 2018. 8
- [5] Rich Caruana. Multitask learning. *Machine Learning*, 1997. 2
- [6] Zhu Chen, Zhao Yanpeng, Huang Shuaiyi, Tu Kewei, and Ma Yi. Structured attentions for visual question answering. In *International Conference on Computer Vision (ICCV)*, 2017. 8
- [7] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M.F. Moura, Devi Parikh, and Dhruv Batra. Visual Dialog. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2
- [8] Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. Multi-task learning for multiple language translation. In *International Joint Conference on Natural Language Processing (IJCNLP)*, 2015. 4
- [9] Aviv Eisenshtat and Lior Wolf. Capturing deep correlations with 2-way nets. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 8
- [10] Fartash Faghri, David J. Fleet, Ryan Kiros, and Sanja Fidler. VSE++: improved visual-semantic embeddings. *arXiv preprint arXiv:1707.05612*, 2017. 8
- [11] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2016. 8
- [12] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2010. 5
- [13] 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 *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 5, 7, 8
- [14] Yan Huang, Wei Wang, and Liang Wang. Instance-aware image and sentence matching with selective multimodal LSTM. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 8
- [15] Yan Huang, Qi Wu, Chunfeng Song, and Liang Wang. Learning semantic concepts and order for image and sentence matching. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 8
- [16] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 5
- [17] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1
- [18] Yikang Li, Nan Duan, Bolei Zhou, Xiao Chu, Wanli Ouyang, Xiaogang Wang, and Ming Zhou. Visual question generation as dual task of visual question answering. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2
- [19] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2
- [20] 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. 1, 2
- [21] Xiao Lin and Devi Parikh. Leveraging visual question answering for image-caption ranking. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 8
- [22] Y. Liu, Y. Guo, E. M. Bakker, and M. S. Lew. Learning a recurrent residual fusion network for multimodal matching. In *International Conference on Computer Vision (ICCV)*, 2017. 8
- [23] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2
- [24] Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. In *International Conference on Learning Representations (ICLR)*, 2016. 1, 4
- [25] Chao Ma, Chunhua Shen, Anthony R. Dick, and Anton van den Hengel. Visual question answering with memory-augmented networks. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 8
- [26] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. Dual attention networks for multimodal reasoning and matching. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 8
- [27] Duy-Kien Nguyen and Takayuki Okatani. Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 4, 7, 8
- [28] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2014. 2

- [29] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *International Journal of Computer Vision (IJCV)*, 2017. 1, 2, 5, 7, 8
- [30] Jamie Ray, Heng Wang, Du Tran, Yufei Wang, Matt Feiszli, Lorenzo Torresani, and Manohar Paluri. Scenes-objects-actions: A multi-task, multi-label video dataset. In *European Conference on Computer Vision (ECCV)*, 2018. 1
- [31] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision (ECCV)*, 2016. 8
- [32] Zhou Su, Chen Zhu, Yinpeng Dong, Dongqi Cai, Yurong Chen, and Jianguo Li. Learning visual knowledge memory networks for visual question answering. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 8
- [33] Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J. Pal. Learning general purpose distributed sentence representations via large scale multi-task learning. In *International Conference on Learning Representations (ICLR)*, 2018. 1
- [34] Damien Teney, Peter Anderson, Xiaodong He, and Anton van den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 5, 8
- [35] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 8
- [36] Mingzhe Wang, Mahmoud Azab, Noriyuki Kojima, Rada Mihalcea, and Jia Deng. Structured matching for phrase localization. In *European Conference on Computer Vision (ECCV)*, 2016. 8
- [37] Raymond Yeh, Jinjun Xiong, Wen-Mei Hwu, Minh Do, and Alexander Schwinger. Interpretable and globally optimal prediction for textual grounding using image concepts. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 2, 7, 8
- [38] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In *Transactions of the Association for Computational Linguistics (TACL)*, 2014. 1, 2
- [39] Yan Zhang, Jonathon Hare, and Adam Prgej-Bennett. Learning to count objects in natural images for visual question answering. In *International Conference on Learning Representations (ICLR)*, 2018. 8

Supplementary Material for “Multi-task Learning of Hierarchical Vision-Language Representation”

Duy-Kien Nguyen¹ and Takayuki Okatani^{1,2}

¹Graduate School of Information Sciences, Tohoku University

²RIKEN Center for Advanced Intelligence Project
`{kien, okatani}@vision.is.tohoku.ac.jp`

This document contains the following: i) more details of setup of the experiments reported in the main paper (Sec.[A](#)); ii) additional results of the selection of optimal layers for the task-specific decoders (Sec.[B](#)); iii) additional results of ICR on MSCOCO 5,000 testing images (Sec.[C](#)); and iv) more visualization of the proposed network for a variety of images including failure cases of VQA and ICR (Sec.[D](#) and [E](#)).

A. More Details of the Experimental Setup

In all the experiments reported in this study, images and sentences (i.e., questions or captions) were preprocessed as follows. We used Faster-RCNN [9] to extract the bottom-up features from each image, which yields from 10 to 100 features (also referred to as regions in this study), i.e., $T \in [10, 100]$. Questions and captions were tokenized using Python Natural Language Toolkit (nltk) [2]. We used the vocabulary provided by the CommonCrawl-840B GloVe model for English word vectors [8], and set out-of-vocabulary words to *unk*.

As is mentioned in Sec.4.2 of the main paper, we conducted a hyper-parameter search on several training parameters including the layers used for the task-specific decoders by training the network on each individual task. The parameters thus determined are shown in Table 1.

We provide below additional details of the training procedures of the three tasks. We used the cross-entropy loss for all the three tasks.

Image Caption Retrieval This task consists of two subtasks; one is to retrieve relevant images given a query caption (image retrieval) and the other is to retrieve relevant captions given a query image (image annotation). In the training, given pairs of image-caption (I, C)’s, where I and C are an image and a caption, respectively, we compute the losses for the two subtasks for each pair as follows. For image retrieval, we randomly sample $F - 1$ images that are different from I , and compute the loss for F images including I , in which the label for the ground truth image (I) is set to 1 and those for the others are all 0.

Table 1: Hyperparameters determined by training on individual tasks and then used for joint training (# step: step size of learning rate decay, # iter: total of training iterations, K=1,000 units).

Task	Level	# step	# iter	Batch size	Cycle (C)
VQA	5	8K	20K	400	1
ICR	3	12K	30K	64	1
VG	2	4K	10K	64	1
VQA + VG	5, 2	12K	30K	400, 64	3
ICR + VG	3, 2	16K	40K	64, 64	4
VQA + ICR	5, 3	20K	50K	400, 64	5
VQA + ICR + VG	5, 3, 2	24K	60K	400, 64, 64	6

For image annotation, we randomly sample $F - 1$ captions that are not the ground truth captions corresponding to the image (I) and compute the loss of image annotation for F captions including C , in which the label for the ground truth caption (C) is 1 and those for the others are 0. We minimize the sum of the two losses. We used $F = 26$ for all the experiments.

Visual Question Answering We followed the procedure of [9]. We treat VQA as a multi-label classification task, where each training question is associated with one or several answers with soft accuracy label(s) in $[0, 1]$. Multiple answers appear in the case of disagreement among human annotators. The scores of answers are computed as in the original paper [1], that is,

$$\text{score}_i = \min\left(\frac{\# \text{ humans that provide the } i\text{-th answer}}{3}, 1\right)$$

where score_i is the score of i -th answer in the predefined answer set.

Visual Grounding The dataset provides a set of samples, each of which is built upon a pair of an image and its caption. Each sample consists of a set of phrases in the caption and the corresponding box(es) in the image. We label each phrase with its corresponding box(es) as 1 and with other boxes as 0. These boxes are obtained in the aforementioned pre-processing using the pre-trained Faster-RCNN. The loss is the sum over all possible phrase-region pairs in the image and caption.

B. Additional Results of Layer Selection for the Three Tasks

As mentioned in Sec.5.2 of the main paper, we train our network on each individual task to choose the layer of the shared encoder fit for each task. Table 2 shows the results. Based on these, we determined $l_R = 3$ (image caption retrieval), $l_Q = 5$ (VQA), and $l_G = 2$ (visual grounding), as reported in the main paper. However, while it is simple and can be performed efficiently, this method may not provide optimal choice of layers for the three tasks, as it does not consider interactions among the three tasks.

