Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering

Peter Anderson[1]* Xiaodong He[2] Chris Buehler[3] Damien Teney[4]
Mark Johnson[5] Stephen Gould[1] Lei Zhang[3]
[1]Australian National University [2]JD AI Research [3]Microsoft Research [4]University of Adelaide [5]Macquarie University
[1]firstname.lastname@anu.edu.au, [2]xiaodong.he@jd.com, [3]{chris.buehler,leizhang}@microsoft.com
[4]damien.teney@adelaide.edu.au, [5]mark.johnson@mq.edu.au

Abstract

Top-down visual attention mechanisms have been used extensively in image captioning and visual question answering (VQA) to enable deeper image understanding through fine-grained analysis and even multiple steps of reasoning. In this work, we propose a combined bottom-up and top-down attention mechanism that enables attention to be calculated at the level of objects and other salient image regions. This is the natural basis for attention to be considered. Within our approach, the bottom-up mechanism (based on Faster R-CNN) proposes image regions, each with an associated feature vector, while the top-down mechanism determines feature weightings. Applying this approach to image captioning, our results on the MSCOCO test server establish a new state-of-the-art for the task, achieving CIDEr / SPICE / BLEU-4 scores of 117.9, 21.5 and 36.9, respectively. Demonstrating the broad applicability of the method, applying the same approach to VQA we obtain first place in the 2017 VQA Challenge.

1. Introduction

Problems combining image and language understanding such as image captioning [4] and visual question answering (VQA) [12] continue to inspire considerable research at the boundary of computer vision and natural language processing. In both these tasks it is often necessary to perform some fine-grained visual processing, or even multiple steps of reasoning to generate high quality outputs. As a result, visual attention mechanisms have been widely adopted in both image captioning [34, 27, 48, 46] and VQA [11, 28, 45, 47, 51]. These mechanisms improve performance by learning to focus on the regions of the image that are salient and are currently based on deep neural network architectures.

*Work performed while interning at Microsoft.

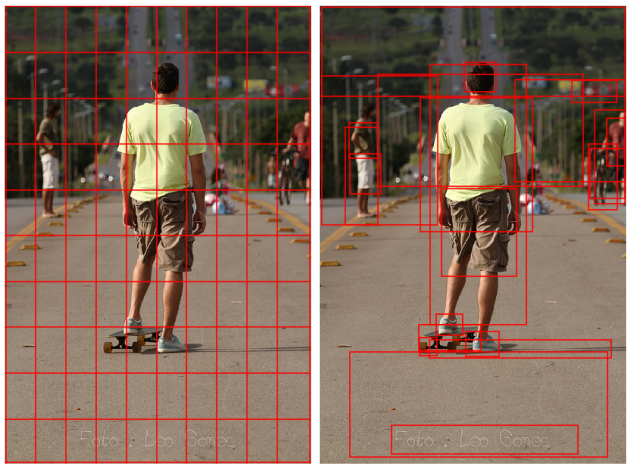


Figure 1. Typically, attention models operate on CNN features corresponding to a uniform grid of equally-sized image regions (left). Our approach enables attention to be calculated at the level of objects and other salient image regions (right).

In the human visual system, attention can be focused volitionally by top-down signals determined by the current task (e.g., looking for something), and automatically by bottom-up signals associated with unexpected, novel or salient stimuli [3, 6]. In this paper we adopt similar terminology and refer to attention mechanisms driven by non-visual or task-specific context as 'top-down', and purely visual feed-forward attention mechanisms as 'bottom-up'.

Most conventional visual attention mechanisms used in image captioning and VQA are of the top-down variety. Taking as context a representation of a partially-completed caption output, or a question relating to the image, these mechanisms are typically trained to selectively attend to the output of one or more layers of a convolutional neural net (CNN). However, this approach gives little consideration to how the image regions that are subject to attention are determined. As illustrated conceptually in Figure 1, the resulting

input regions correspond to a uniform grid of equally sized and shaped neural receptive fields – irrespective of the content of the image. To generate more human-like captions and question answers, objects and other salient image regions are a much more natural basis for attention [10, 36].

In this paper we propose a combined bottom-up and top-down visual attention mechanism. The bottom-up mechanism proposes a set of salient image regions, with each region represented by a pooled convolutional feature vector. Practically, we implement bottom-up attention using Faster R-CNN [33], which represents a natural expression of a bottom-up attention mechanism. The top-down mechanism uses task-specific context to predict an attention distribution over the image regions. The attended feature vector is then computed as a weighted average of image features over all regions.

We evaluate the impact of combining bottom-up and top-down attention on two tasks. We first present an image captioning model that takes multiple glimpses of salient image regions during caption generation. Empirically, we find that the inclusion of bottom-up attention has a significant positive benefit for image captioning. Our results on the MSCOCO test server establish a new state-of-the-art for the task, achieving CIDEr / SPICE / BLEU-4 scores of 117.9, 21.5 and 36.9. respectively (outperforming all published and unpublished work at the time). Demonstrating the broad applicability of the method, we additionally present a VQA model using the same bottom-up attention features. Using this model we obtain first place in the 2017 VQA Challenge, achieving 70.3% overall accuracy on the VQA v2.0 test-standard server. Code, models and pre-computed image features are available from the project website[1].

2. Related Work

A large number of attention-based deep neural networks have been proposed for image captioning and VQA. Typically, these models can be characterized as top-down approaches, with context provided by a representation of a partially-completed caption in the case of image captioning [34, 27, 48, 46], or a representation of the question in the case of VQA [11, 28, 45, 47, 51]. In each case attention is applied to the output of one or more layers of a CNN, by predicting a weighting for each spatial location in the CNN output. However, determining the optimal number of image regions invariably requires an unwinnable trade-off between coarse and fine levels of detail. Furthermore, the arbitrary positioning of the regions with respect to image content may make it more difficult to detect objects that are poorly aligned to regions and to bind visual concepts associated with the same object.

Comparatively few previous works have considered applying attention to salient image regions. We are aware of two papers. Jin et al. [18] use selective search [42] to identify salient image regions, which are filtered with a classifier then resized and CNN-encoded as input to an image captioning model with attention. The Areas of Attention captioning model [30] uses either edge boxes [52] or spatial transformer networks [17] to generate image features, which are processed using an attention model based on three bi-linear pairwise interactions [30]. In this work, rather than using hand-crafted or differentiable region proposals [42, 52, 17], we leverage Faster R-CNN [33], establishing a closer link between vision and language tasks and recent progress in object detection. With this approach we are able to pre-train our region proposals on object detection datasets. Conceptually, the advantages should be similar to pre-training visual representations on ImageNet [35] and leveraging significantly larger cross-domain knowledge. We additionally apply our method to VQA, establishing the broad applicability of our approach.