Table 2: Performance of the proposed network trained and tested on the same individual task. These are used to determine the layer of the shared encoder for each of the three tasks.

Task \ Layer	1	2	3	4	5	6
VQA	64.72	65.21	65.34	65.35	65.50	65.27
ICR	56.45	56.78	57.15	54.18	-	-
VG	46.30	46.00	48.05	42.64	-	-
	57.74	58.09	57.80	-	-	-

Thus, we also tested another method for choosing the layers that is based on joint-training of the three tasks. Table 3 shows the results, which were obtained by the following procedure. Initially, we determine the order of the three tasks in terms of the level in the hierarchy of the shared representation. Based on the above results, we determine their (descending) order as follows: VQA, ICR, and VG. We first determine the optimal layer for VQA by training the network on VQA alone, which is the same as the first row (VQA) of Table 2; this results in $l_Q = 5$. Next, we determine the optimal layer for ICR. To do this, we train the network on VQA+ICR for different choice of the layer for ICR (i.e., $l_R = 1, \dots$) while fixing the layer for VQA (i.e., $l_Q = 5$). We evaluate the performance for different l_R 's on VQA, ICR (image annotation) and ICR (image retrieval). The second to forth rows of Table 3 show the performance on VQA, image annotation, and image retrieval, respectively. From this, we choose $l_R = 3$. Finally, we determine the layer for VG. To do this, we train the network on VQA+ICR+VG for different layer l_G ($= 1, 2, 3$) for VG. As above, we evaluate the performance on VQA, image annotation, and image retrieval, and VG, which are shown in the fifth to eighth rows of Table 3, respectively. From this, we choose $l_G = 2$.

In short, we obtain the same results as the first method based on individual task training. This confirms the validity of our choice of the layers for the three tasks. In the above experiments, we set $F = 16$ in image caption retrieval for efficient computation; the reduction of F contributes the most to reducing necessary computational resource.

Table 3: Performance of the proposed network trained and tested on several combinations of tasks. Their combinations are created in a cumulative fashion, assuming the order of the three tasks to be VQA, ICR, and VG in terms of level in the representaion hierarchy in the shared encoder. ICR1 and ICR2 indicate image annotation and image retrieval, respectively.

Task \ Layer		1	2	3	4	5	6
Trained/tested on VQA alone	(VQA)	64.72	65.21	65.34	65.35	65.50	65.27
Trained/tested on VQA+ICR(2 subtasks) for different layers for ICR and layer = 5 for VQA	(VQA)	65.80	66.00	66.09	65.90	-	-
	(ICR1)	58.01	58.63	59.25	56.37	-	-
	(ICR2)	47.91	48.70	49.03	47.27	-	-
Trained/tested on VQA+ICR(2)+VG for different layers for VG and layer = 5 for VQA and 3 for ICR	(VQA)	66.10	66.15	66.15	-	-	-
	(ICR1)	61.94	61.88	61.90	-	-	-
	(ICR2)	49.84	50.81	50.01	-	-	-
	(VG)	57.86	58.17	57.83	-	-	-

C. Comparisons on MS-COCO dataset of 5,000 testing images

As noted in the main paper, we conducted evaluation on MSCOCO 5,000 testing images. The results are shown in Table 4. Our method is comparable to the state-of-the-art method (S-E Model). It is noteworthy that our method provided only 50 mismatched pairs for each matched pair, while S-E Model provided 128.

Table 4: Results of image annotation and retrieval on the MSCOCO (5,000 testing) datasets.

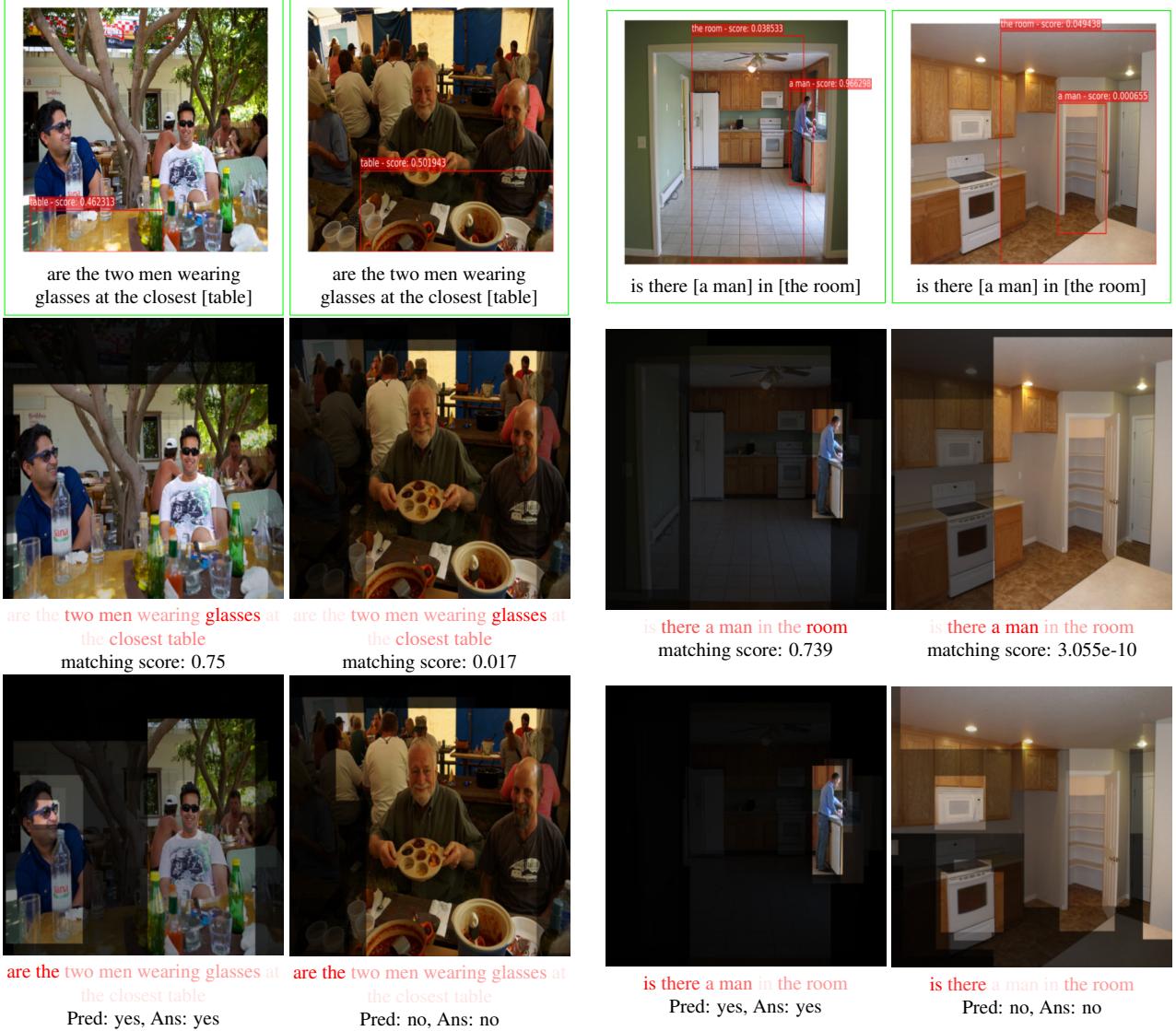
Method	Image Annotation			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10
DVSA [5]	11.8	32.5	45.4	8.9	24.9	36.3
FV [6]	17.3	39.0	50.2	10.8	28.3	40.1
OEM [10]	23.3	50.5	65.0	18.0	43.6	57.6
VQA [7]	23.5	50.7	63.6	16.7	40.5	53.8
VSE++ [3]	41.3	69.2	81.2	30.3	59.1	72.4
S-E Model [4]	42.8	72.3	83.0	33.1	62.9	75.5
Ours	42.2	69.1	80.6	33.2	64.2	76.5

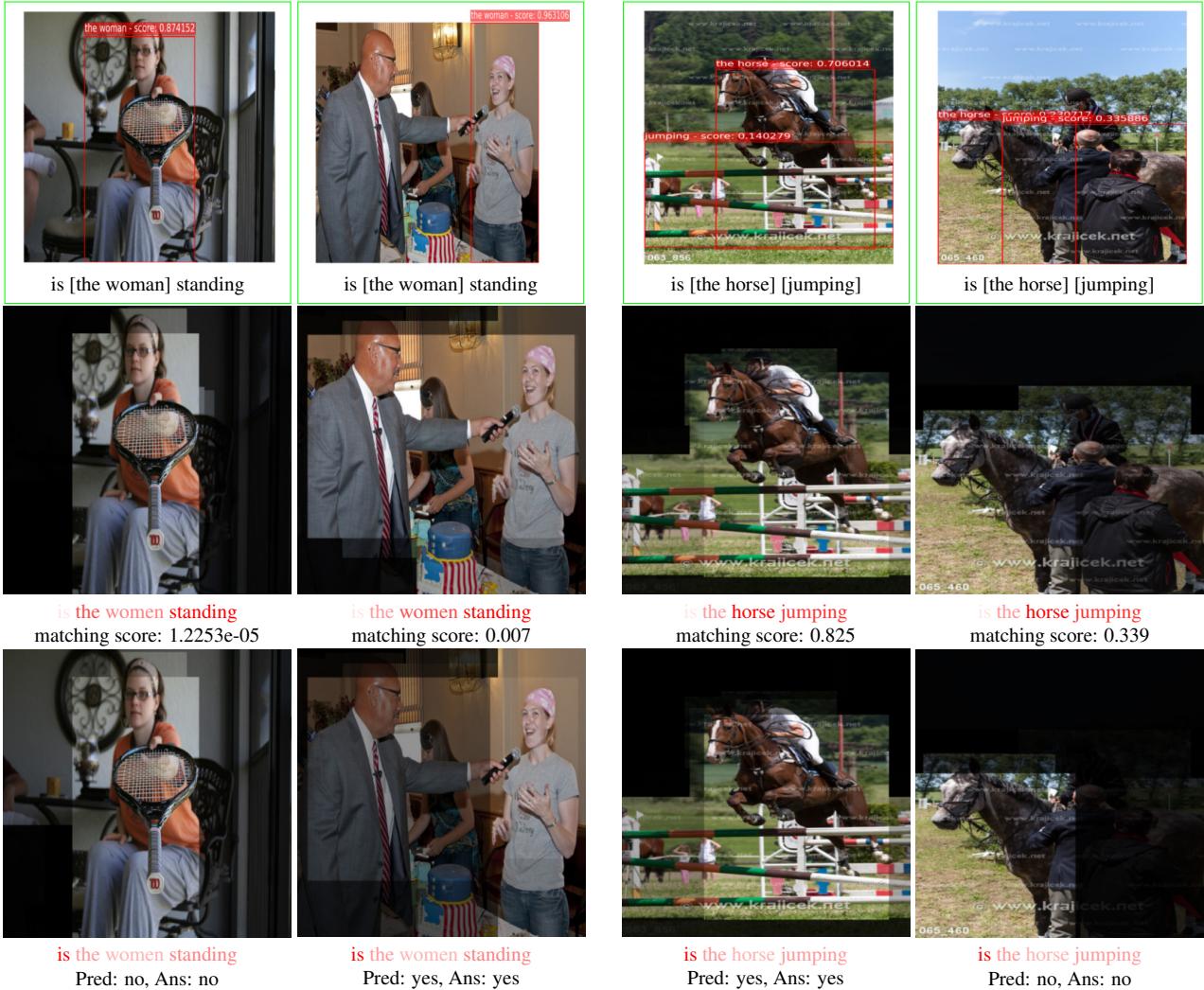
D. Visualization of Inference of Three Tasks on Visual Question Answering

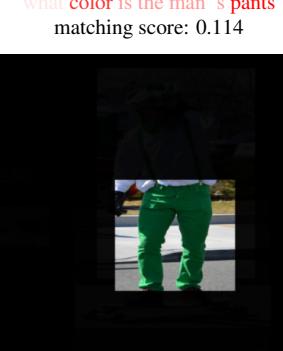
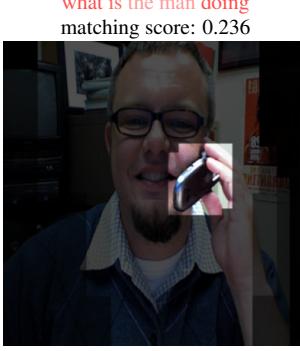
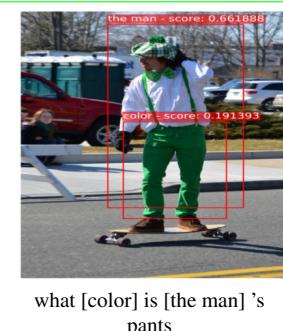
In Fig. 2 of the main paper, we show a few examples of visualization of inference of VG, ICR, and VQA on complementary image-question pairs of VQA 2.0 (i.e., pairs of the same questions and different images). We show here more examples for success cases (Sec. D.1) as well as failure cases (Sec. D.2).

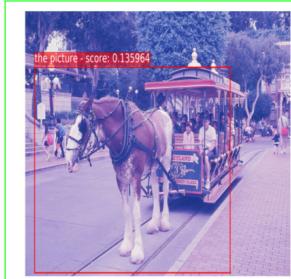
D.1. Success Cases

We first show visualization for success cases, i.e., image-question pairs for which our network provides the correct answers. As in Fig. 2 of the main paper, each of the left and right panels on each page shows visualization for a complementary image-question pair. The same observation given in the main paper applies to these examples.









how many horses are in [the picture]
matching score: 0.136



how many horses are in [the picture]
matching score: 0.161



how many different poses are in
[this shot]
matching score: 0.043711



how many different poses are in
[this shot]
matching score: 0.090226



how many horses are in the picture
matching score: 0.136



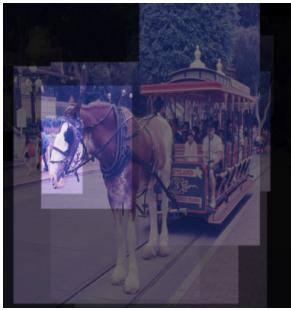
how many horses are in the picture
matching score: 0.161



how many different poses are in
[this shot]
matching score: 0.03



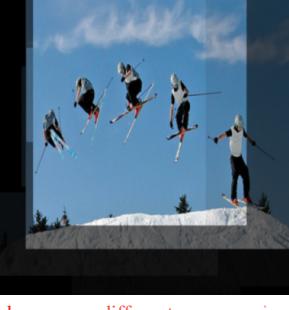
how many different poses are in
[this shot]
matching score: 0.01



how many horses are in the picture
Pred: 1, Ans: 1



how many horses are in the picture
Pred: 2, Ans: 2



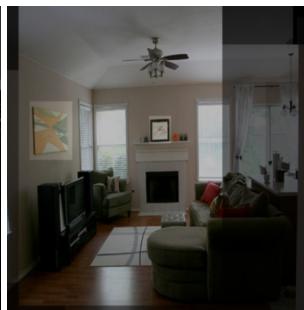
how many different poses are in
[this shot]
Pred: 5, Ans: 5



how many different poses are in
[this shot]
Pred: 2, Ans: 2



how many pictures are on the wall
matching score: 9.477e-07



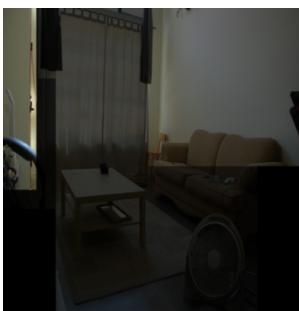
how many pictures are on the wall
matching score: 5.978e-04



what color is lit up on the street
lights
matching score: 2.6e-05



what color is lit up on the street
lights
matching score: 4.221e-06



how many pictures are on the wall
Pred: 0, Ans: 0



how many pictures are on the wall
Pred: 2, Ans: 2



what color is lit up on the street
lights
Pred: green, Ans: green

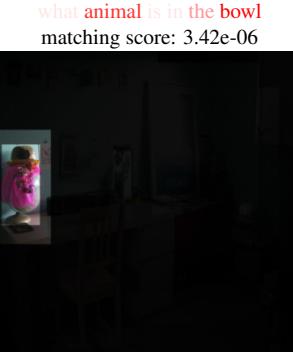
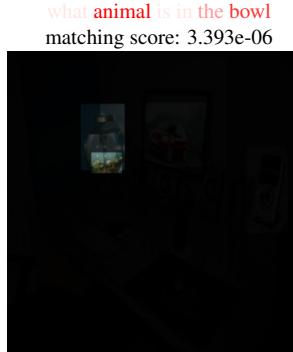
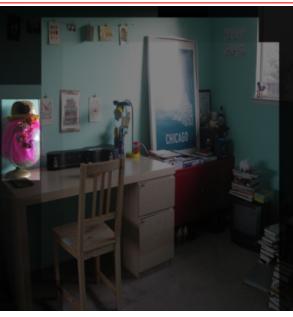
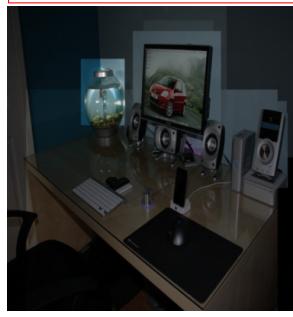
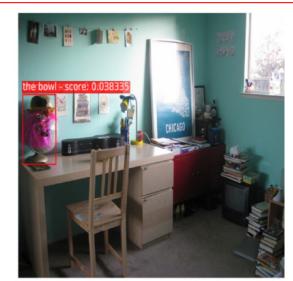