3. Approach

Given an image I, both our image captioning model and our VQA model take as input a possibly variably-sized set of k image features, $V = \{\boldsymbol{v}_1, ..., \boldsymbol{v}_k\}, \boldsymbol{v}_i \in \mathbb{R}^D$, such that each image feature encodes a salient region of the image. The spatial image features V can be variously defined as the output of our bottom-up attention model, or, following standard practice, as the spatial output layer of a CNN. We describe our approach to implementing a bottom-up attention model in Section 3.1. In Section 3.2 we outline the architecture of our image captioning model and in Section 3.3 we outline our VQA model. We note that for the top-down attention component, both models use simple one-pass attention mechanisms, as opposed to the more complex schemes of recent models such as stacked, multi-headed, or bidirectional attention [47, 16, 20, 28] that could also be applied.

3.1. Bottom-Up Attention Model

The definition of spatial image features V is generic. However, in this work we define spatial regions in terms of bounding boxes and implement bottom-up attention using Faster R-CNN [33]. Faster R-CNN is an object detection model designed to identify instances of objects belonging to certain classes and localize them with bounding boxes. Other region proposal networks could also be trained as an attentive mechanism [32, 25].

Faster R-CNN detects objects in two stages. The first stage, described as a Region Proposal Network (RPN), predicts object proposals. A small network is slid over features at an intermediate level of a CNN. At each spatial location the network predicts a class-agnostic objectness score and a bounding box refinement for anchor boxes of multiple scales and aspect ratios. Using greedy non-maximum

[1] http://www.panderson.me/up-down-attention

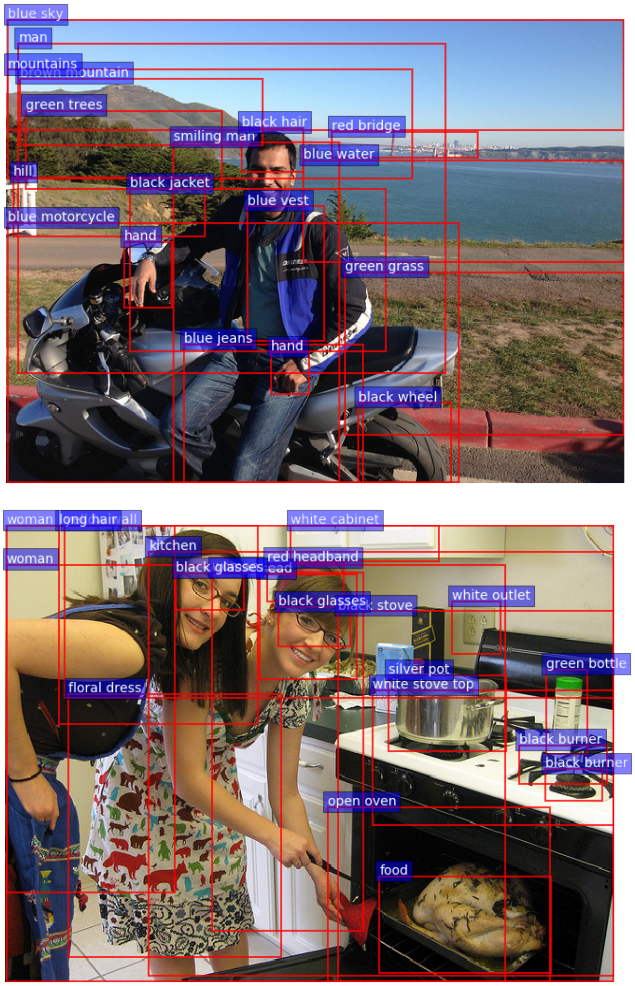


Figure 2. Example output from our Faster R-CNN bottom-up attention model. Each bounding box is labeled with an attribute class followed by an object class. Note however, that in captioning and VQA we utilize only the feature vectors – not the predicted labels.

suppression with an intersection-over-union (IoU) threshold, the top box proposals are selected as input to the second stage. In the second stage, region of interest (RoI) pooling is used to extract a small feature map (e.g. 14×14) for each box proposal. These feature maps are then batched together as input to the final layers of the CNN. The final output of the model consists of a softmax distribution over class labels and class-specific bounding box refinements for each box proposal.

In this work, we use Faster R-CNN in conjunction with the ResNet-101 [13] CNN. To generate an output set of image features V for use in image captioning or VQA, we take the final output of the model and perform non-maximum suppression for each object class using an IoU threshold. We then select all regions where any class detection probability exceeds a confidence threshold. For each selected region i, $\boldsymbol{v}_i$ is defined as the mean-pooled convolutional feature from this region, such that the dimension D of the image feature vectors is 2048. Used in this fashion, Faster R-CNN effectively functions as a 'hard' attention mechanism, as only a relatively small number of image bounding box features are selected from a large number of possible configurations.

To pretrain the bottom-up attention model, we first initialize Faster R-CNN with ResNet-101 pretrained for classification on ImageNet [35]. We then train on Visual Genome [21] data. To aid the learning of good feature representations, we add an additional training output for predicting attribute classes (in addition to object classes). To predict attributes for region i, we concatenate the mean pooled convolutional feature $\boldsymbol{v}_i$ with a learned embedding of the ground-truth object class, and feed this into an additional output layer defining a softmax distribution over each attribute class plus a 'no attributes' class.

The original Faster R-CNN multi-task loss function contains four components, defined over the classification and bounding box regression outputs for both the RPN and the final object class proposals respectively. We retain these components and add an additional multi-class loss component to train the attribute predictor. In Figure 2 we provide some examples of model output.

3.2. Captioning Model

Given a set of image features V, our proposed captioning model uses a 'soft' top-down attention mechanism to weight each feature during caption generation, using the existing partial output sequence as context. This approach is broadly similar to several previous works [34, 27, 46]. However, the particular design choices outlined below make for a relatively simple yet high-performing baseline model. Even without bottom-up attention, our captioning model achieves performance comparable to state-of-the-art on most evaluation metrics (refer Table 1).

At a high level, the captioning model is composed of two LSTM [15] layers using a standard implementation [9]. In the sections that follow we will refer to the operation of the LSTM over a single time step using the following notation:

$$\boldsymbol{h}_t = \text{LSTM}(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}) \tag{1}$$

where $\boldsymbol{x}_t$ is the LSTM input vector and $\boldsymbol{h}_t$ is the LSTM output vector. Here we have neglected the propagation of memory cells for notational convenience. We now describe the formulation of the LSTM input vector $\boldsymbol{x}_t$ and the output vector $\boldsymbol{h}_t$ for each layer of the model. The overall captioning model is illustrated in Figure 3.

3.2.1 Top-Down Attention LSTM

Within the captioning model, we characterize the first LSTM layer as a top-down visual attention model, and the

second LSTM layer as a language model, indicating each layer with superscripts in the equations that follow. Note that the bottom-up attention model is described in Section 3.1, and in this section its outputs are simply considered as features V. The input vector to the attention LSTM at each time step consists of the previous output of the language LSTM, concatenated with the mean-pooled image feature $\bar{\boldsymbol{v}} = \frac{1}{k}\sum_i \boldsymbol{v}_i$ and an encoding of the previously generated word, given by:

$$\boldsymbol{x}_t^1 = [\boldsymbol{h}_{t-1}^2, \bar{\boldsymbol{v}}, W_e \Pi_t] \tag{2}$$

where $W_e \in \mathbb{R}^{E\times|\Sigma|}$ is a word embedding matrix for a vocabulary Σ, and Π_t is one-hot encoding of the input word at timestep t. These inputs provide the attention LSTM with maximum context regarding the state of the language LSTM, the overall content of the image, and the partial caption output generated so far, respectively. The word embedding is learned from random initialization without pretraining.

Given the output $\boldsymbol{h}_t^1$ of the attention LSTM, at each time step t we generate a normalized attention weight $\alpha_{i,t}$ for each of the k image features $\boldsymbol{v}_i$ as follows:

$$a_{i,t} = \boldsymbol{w}_a^T \tanh\left(W_{va}\boldsymbol{v}_i + W_{ha}\boldsymbol{h}_t^1\right) \tag{3}$$

$$\boldsymbol{\alpha}_t = \text{softmax}\left(\boldsymbol{a}_t\right) \tag{4}$$

where $W_{va} \in \mathbb{R}^{H\times V}$, $W_{ha} \in \mathbb{R}^{H\times M}$ and $\boldsymbol{w}_a \in \mathbb{R}^H$ are learned parameters. The attended image feature used as input to the language LSTM is calculated as a convex combination of all input features:

$$\hat{\boldsymbol{v}}_t = \sum_{i=1}^{K} \alpha_{i,t}\boldsymbol{v}_i \tag{5}$$

3.2.2 Language LSTM

The input to the language model LSTM consists of the attended image feature, concatenated with the output of the attention LSTM, given by:

$$\boldsymbol{x}_t^2 = [\hat{\boldsymbol{v}}_t, \boldsymbol{h}_t^1] \tag{6}$$

Using the notation $y_{1:T}$ to refer to a sequence of words $(y_1, ..., y_T)$, at each time step t the conditional distribution over possible output words is given by:

$$p(y_t \mid y_{1:t-1}) = \text{softmax}\left(W_p\boldsymbol{h}_t^2 + \boldsymbol{b}_p\right) \tag{7}$$

where $W_p \in \mathbb{R}^{|\Sigma|\times M}$ and $\boldsymbol{b}_p \in \mathbb{R}^{|\Sigma|}$ are learned weights and biases. The distribution over complete output sequences is calculated as the product of conditional distributions:

$$p(y_{1:T}) = \prod_{t=1}^{T} p(y_t \mid y_{1:t-1}) \tag{8}$$

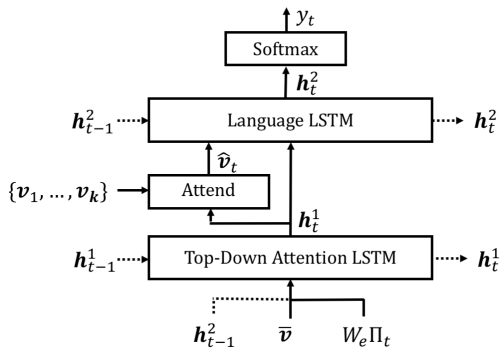


Figure 3. Overview of the proposed captioning model. Two LSTM layers are used to selectively attend to spatial image features $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_k\}$. These features can be defined as the spatial output of a CNN, or following our approach, generated using bottom-up attention.

3.2.3 Objective

Given a target ground truth sequence $y_{1:T}^*$ and a captioning model with parameters θ, we minimize the following cross entropy loss:

$$L_{XE}(\theta) = -\sum_{t=1}^{T} \log(p_\theta(y_t^* \mid y_{1:t-1}^*)) \tag{9}$$

For fair comparison with recent work [34] we also report results optimized for CIDEr [43]. Initializing from the cross-entropy trained model, we seek to minimize the negative expected score:

$$L_R(\theta) = -\mathbf{E}_{y_{1:T}\sim p_\theta}[r(y_{1:T})] \tag{10}$$

where r is the score function (e.g., CIDEr). Following the approach described as Self-Critical Sequence Training [34] (SCST), the gradient of this loss can be approximated:

$$\nabla_\theta L_R(\theta) \approx -(r(y_{1:T}^s) - r(\hat{y}_{1:T}))\nabla_\theta \log p_\theta(y_{1:T}^s) \tag{11}$$

where $y_{1:T}^s$ is a sampled caption and $r(\hat{y}_{1:T})$ defines the baseline score obtained by greedily decoding the current model. SCST (like other REINFORCE [44] algorithms) explores the space of captions by sampling from the policy during training. This gradient tends to increase the probability of sampled captions that score higher than the score from the current model.

In our experiments, we follow SCST but we speed up the training process by restricting the sampling distribution. Using beam search decoding, we sample only from those captions in the decoded beam. Empirically, we have observed when decoding using beam search that the resulting beam typically contains at least one very high scoring caption – although frequently this caption does not have the