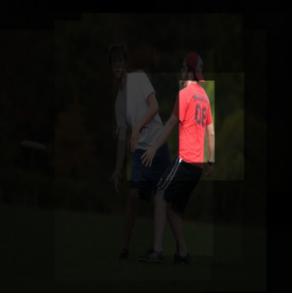
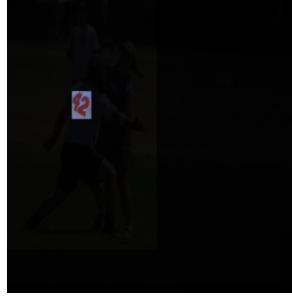
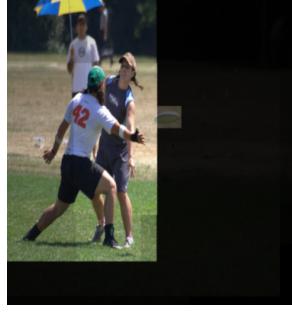
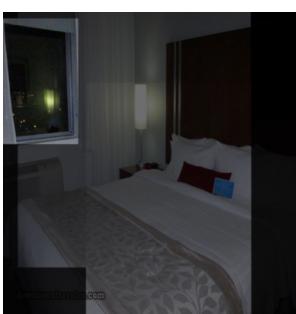
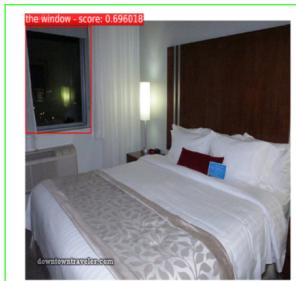
what color is lit up on the street
lights
Pred: white, Ans: white

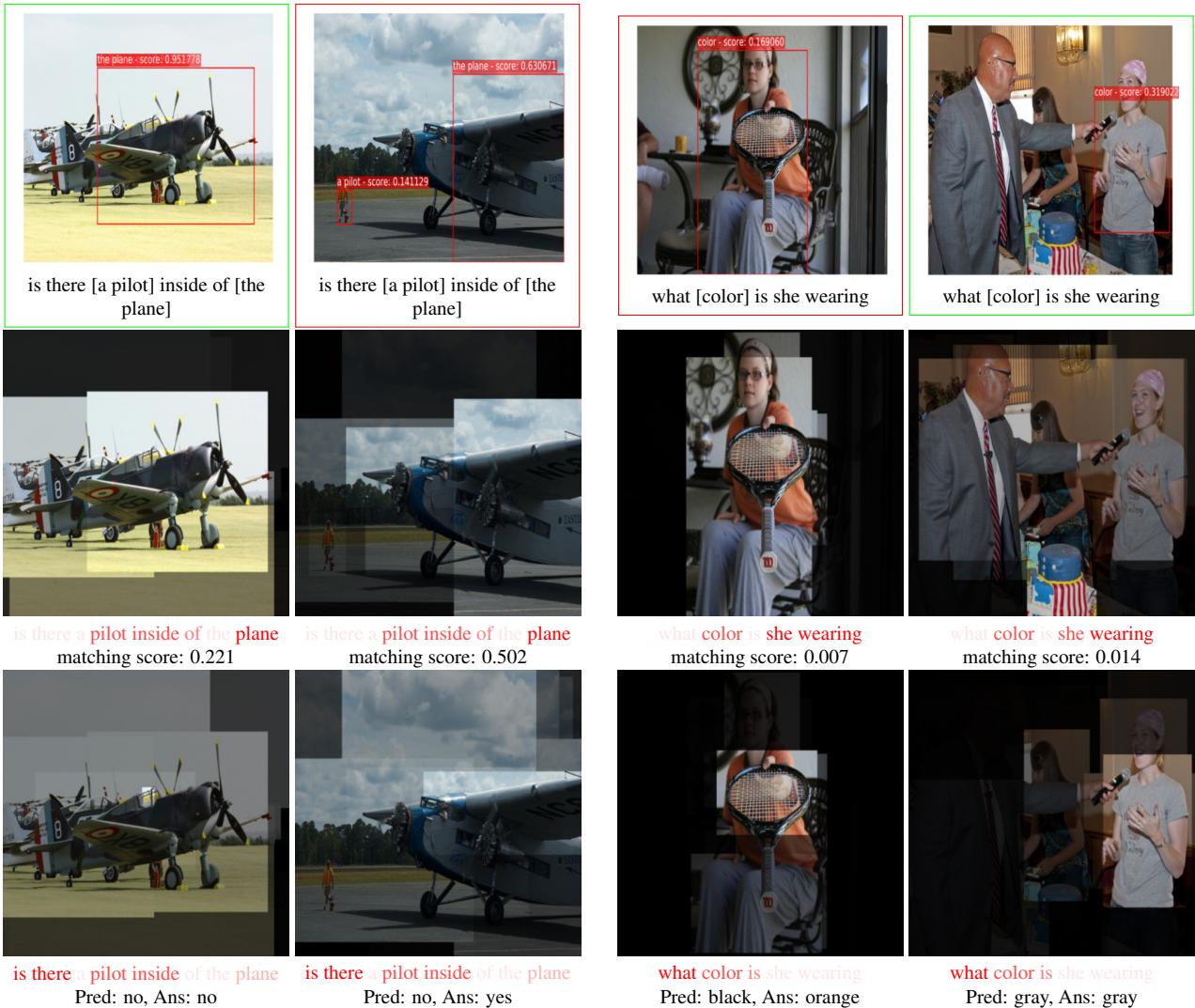
D.2. Failure Cases

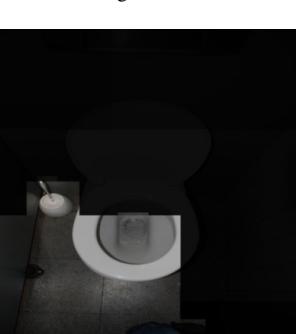
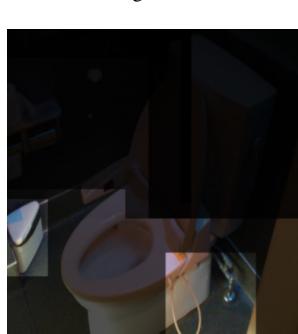
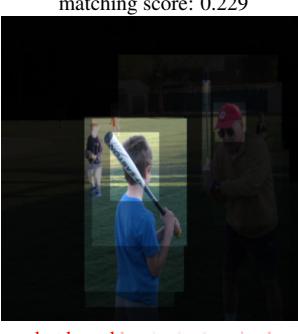
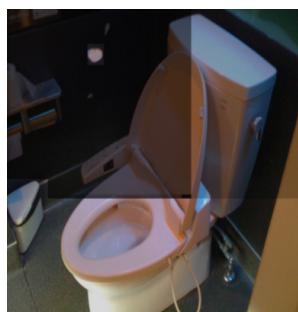
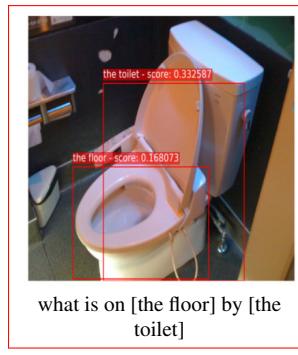
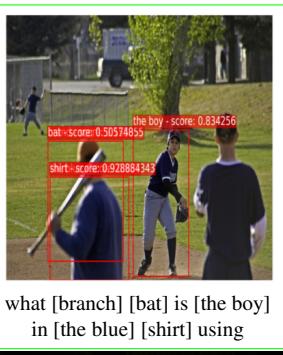
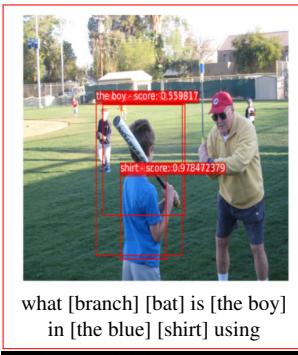
We next show failure cases, i.e., image-question pairs for which our network provides at least one wrong answer for the VQA task. The red bounding boxes indicate wrong answers and the green ones indicate correct answers. From the examples shown below, we can categorize failures for the VQA task into the following typical cases, for each of which we can explain why our network provides wrong answers and suggest possible solutions:

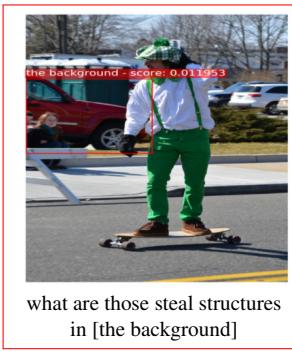
- 1) Although the VG and ICR decoders are able to correctly locate objects or concepts in the input image that appear in the input question, the VQA decoder fails to distinguish different objects or concepts that have similar appearance. This may be attributable to that the pretrained Faster R-CNN used for extracting image features is not trained to distinguish fine-grained concepts (e.g., “*terrier*” and “*lab*” (i.e., Labrador retriever), “*round*” and “*oval*”). It may help to train the Faster R-CNN with more fine-grained concepts.
- 2) The network is unable to locate relevant image regions. This often occurs when the VG and ICR decoder at the lower layer of the network are unable to detect correct regions (e.g., “*the bottom corner*” or “*inside of the plane*”), which leads to the failure of the VQA decoder. This is mostly because the Faster R-CNN fails to extract right regions or extracts excessively large regions containing many objects.
- 3) Questions require general knowledge that cannot be learned from only the training data (e.g., “*acidic food*” or “*reflection*”). For instance, for the question “*is this acidic food*”, we can observe from the response of the VG and ICR decoders that the network recognizes all the food in the image as “*acidic food*”; for the question “*is there a reflection in the window*”, the VG and ICR decoders give high confident scores for “*reflection*” even there is not.
- 4) The answers given by the network are judged incorrect simply because they are not listed in the given set of correct answers in the dataset, but they are actually considered to be correct answers. For example, for the question “*what is on the floor by the toilet*”, both of “*tile*” and “*trash can*” should be correct answers, but only the latter is listed in the correct answer set.











what are those steal structures
in [the background]



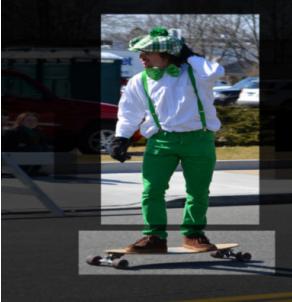
what are those steal structures
in [the background]



how many tags are on [the
suitcase]



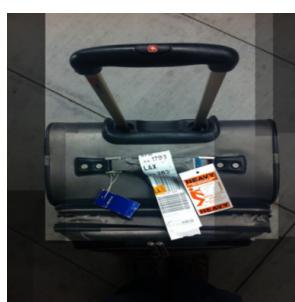
how many tags are on [the
suitcase]



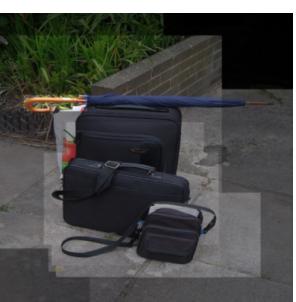
matching score: 7.355e-06



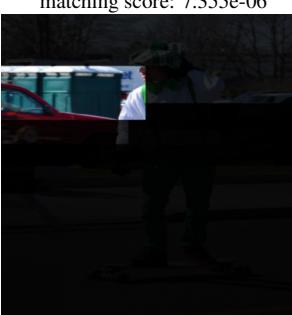
matching score: 1.032e-06



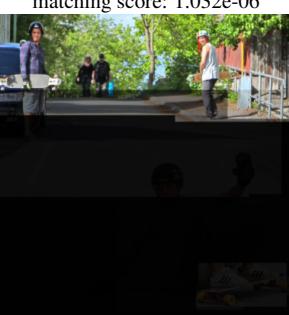
matching score: 0.893



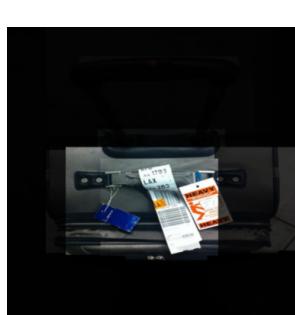
matching score: 0.457



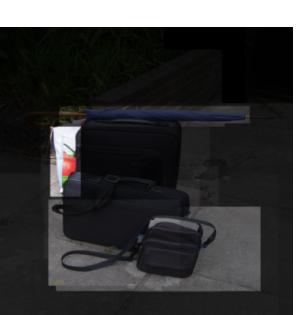
Pred: houses, Ans: toilets



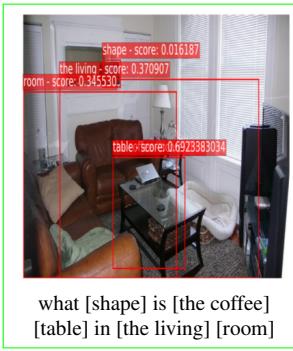
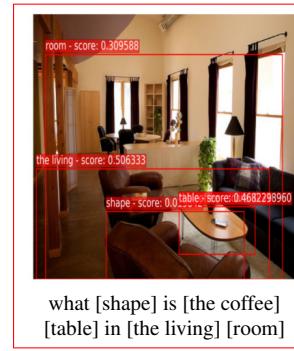
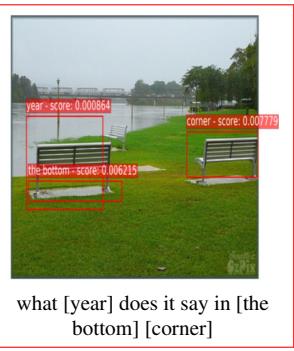
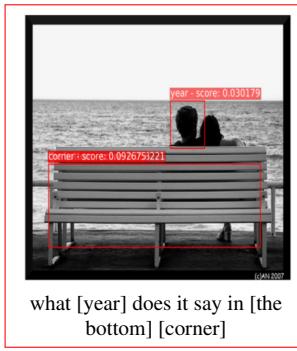
Pred: fence, Ans: fence



Pred: 4, Ans: 3



Pred: 0, Ans: 0



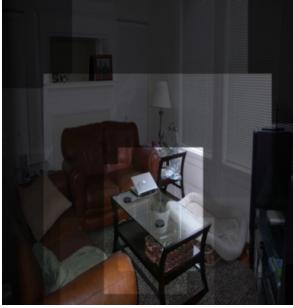
what year does it say in the bottom
corner
matching score: 0.001



what year does it say in the bottom
corner
matching score: 0.008



what shape is the coffee table in the living room
matching score: 0.821



what shape is the coffee table in the living room
matching score: 0.902



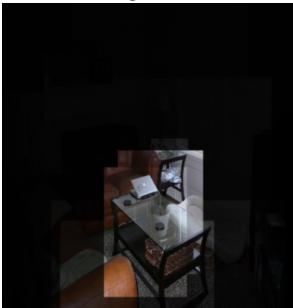
what year does it say in the bottom
corner
Pred: 0, Ans: 2007