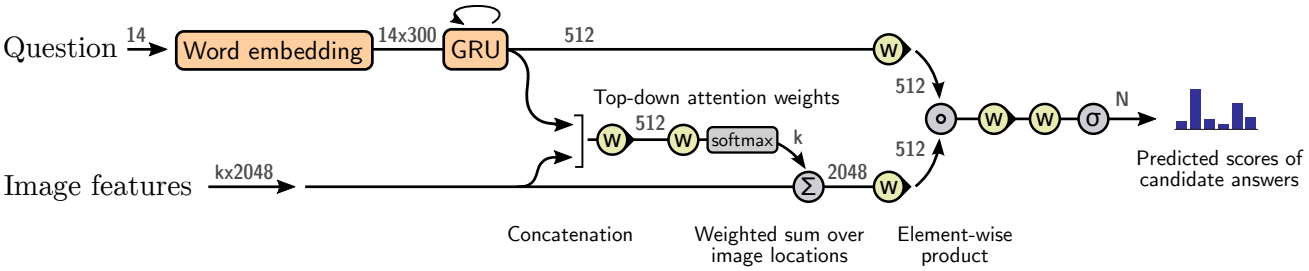


Figure 4. Overview of the proposed VQA model. A deep neural network implements a joint embedding of the question and image features $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_k\}$. These features can be defined as the spatial output of a CNN, or following our approach, generated using bottom-up attention. Output is generated by a multi-label classifier operating over a fixed set of candidate answers. Gray numbers indicate the dimensions of the vector representations between layers. Yellow elements use learned parameters.

highest log-probability of the set. In contrast, we observe that very few unrestricted caption samples score higher than the greedily-decoded caption. Using this approach, we complete CIDEr optimization in a single epoch.

3.3. VQA Model

Given a set of spatial image features V, our proposed VQA model also uses a 'soft' top-down attention mechanism to weight each feature, using the question representation as context. As illustrated in Figure 4, the proposed model implements the well-known joint multimodal embedding of the question and the image, followed by a prediction of regression of scores over a set of candidate answers. This approach has been the basis of numerous previous models [16, 20, 39]. However, as with our captioning model, implementation decisions are important to ensure that this relatively simple model delivers high performance.

The learned non-linear transformations within the network are implemented with gated hyperbolic tangent activations [7]. These are a special case of highway networks [37] that have shown a strong empirical advantage over traditional ReLU or tanh layers. Each of our 'gated tanh' layers implements a function $f_a : \boldsymbol{x} \in \mathbb{R}^m \rightarrow \boldsymbol{y} \in \mathbb{R}^n$ with parameters $a = \{W, W', b, b'\}$ defined as follows:

$$\tilde{\boldsymbol{y}} = \tanh\left(W\boldsymbol{x} + \boldsymbol{b}\right) \tag{12}$$

$$\boldsymbol{g} = \sigma(W'\boldsymbol{x} + \boldsymbol{b}') \tag{13}$$

$$\boldsymbol{y} = \tilde{\boldsymbol{y}} \circ \boldsymbol{g} \tag{14}$$

where σ is the sigmoid activation function, $W, W' \in \mathbb{R}^{n \times m}$ are learned weights, $\boldsymbol{b}, \boldsymbol{b}' \in \mathbb{R}^n$ are learned biases, and $\circ$ is the Hadamard (element-wise) product. The vector $\boldsymbol{g}$ acts multiplicatively as a gate on the intermediate activation $\tilde{\boldsymbol{y}}$.

Our proposed approach first encodes each question as the hidden state $\boldsymbol{q}$ of a gated recurrent unit [5] (GRU), with each input word represented using a learned word embedding. Similar to Equation 3, given the output $\boldsymbol{q}$ of the GRU, we generate an unnormalized attention weight a_i for each of the k image features $\boldsymbol{v}_i$ as follows:

$$a_i = \boldsymbol{w}_a^T f_a([\boldsymbol{v}_i, \boldsymbol{q}]) \tag{15}$$

where $\boldsymbol{w}_a^T$ is a learned parameter vector. Equation 4 and Equation 5 (neglecting subscripts t) are used to calculate the normalized attention weight and the attended image feature $\hat{\boldsymbol{v}}$. The distribution over possible output responses y is given by:

$$\boldsymbol{h} = f_q(\boldsymbol{q}) \circ f_v(\hat{\boldsymbol{v}}) \tag{16}$$

$$p(y) = \sigma(W_o\, f_o(\boldsymbol{h})) \tag{17}$$

Where $\boldsymbol{h}$ is a joint representation of the question and the image, and $W_o \in \mathbb{R}^{|\Sigma| \times M}$ are learned weights.

Due to space constraints, some important aspects of our VQA approach are not detailed here. For full specifics of the VQA model including a detailed exploration of architectures and hyperparameters, refer to Teney et al. [38].

4. Evaluation

4.1. Datasets

4.1.1 Visual Genome Dataset

We use the Visual Genome [21] dataset to pretrain our bottom-up attention model, and for data augmentation when training our VQA model. The dataset contains 108K images densely annotated with scene graphs containing objects, attributes and relationships, as well as 1.7M visual question answers.

For pretraining the bottom-up attention model, we use only the object and attribute data. We reserve 5K images for validation, and 5K images for future testing, treating the remaining 98K images as training data. As approximately 51K Visual Genome images are also found in the MSCOCO captions dataset [23], we are careful to avoid contamination of our MSCOCO validation and test sets. We ensure that any images found in both datasets are contained in the same split in both datasets.

	Cross-Entropy Loss						CIDEr Optimization					
	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
SCST:Att2in [34]	-	31.3	26.0	54.3	101.3	-	-	33.3	26.3	55.3	111.4	-
SCST:Att2all [34]	-	30.0	25.9	53.4	99.4	-	-	34.2	26.7	55.7	114.0	-
Ours: ResNet	74.5	33.4	26.1	54.4	105.4	19.2	76.6	34.0	26.5	54.9	111.1	20.2
Ours: Up-Down	**77.2**	**36.2**	**27.0**	**56.4**	**113.5**	**20.3**	**79.8**	**36.3**	**27.7**	**56.9**	**120.1**	**21.4**
Relative Improvement	4%	8%	3%	4%	8%	6%	4%	7%	5%	4%	8%	6%

Table 1. Single-model image captioning performance on the MSCOCO Karpathy test split. Our baseline ResNet model obtains similar results to SCST [34], the existing state-of-the-art on this test set. Illustrating the contribution of bottom-up attention, our Up-Down model achieves significant (3–8%) relative gains across all metrics regardless of whether cross-entropy loss or CIDEr optimization is used.

	Cross-Entropy Loss							CIDEr Optimization						
	SPICE	Objects	Attributes	Relations	Color	Count	Size	SPICE	Objects	Attributes	Relations	Color	Count	Size
Ours: ResNet	19.2	35.4	8.6	5.3	12.2	4.1	3.9	20.2	37.0	9.2	6.1	10.6	12.0	**4.3**
Ours: Up-Down	**20.3**	**37.1**	**9.2**	**5.8**	**12.7**	**6.5**	**4.5**	**21.4**	**39.1**	**10.0**	**6.5**	**11.4**	**18.4**	3.2

Table 2. Breakdown of SPICE F-scores over various subcategories on the MSCOCO Karpathy test split. Our Up-Down model outperforms the ResNet baseline at identifying objects, as well as detecting object attributes and the relations between objects.