Pred: 2000, Ans: 2010



Pred: round, Ans: oval



Pred: rectangle, Ans: rectangle



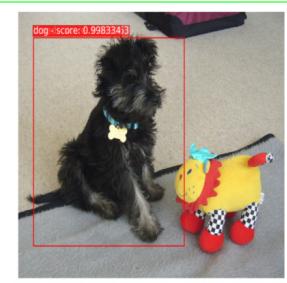
what [color] are the curtains



what [color] are the curtains



what [breed] of [dog] is this



what [breed] of [dog] is this



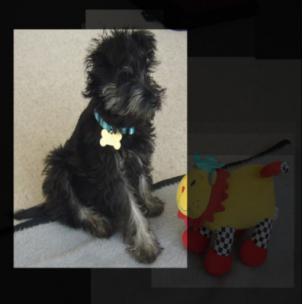
what color are the curtains
matching score: 0.607



what color are the curtains
matching score: 0.013



what breed of dog is this
matching score: 0.239



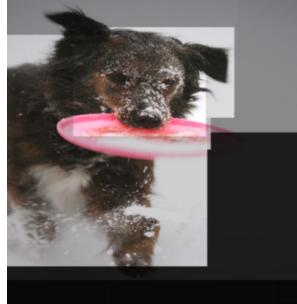
what breed of dog is this
matching score: 0.145



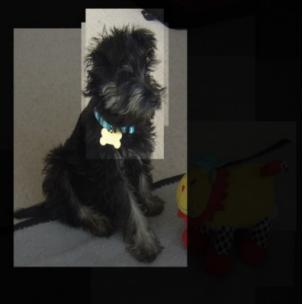
what color are the curtains
Pred: white, Ans: white



what color are the curtains
Pred: white, Ans: yellow and white



what breed of dog is this
Pred: terrier, Ans: lab



what breed of dog is this
Pred: terrier, Ans: terrier

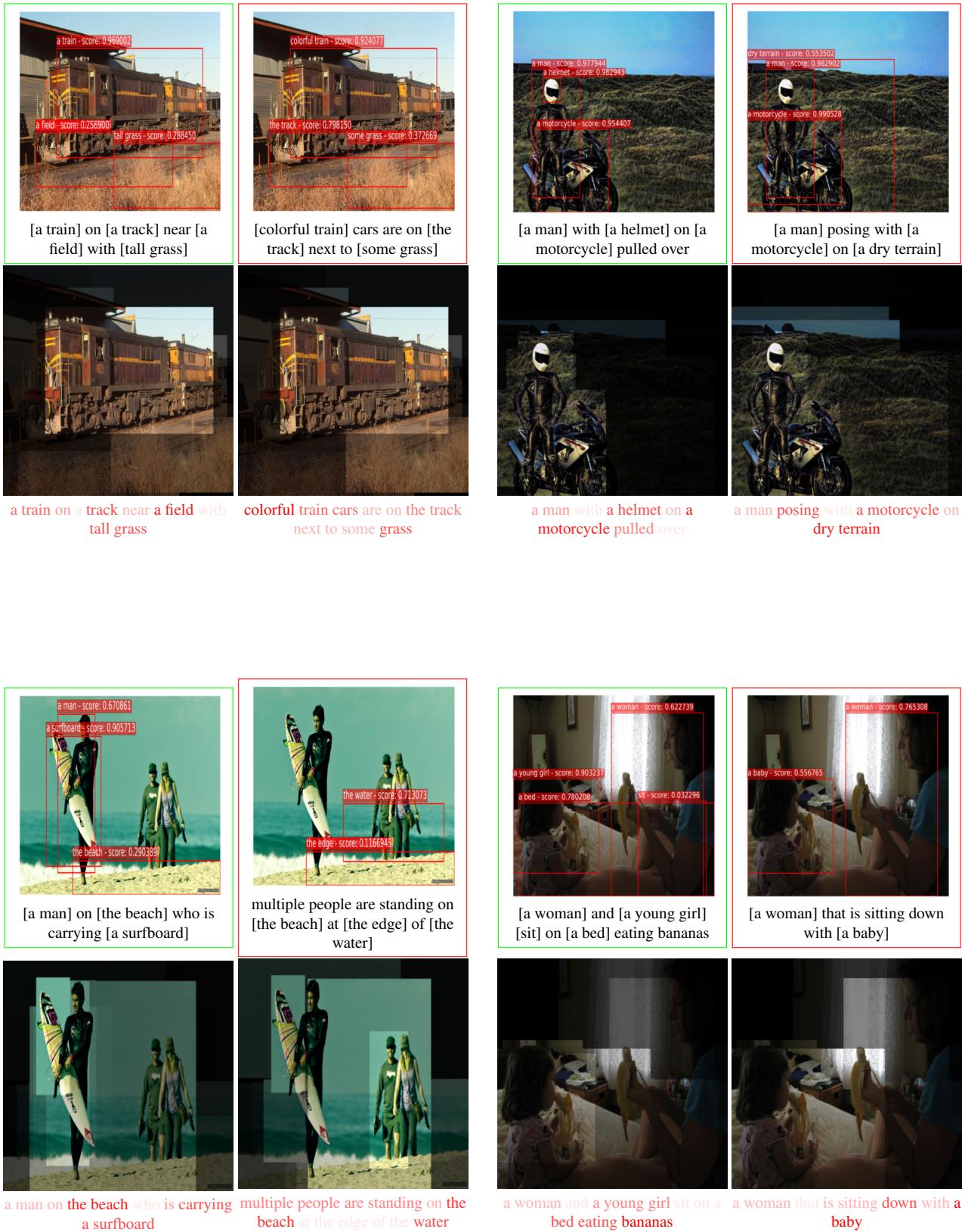
E. Visualization of Inference of VG and ICR on Image Caption Retrieval

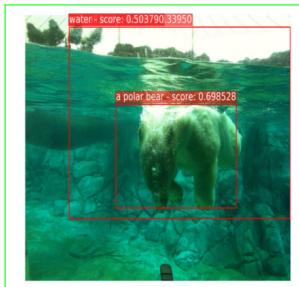
We show visualization of our network using an ICR dataset, MSCOCO. As success cases are not so informative, which are often shown in previous studies, we show only failure cases for the two subtasks of ICR, more specifically, the cases where the top-1 prediction is not correct for image annotation (Sec. E.1) and image retrieval (Sec. E.2). Each panel consists of 2×2 image-caption pairs; the first row shows VG and the second row shows ICR; the first column (with a green box) shows VG and ICR visualization for the ground-truth image-caption pair and the second column (with a red box) shows a pair of an input image and the predicted top-1 caption for image annotation and a pair of the predicted top-1 image and an input caption for image retrieval .

We think that the failure cases can be categorized into the following three types:

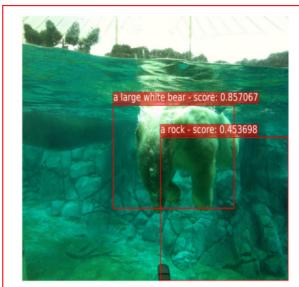
- 1) The network incorrectly recognizes objects or concepts that have similar appearance, since the Faster R-CNN features are not rich enough to distinguish them (e.g., “computer monitor” and “television”). Specifically, the VG decoder detects wrong objects, matching the input image with a wrong caption.
- 2) The VG and ICR decoders correctly align objects in the input caption with the corresponding image regions, but fail to recognize their actions (e.g., “*submerged in a small body of water*” vs. “*standing on a rock*”; “*perches*” vs. “*standing*”). Such failures may be eliminated by creating a dataset for the task of predicting such actions and using it in the joint training.
- 3) As in the case of VQA, although the top-1 caption or image predicted by our network matches well with the input image or caption, it is judged wrong because it is not listed in the set of correct answers. For image annotation, this happens because the same image content can be described in many ways (e.g., “*a train on a track near a field with tall grass*” or “*colorful train cars are on the track next to some grass*”); or because “correct” captions explain only one of multiple contents contained in input images (e.g., “*a man on the beach who is carrying a surfboard*” and “*multiple people are standing on the beach at the edge of the water*”). For image retrieval, the same often occurs when the input captions provide only too general explanation of a scene, most of which tend to be short simple captions, such as “*there are people flying kites in the park*”. It should be noted that even if this is the case, the VG decoder is able to correctly detect objects in most cases.