As the object and attribute annotations consist of freely annotated strings, rather than classes, we perform extensive cleaning and filtering of the training data. Starting from 2,000 object classes and 500 attribute classes, we manually remove abstract classes that exhibit poor detection performance in initial experiments. Our final training set contains 1,600 object classes and 400 attribute classes. Note that we do not merge or remove overlapping classes (e.g. 'person', 'man', 'guy'), classes with both singular and plural versions (e.g. 'tree', 'trees') and classes that are difficult to precisely localize (e.g. 'sky', 'grass', 'buildings').

When training the VQA model, we augment the VQA v2.0 training data with Visual Genome question and answer pairs provided the correct answer is present in model's answer vocabulary. This represents about 30% of the available data, or 485K questions.

4.1.2 Microsoft COCO Dataset

To evaluate our proposed captioning model, we use the MSCOCO 2014 captions dataset [23]. For validation of model hyperparameters and offline testing, we use the 'Karpathy' splits [19] that have been used extensively for reporting results in prior work. This split contains 113,287 training images with five captions each, and 5K images respectively for validation and testing. Our MSCOCO test server submission is trained on the entire MSCOCO 2014 training and validation set (123K images).

We follow standard practice and perform only minimal text pre-processing, converting all sentences to lower case, tokenizing on white space, and filtering words that do not occur at least five times, resulting in a model vocabulary of 10,010 words. To evaluate caption quality, we use the standard automatic evaluation metrics, namely SPICE [1], CIDEr [43], METEOR [8], ROUGE-L [22] and BLEU [29].

4.1.3 VQA v2.0 Dataset

To evaluate our proposed VQA model, we use the recently introduced VQA v2.0 dataset [12], which attempts to minimize the effectiveness of learning dataset priors by balancing the answers to each question. The dataset, which was used as the basis of the 2017 VQA Challenge[2], contains 1.1M questions with 11.1M answers relating to MSCOCO images.

We perform standard question text preprocessing and tokenization. Questions are trimmed to a maximum of 14 words for computational efficiency. The set of candidate answers is restricted to correct answers in the training set that appear more than 8 times, resulting in an output vocabulary size of 3,129. Our VQA test server submissions are trained on the training and validation sets plus additional questions and answers from Visual Genome. To evaluate answer quality, we report accuracies using the standard VQA metric [2], which takes into account the occasional disagreement between annotators for the ground truth answers.

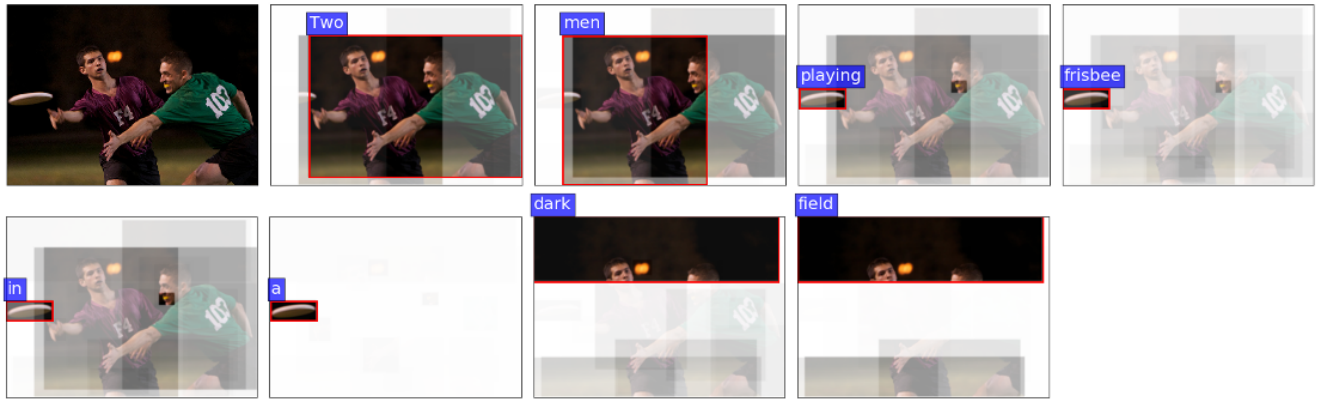
4.2. ResNet Baseline

To quantify the impact of bottom-up attention, in both our captioning and VQA experiments we evaluate our full

[2]http://www.visualqa.org/challenge.html

	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE-L		CIDEr		SPICE	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Review Net [48]	72.0	90.0	55.0	81.2	41.4	70.5	31.3	59.7	25.6	34.7	53.3	68.6	96.5	96.9	18.5	64.9
Adaptive [27]	74.8	92.0	58.4	84.5	44.4	74.4	33.6	63.7	26.4	35.9	55.0	70.5	104.2	105.9	19.7	67.3
PG-BCMR [24]	75.4	-	59.1	-	44.5	-	33.2	-	25.7	-	55	-	101.3	-	-	-
SCST:Att2all [34]	78.1	93.7	61.9	86.0	47.0	75.9	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.7	20.7	68.9
LSTM-A_3 [49]	78.7	93.7	62.7	86.7	47.6	76.5	35.6	65.2	27	35.4	56.4	70.5	116	118	-	-
Ours: Up-Down	**80.2**	**95.2**	**64.1**	**88.8**	**49.1**	**79.4**	**36.9**	**68.5**	**27.6**	**36.7**	**57.1**	**72.4**	**117.9**	**120.5**	**21.5**	**71.5**

Table 3. Highest ranking published image captioning results on the online MSCOCO test server. Our submission, an ensemble of 4 models optimized for CIDEr with different initializations, outperforms previously published work on all reported metrics. At the time of submission (18 July 2017), we also outperformed all unpublished test server submissions.



Two men playing frisbee in a dark field.

Figure 5. Example of a generated caption showing attended image regions. For each generated word, we visualize the attention weights on individual pixels, outlining the region with the maximum attention weight in red. Avoiding the conventional trade-off between coarse and fine levels of detail, our model focuses on both closely-cropped details, such as the frisbee and the green player's mouthguard when generating the word 'playing', as well as large regions, such as the night sky when generating the word 'dark'.

model (*Up-Down*) against prior work as well as an ablated baseline. In each case, the baseline (*ResNet*), uses a ResNet [13] CNN pretrained on ImageNet [35] to encode each image in place of the bottom-up attention mechanism.

In image captioning experiments, similarly to previous work [34] we encode the full-sized input image with the final convolutional layer of Resnet-101, and use bilinear interpolation to resize the output to a fixed size spatial representation of 10×10. This is equivalent to the maximum number of spatial regions used in our full model. In VQA experiments, we encode the resized input image with ResNet-200 [14]. In separate experiments we use evaluate the effect of varying the size of the spatial output from its original size of 14×14, to 7×7 (using bilinear interpolation) and 1×1 (i.e., mean pooling without attention).

4.3. Image Captioning Results

In Table 1 we report the performance of our full model and the ResNet baseline in comparison to the existing state-of-the-art Self-critical Sequence Training [34] (SCST) approach on the test portion of the Karpathy splits. For fair comparison, results are reported for models trained with both standard cross-entropy loss, and models optimized for CIDEr score. Note that the SCST approach uses ResNet-101 encoding of full images, similar to our ResNet baseline. All results are reported for a single model with no fine-tuning of the input ResNet / R-CNN model. However, the SCST results are selected from the best of four random initializations, while our results are outcomes from a single initialization.

Relative to the SCST models, our ResNet baseline obtains slightly better performance under cross-entropy loss, and slightly worse performance when optimized for CIDEr score. After incorporating bottom-up attention, our full Up-Down model shows significant improvements across all metrics regardless of whether cross-entropy loss or CIDEr optimization is used. Using just a single model, we obtain the best reported results for the Karpathy test split. As illustrated in Table 2, the contribution from bottom-up attention is broadly based, illustrated by improved performance in

	Yes/No	Number	Other	Overall
Ours: ResNet (1×1)	76.0	36.5	46.8	56.3
Ours: ResNet (14×14)	76.6	36.2	49.5	57.9
Ours: ResNet (7×7)	77.6	37.7	51.5	59.4
Ours: Up-Down	**80.3**	**42.8**	**55.8**	**63.2**
Relative Improvement	3%	14%	8%	6%

Table 4. Single-model performance on the VQA v2.0 validation set. The use of bottom-up attention in the Up-Down model provides a significant improvement over the best ResNet baseline across all question types, even though the ResNet baselines use almost twice as many convolutional layers.

	Yes/No	Number	Other	Overall
Prior [12]	61.20	0.36	1.17	25.98
Language-only [12]	67.01	31.55	27.37	44.26
d-LSTM+n-I [26, 12]	73.46	35.18	41.83	54.22
MCB [11, 12]	78.82	38.28	53.36	62.27
UPMC-LIP6	82.07	41.06	57.12	65.71
Athena	82.50	44.19	59.97	67.59
HDU-USYD-UNCC	84.50	45.39	59.01	68.09
Ours: Up-Down	**86.60**	**48.64**	**61.15**	**70.34**

Table 5. VQA v2.0 test-standard server accuracy as at 8 August 2017, ranking our submission against published and unpublished work for each question type. Our approach, an ensemble of 30 models, outperforms all other leaderboard entries.

terms of identifying objects, object attributes and also the relationships between objects.

Table 3 reports the performance of 4 ensembled models trained with CIDEr optimization on the official MSCOCO evaluation server, along with the highest ranking previously published results. At the time of submission (18 July 2017), we outperform all other test server submissions on all reported evaluation metrics.

4.4. VQA Results

In Table 4 we report the single model performance of our full Up-Down VQA model relative to several ResNet baselines on the VQA v2.0 validation set. The addition of bottom-up attention provides a significant improvement over the best ResNet baseline across all question types, even though the ResNet baseline uses approximately twice as many convolutional layers. Table 5 reports the performance of 30 ensembled models on the official VQA 2.0 test-standard evaluation server, along with the previously published baseline results and the highest ranking other entries. At the time of submission (8 August 2017), we outperform all other test server submissions. Our submission also achieved first place in the 2017 VQA Challenge.

Question: What room are they in? Answer: kitchen

Figure 6. VQA example illustrating attention output. Given the question 'What room are they in?', the model focuses on the stove-top, generating the answer 'kitchen'.

4.5. Qualitative Analysis

To help qualitatively evaluate our attention methodology, in Figure 5 we visualize the attended image regions for different words generated by our Up-Down captioning model. As indicated by this example, our approach is equally capable of focusing on fine details or large image regions. This capability arises because the attention candidates in our model consist of many overlapping regions with varying scales and aspect ratios – each aligned to an object, several related objects, or an otherwise salient image patch.

Unlike conventional approaches, when a candidate attention region corresponds to an object, or several related objects, all the visual concepts associated with those objects appear to be spatially co-located – and are processed together. In other words, our approach is able to consider all of the information pertaining to an object at once. This is also a natural way for attention to be implemented. In the human visual system, the problem of integrating the separate features of objects in the correct combinations is known as the feature binding problem, and experiments suggest that attention plays a central role in the solution [41, 40]. We include an example of VQA attention in Figure 6.

5. Conclusion

We present a novel combined bottom-up and top-down visual attention mechanism. Our approach enables attention to be calculated more naturally at the level of objects and other salient regions. Applying this approach to image captioning and visual question answering, we achieve state-of-the-art results in both tasks, while improving the interpretability of the resulting attention weights.

At a high level, our work more closely unifies tasks involving visual and linguistic understanding with recent progress in object detection. While this suggests several directions for future research, the immediate benefits of our approach may be captured by simply replacing pretrained CNN features with pretrained bottom-up attention features.

Acknowledgements

This research is partially supported by an Australian Government Research Training Program (RTP) Scholarship, by the Australian Research Council Centre of Excellence for Robotic Vision (project number CE140100016), by a Google award through the Natural Language Understanding Focused Program, and under the Australian Research Councils Discovery Projects funding scheme (project number DP160102156).