E.1. Image Annotation

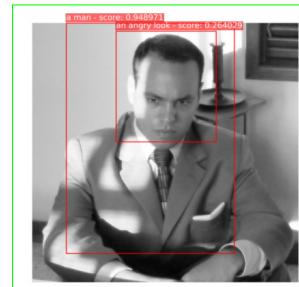




[a polar bear] fully submerged in a small body of [water]



[a large white bear] standing on [a rock]



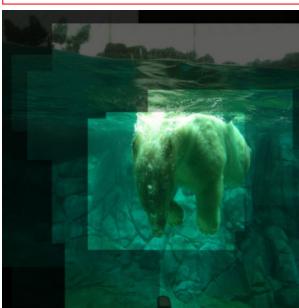
[a man] sitting down with [an angry look]



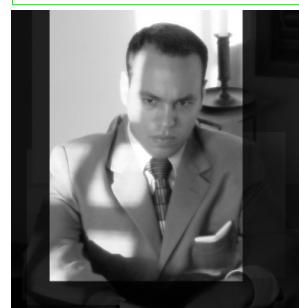
[a man] who is wearing [a tie] that is too small for him



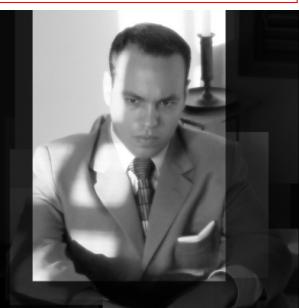
a polar bear fully submerged in a small body of water



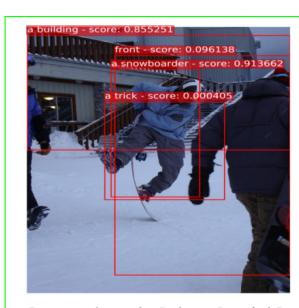
a large white bear standing on a rock



a man sitting down with an angry look



a man who is wearing a tie that is too small for him



[a snowboarder] does [a trick] in [front] of [a building]



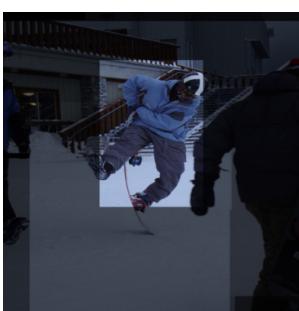
[a person] doing [a snowboarding] [stunt] on [a rail]



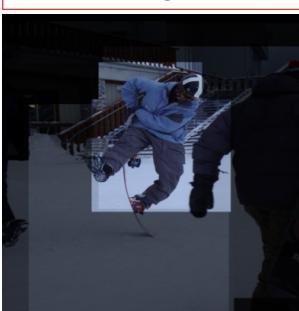
[an orange tabby] [cat] perches near [a computer] [monitor]



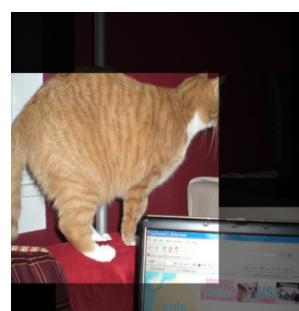
[a cat] is sitting on [the floor] and watching [television]



a snowboarder does a trick in front of a building



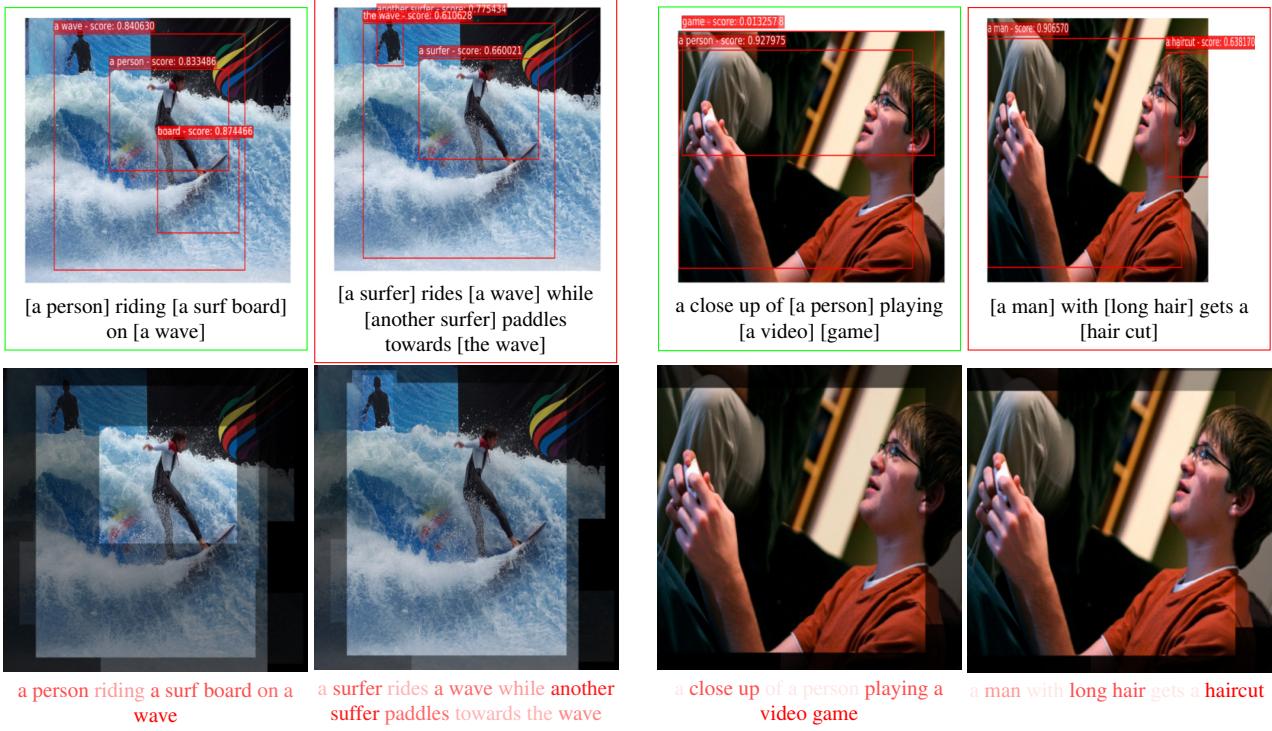
a person doing a snow boarding stunt on a rail



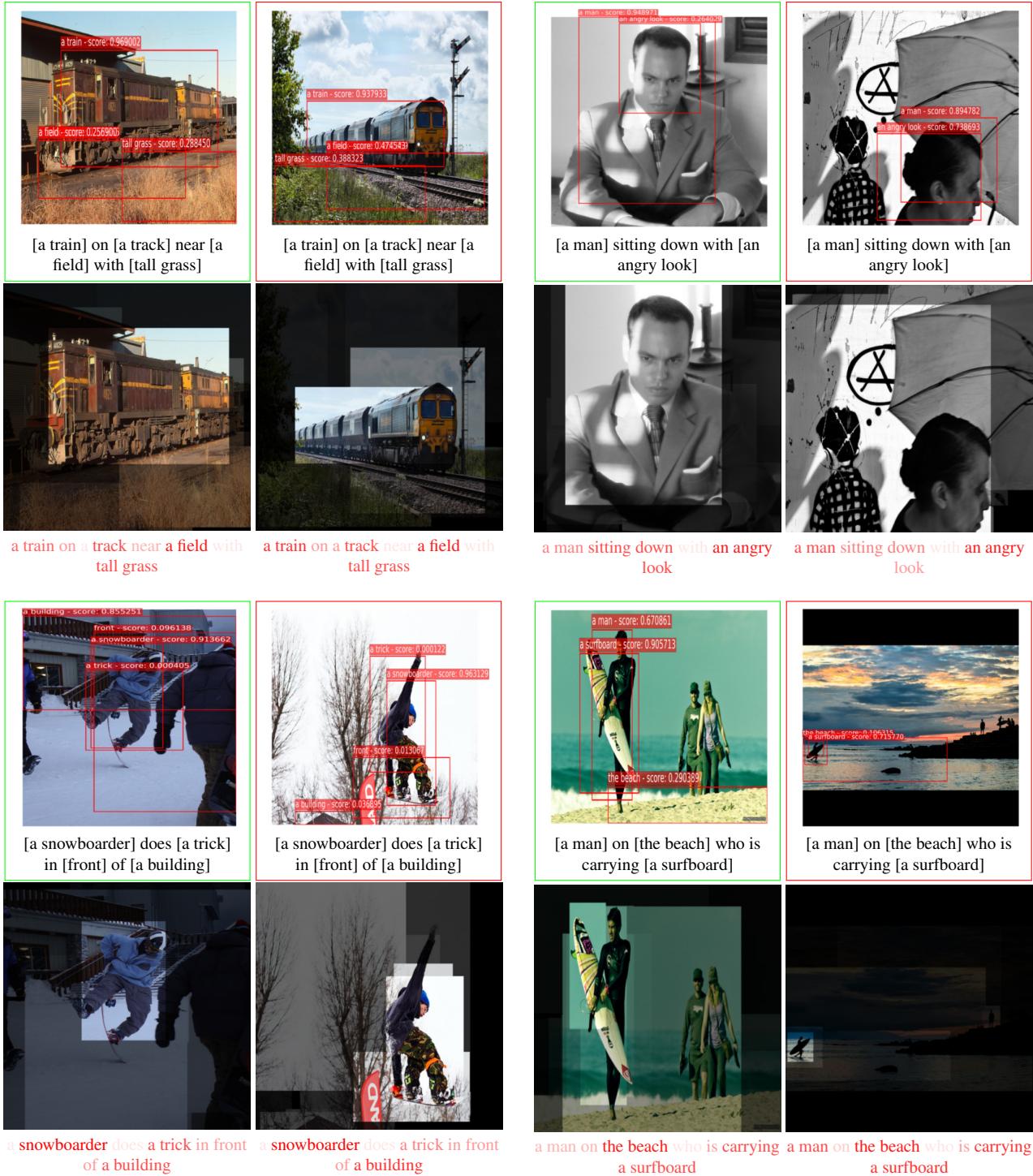
an orange tabby cat perches near a computer monitor

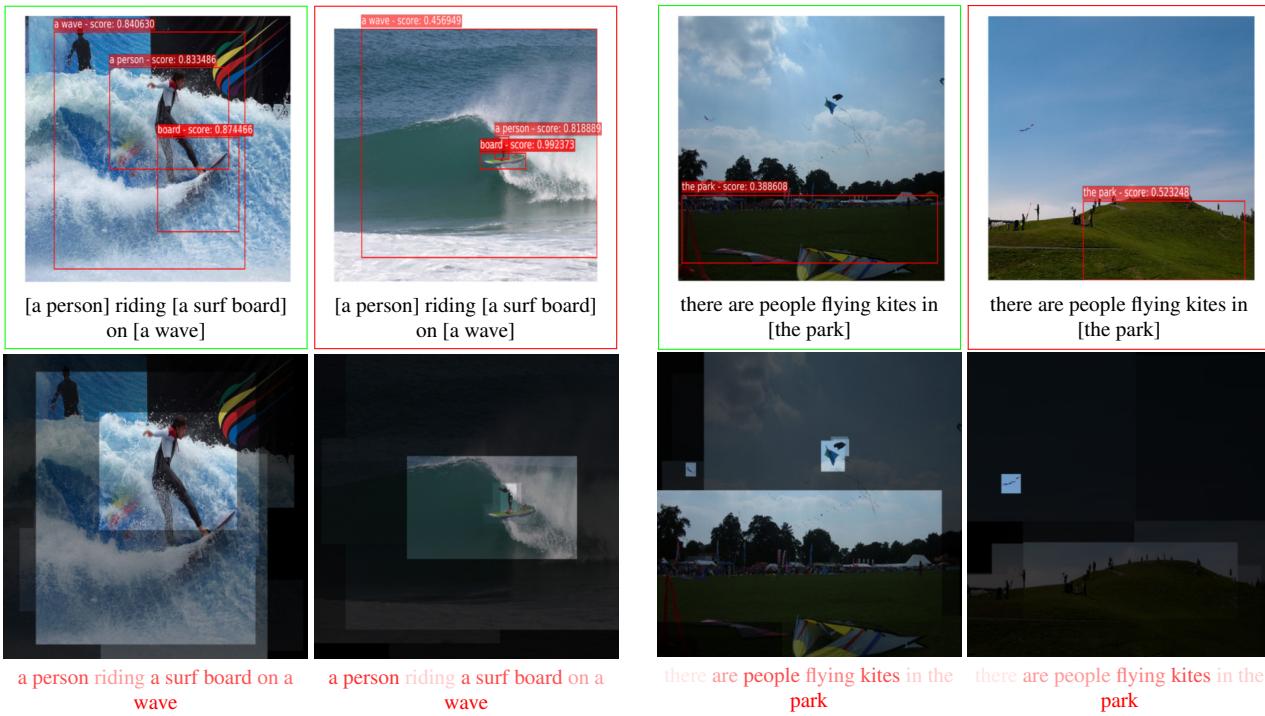


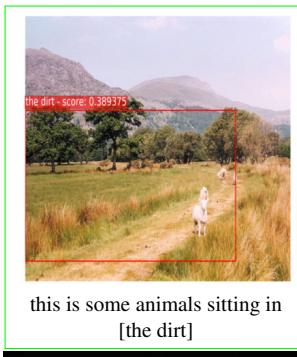
a cat is sitting on the floor and watching television



E.2. Image Retrieval







this is some animals sitting in [the dirt]



this is some animals sitting in [the dirt]



[a little girl] doing some arts and crafts



[a little girl] doing some arts and crafts



this is some animals sitting in the dirt



this is some animals sitting in the dirt



a little girl doing some arts and crafts



a little girl doing some arts and crafts



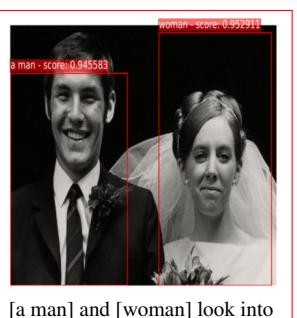
[a smiling man] stands on [a snowy hill] with [some ski] poles



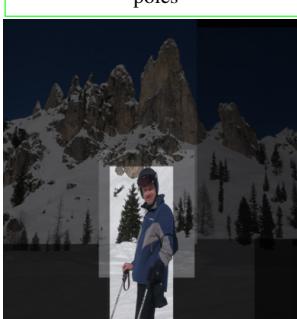
[a smiling man] stands on [a snowy hill] with [some ski] poles



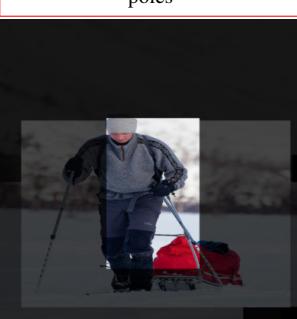
[a man] and [woman] look into each others eyes while getting married



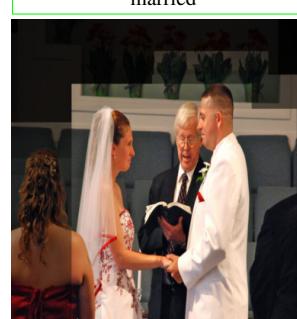
[a man] and [woman] look into each others eyes while getting married



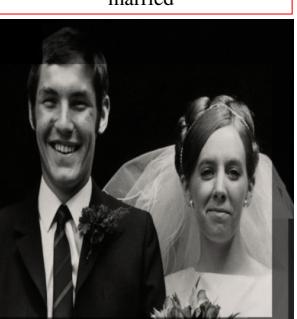
a smiling man stands on a snowy hill with some ski poles



a smiling man stands on a snowy hill with some ski poles



a man and woman look into each others eyes while getting married



a man and woman look into each others eyes while getting married

References

- [1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015. [2](#)
- [2] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 2009. [1](#)
- [3] F. Faghri, D. J. Fleet, R. Kiros, and S. Fidler. VSE++: improved visual-semantic embeddings. *arXiv preprint arXiv:1707.05612*, 2017. [3](#)
- [4] Y. Huang, Q. Wu, C. Song, and L. Wang. Learning semantic concepts and order for image and sentence matching. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [3](#)
- [5] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. [3](#)
- [6] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. [3](#)
- [7] X. Lin and D. Parikh. Leveraging visual question answering for image-caption ranking. In *European Conference on Computer Vision (ECCV)*, 2016. [3](#)
- [8] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2014. [1](#)
- [9] D. Teney, P. Anderson, X. He, and A. van den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [1, 2](#)
- [10] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun. Order-embeddings of images and language. In *International Conference on Learning Representations (ICLR)*, 2016. [3](#)