References

[1] P. Anderson, B. Fernando, M. Johnson, and S. Gould. SPICE: Semantic propositional image caption evaluation. In *ECCV*, 2016. 6

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *ICCV*, 2015. 6

[3] T. J. Buschman and E. K. Miller. Top-down versus bottom-up control of attention in the prefrontal and posterior parietal cortices. *Science*, 315(5820):1860–1862, 2007. 1

[4] X. Chen, T.-Y. L. Hao Fang, R. Vedantam, S. Gupta, P. Dollar, and C. L. Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 1

[5] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, 2014. 5

[6] M. Corbetta and G. L. Shulman. Control of goal-directed and stimulus-driven attention in the brain. *Nature Reviews Neuroscience*, 3(3):201–215, 2002. 1

[7] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. *arXiv preprint arXiv:1612.08083*, 2016. 5

[8] M. Denkowski and A. Lavie. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014. 6

[9] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 3

[10] R. Egly, J. Driver, and R. D. Rafal. Shifting visual attention between objects and locations: evidence from normal and parietal lesion subjects. *Journal of Experimental Psychology: General*, 123(2):161, 1994. 2

[11] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, 2016. 1, 2, 8

[12] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *CVPR*, 2017. 1, 6, 8

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 7

[14] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. *arXiv preprint arXiv:1603.05027*, 2016. 7

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997. 3

[16] A. Jabri, A. Joulin, and L. van der Maaten. Revisiting visual question answering baselines. *arXiv preprint arXiv:1606.08390*, 2016. 2, 5

[17] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015. 2

[18] J. Jin, K. Fu, R. Cui, F. Sha, and C. Zhang. Aligning where to see and what to tell: image caption with region-based attention and scene factorization. *arXiv preprint arXiv:1506.06272*, 2015. 2

[19] A. Karpathy and F.-F. Li. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 6

[20] V. Kazemi and A. Elqursh. Show, ask, attend, and answer: A strong baseline for visual question answering. *arXiv preprint arXiv:1704.03162*, 2017. 2, 5

[21] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016. 3, 5, 10

[22] C. Lin. Rouge: a package for automatic evaluation of summaries. In *ACL Workshop*, 2004. 6

[23] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 5, 6

[24] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy. Improved image captioning via policy gradient optimization of spider. In *ICCV*, 2017. 7

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. *arXiv preprint arXiv:1512.02325*, 2015. 2

[26] J. Lu, X. Lin, D. Batra, and D. Parikh. Deeper lstm and normalized cnn visual question answering model. https://github.com/VT-vision-lab/VQA_LSTM_CNN, 2015. 8

[27] J. Lu, C. Xiong, D. Parikh, and R. Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*, 2017. 1, 2, 3, 7

[28] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *NIPS*, 2016. 1, 2

[29] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 6

[30] M. Pedersoli, T. Lucas, C. Schmid, and J. Verbeek. Areas of attention for image captioning. In *ICCV*, 2017. 2

[31] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global Vectors for Word Representation. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2014. 10

[32] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 2

[33] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2

[34] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017. 1, 2, 3, 4, 6, 7

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 2, 3, 7

[36] B. J. Scholl. Objects and attention: The state of the art. *Cognition*, 80(1):1–46, 2001. 2

[37] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387v1*, 2015. 5

[38] D. Teney, P. Anderson, X. He, and A. van den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *CVPR*, 2018. 5, 10

[39] D. Teney and A. van den Hengel. Zero-shot visual question answering. *arXiv preprint arXiv:1611.05546*, 2016. 5

[40] A. Treisman. Perceptual grouping and attention in visual search for features and for objects. *Journal of Experimental Psychology: Human Perception and Performance*, 8(2):194, 1982. 8

[41] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12:97–136, 1980. 8

[42] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 2013. 2

[43] R. Vedantam, C. L. Zitnick, and D. Parikh. CIDEr: Consensus-based image description evaluation. In *CVPR*, 2015. 4, 6

[44] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3-4):229–256, May 1992. 4

[45] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*, 2016. 1, 2

[46] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 1, 2, 3

[47] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016. 1, 2

[48] Z. Yang, Y. Yuan, Y. Wu, R. Salakhutdinov, and W. W. Cohen. Review networks for caption generation. In *NIPS*, 2016. 1, 2, 7

[49] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei. Boosting image captioning with attributes. In *ICCV*, 2017. 7

[50] M. D. Zeiler. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. 10

[51] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. In *CVPR*, 2016. 1, 2

[52] L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 2

SUPPLEMENTARY MATERIALS

6. Implementation Details

6.1. Bottom-Up Attention Model

Our bottom-up attention Faster R-CNN implementation uses an IoU threshold of 0.7 for region proposal suppression, and 0.3 for object class suppression. To select salient image regions, a class detection confidence threshold of 0.2 is used, allowing the number of regions per image k to vary with the complexity of the image, up to a maximum of 100. However, in initial experiments we find that simply selecting the top 36 features in each image works almost as well in both downstream tasks. Since Visual Genome [21] contains a relatively large number of annotations per image, the model is relatively intensive to train. Using 8 Nvidia M40 GPUs, we take around 5 days to complete 380K training iterations, although we suspect that faster training regimes could also be effective.

6.2. Captioning Model

In the captioning model, we set the number of hidden units M in each LSTM to 1,000, the number of hidden units H in the attention layer to 512, and the size of the input word embedding E to 1,000. In training, we use a simple learning rate schedule, beginning with a learning rate of 0.01 which is reduced to zero on a straight-line basis over 60K iterations using a batch size of 100 and a momentum parameter of 0.9. Training using two Nvidia Titan X GPUs takes around 9 hours (including less than one hour for CIDEr optimization). During optimization and decoding we use a beam size of 5. When decoding we also enforce the constraint that a single word cannot be predicted twice in a row. Note that in both our captioning and VQA models, image features are fixed and not finetuned.

6.3. VQA Model

In the VQA model, we use 300 dimension word embeddings, initialized with pretrained GloVe vectors [31], and we use hidden states of dimension 512. We train the VQA model using AdaDelta [50] and regularize with early stopping. The training of the model takes in the order of 12–18 hours on a single Nvidia K40 GPU. Refer to Teney et al. [38] for further details of the VQA model implementation.

7. Additional Examples

In Figure 7 we qualitatively compare attention methodologies for image caption generation, by illustrating attention weights for the ResNet baseline and our full Up-Down model on the same image. The baseline ResNet model hallucinates a toilet and therefore generates a poor quality caption. In contrast, our Up-Down model correctly identifies the couch, despite the novel scene composition. Additional examples of generated captions can be found in Figures 8 and 9. Additional visual question answering examples can be found in Figures 10 and 11.

Figure 7. Qualitative differences between attention methodologies in caption generation. For each generated word, we visualize the attended image region, outlining the region with the maximum attention weight in red. The selected image is unusual because it depicts a bathroom containing a couch but no toilet. Nevertheless, our baseline ResNet model (top) hallucinates a toilet, presumably from language priors, and therefore generates a poor quality caption. In contrast, our Up-Down model (bottom) clearly identifies the out-of-context couch, generating a correct caption while also providing more interpretable attention weights.

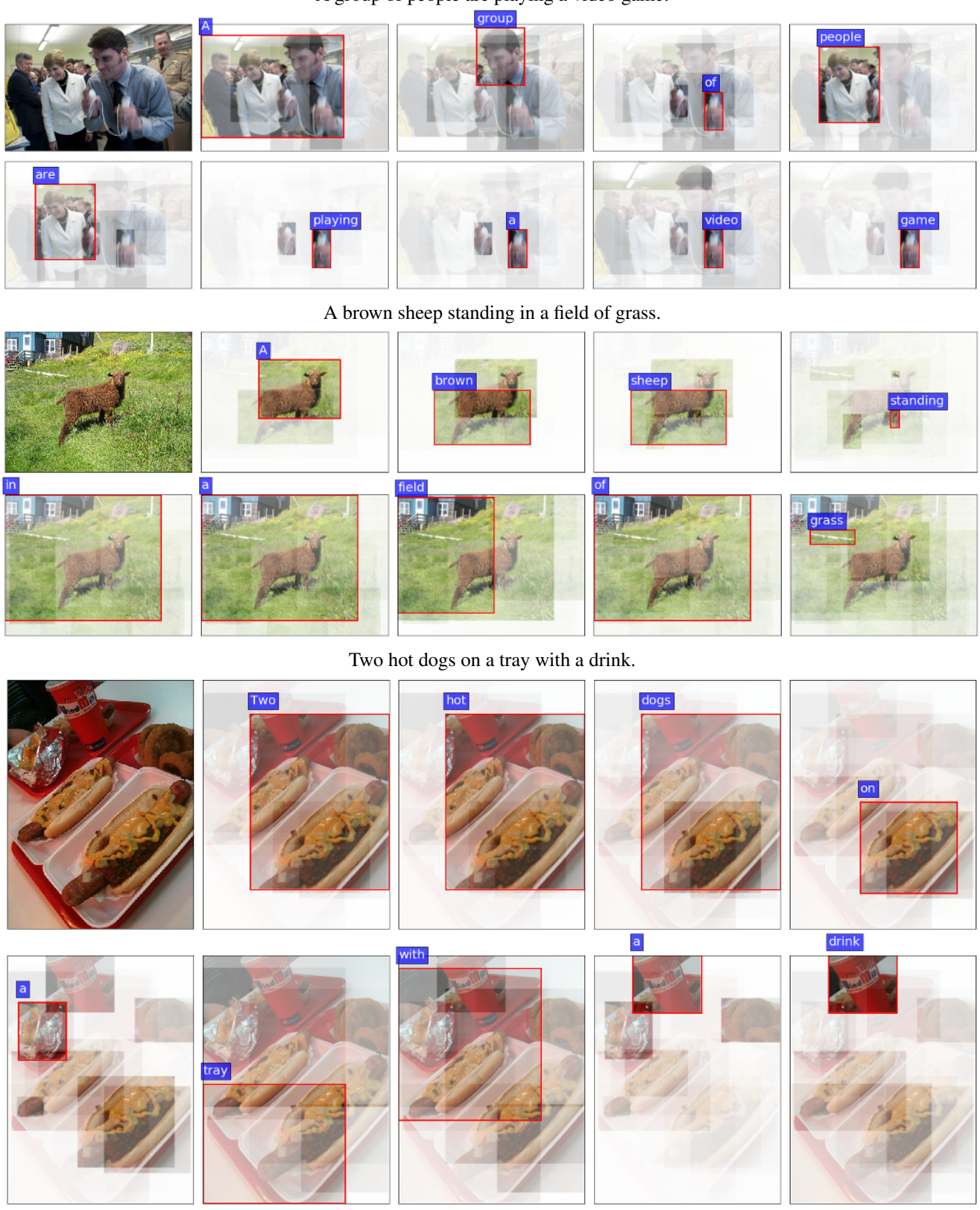


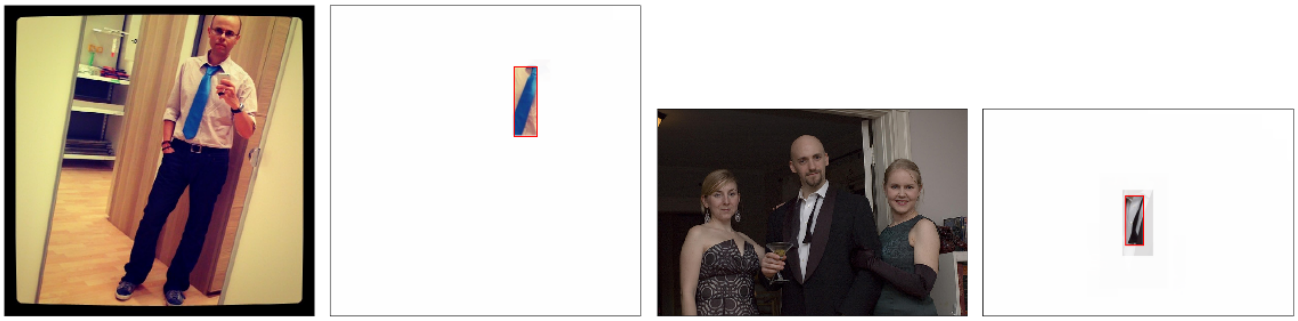
Figure 8. Examples of generated captions showing attended image regions. Attention is given to fine details, such as: (1) the man's hands holding the game controllers in the top image, and (2) the sheep's legs when generating the word 'standing' in the middle image. Our approach can avoid the trade-off between coarse and fine levels of detail.

Figure 9. Further examples of generated captions showing attended image regions. The first example suggests an understanding of spatial relationships when generating the word 'together'. The middle image demonstrates the successful captioning of a compositionally novel scene. The bottom example is a failure case. The dog's pose is mistaken for laying, rather than jumping – possibly due to poor salient region cropping that misses the dog's head and feet.

Question: What color is illuminated on the traffic light? Answer left: green. Answer right: red.

Question: What is the man holding? Answer left: phone. Answer right: controller.

Question: What color is his tie? Answer left: blue. Answer right: black.



Question: What sport is shown? Answer left: frisbee. Answer right: skateboarding.

Question: Is this the handlebar of a motorcycle? Answer left: yes. Answer right: no.

Figure 10. Further examples of successful visual question answering results, showing attended image regions.

Question: What is the name of the realty company? Answer left: none. Answer right: none.

Question: What is the bus number? Answer left: 2. Answer right: 23.

Question: How many cones have reflective tape? Answer left: 2. Answer right: 1.

Question: How many oranges are on pedestals? Answer left: 2. Answer right: 2.

Figure 11. Examples of visual question answering (VQA) failure cases. Although our simple VQA model has limited reading and counting capabilities, the attention maps are often correctly focused.