Self-critical Sequence Training for Image Captioning

Steven J. Rennie, Etienne Marcheret[1], Youssef Mroueh, Jerret Ross and Vaibhava Goel[1]
Watson Multimodal Algorithms and Engines Group
IBM T.J. Watson Research Center, NY, USA
{sjrennie, mroueh, rossja}@us.ibm.com, {etiennemarcheret, vaibhavagoel}@gmail.com

Abstract

Recently it has been shown that policy-gradient methods for reinforcement learning can be utilized to train deep end-to-end systems directly on non-differentiable metrics for the task at hand. In this paper we consider the problem of optimizing image captioning systems using reinforcement learning, and show that by carefully optimizing our systems using the test metrics of the MSCOCO task, significant gains in performance can be realized. Our systems are built using a new optimization approach that we call self-critical sequence training (SCST). SCST is a form of the popular REINFORCE algorithm that, rather than estimating a "baseline" to normalize the rewards and reduce variance, utilizes the output of its own test-time inference algorithm to normalize the rewards it experiences. Using this approach, estimating the reward signal (as actor-critic methods must do) and estimating normalization (as REINFORCE algorithms typically do) is avoided, while at the same time harmonizing the model with respect to its test-time inference procedure. Empirically we find that directly optimizing the CIDEr metric with SCST and greedy decoding at test-time is highly effective. Our results on the MSCOCO evaluation sever establish a new state-of-the-art on the task, improving the best result in terms of CIDEr from 104.9 to 114.7.

1. Introduction

Image captioning aims at generating a natural language description of an image. Open domain captioning is a very challenging task, as it requires a fine-grained understanding of the global and the local entities in an image, as well as their attributes and relationships. The recently released MSCOCO challenge [1] provides a new, larger scale platform for evaluating image captioning systems, complete with an evaluation server for benchmarking competing methods. Deep learning approaches to sequence modeling have yielded impressive results on the task, dominating the task leaderboard. Inspired by the recently introduced encoder/decoder paradigm for machine translation using recurrent neural networks (RNNs) [2], [3], and [4] use a deep convolutional neural network (CNN) to encode the input image, and a Long Short Term Memory (LSTM) [5] RNN decoder to generate the output caption. These systems are trained end-to-end using back-propagation, and have achieved state-of-the-art results on MSCOCO. More recently in [6], the use of spatial attention mechanisms on CNN layers to incorporate visual context—which implicitly conditions on the text generated so far—was incorporated into the generation process. It has been shown and we have qualitatively observed that captioning systems that utilize attention mechanisms lead to better generalization, as these models can compose novel text descriptions based on the recognition of the global and local entities that comprise images.

As discussed in [7], deep generative models for text are typically trained to maximize the likelihood of the next ground-truth word given the previous *ground-truth* word using back-propagation. This approach has been called "Teacher-Forcing" [8]. However, this approach creates a mismatch between training and testing, since at test-time the model uses the previously generated words from the model distribution to predict the next word. This *exposure bias* [7], results in error accumulation during generation at test time, since the model has never been exposed to its own predictions.

Several approaches to overcoming the exposure bias problem described above have recently been proposed. In [8] they show that feeding back the model's own predictions and slowly increasing the feedback probability p during training leads to significantly better test-time performance. Another line of work proposes "Professor-Forcing" [9], a technique that uses adversarial training to encourage the dynamics of the recurrent network to be the same when training conditioned on ground truth previous words and when sampling freely from the network.

While sequence models are usually trained using the

[1]Authors Etienne Marcheret and Vaibhava Goel were at IBM while the work was being completed.

cross entropy loss, they are typically evaluated at test time using discrete and non-differentiable NLP metrics such as BLEU [10], ROUGE [11], METEOR [12] or CIDEr [13]. Ideally sequence models for image captioning should be trained to avoid exposure bias *and* directly optimize metrics for the task at hand.

Recently it has been shown that both the exposure bias and non-differentiable task metric issues can be addressed by incorporating techniques from Reinforcement Learning (RL) [14]. Specifically in [7], Ranzato et al. use the REINFORCE algorithm [15] to directly optimize non-differentiable, sequence-based test metrics, and overcome both issues. REINFORCE as we will describe, allows one to optimize the gradient of the expected reward by sampling from the model during training, and treating those samples as ground-truth labels (that are re-weighted by the reward they deliver). The major limitation of the approach is that the expected gradient computed using mini-batches under REINFORCE typically exhibit high variance, and without proper context-dependent normalization, is typically unstable. The recent discovery that REINFORCE with proper bias correction using learned "baselines" is effective has led to a flurry of work in applying REINFORCE to problems in RL, supervised learning, and variational inference [16, 17, 18]. Actor-critic methods [14] , which instead train a second "critic" network to provide an *estimate* of the value of each generated word given the policy of an actor network, have also been investigated for sequence problems recently [19]. These techniques overcome the need to sample from the policy's (actor's) action space, which can be enormous, at the expense of estimating future rewards, and training multiple networks based on one another's outputs, which as [19] explore, can also be unstable.

In this paper we present a new approach to sequence training which we call self-critical sequence training (SCST), and demonstrate that SCST can improve the performance of image captioning systems dramatically. SCST is a REINFORCE algorithm that, rather than estimating the reward signal, or how the reward signal should be normalized, utilizes the output of its own test-time inference algorithm to normalize the rewards it experiences. As a result, only samples from the model that outperform the current test-time system are given positive weight, and inferior samples are suppressed. Using SCST, attempting to estimate the reward signal, as actor-critic methods must do, and estimating normalization, as REINFORCE algorithms must do, is avoided, while at the same time harmonizing the model with respect to its test-time inference procedure. Empirically we find that directly optimizing the CIDEr metric with SCST and greedy decoding at test-time is highly effective. Our results on the MSCOCO evaluation sever establish a new state-of-the-art on the task, improving the best result in terms of CIDEr from 104.9 to 114.7.

2. Captioning Models

In this section we describe the recurrent models that we use for caption generation.

FC models. Similarly to [3, 4], we first encode the input image F using a deep CNN, and then embed it through a linear projection W_I . Words are represented with one hot vectors that are embedded with a linear embedding E that has the same output dimension as W_I. The beginning of each sentence is marked with a special BOS token, and the end with an EOS token. Under the model, words are generated and then fed back into the LSTM, with the image treated as the first word $W_I CNN(F)$. The following updates for the hidden units and cells of an LSTM define the model [5]:

$$\begin{aligned}
x_t &= E 1_{w_{t-1}} \text{ for } t > 1, x_1 = W_I CNN(F) \\
i_t &= \sigma\left(W_{ix} x_t + W_{ih} h_{t-1} + b_i\right) \quad \text{(Input Gate)} \\
f_t &= \sigma\left(W_{fx} x_t + W_{fh} h_{t-1} + b_f\right) \quad \text{(Forget Gate)} \\
o_t &= \sigma\left(W_{ox} x_t + W_{oh} h_{t-1} + b_o\right) \quad \text{(Output Gate)} \\
c_t &= i_t \odot \phi(W_{zx}^{\otimes} x_t + W_{zh}^{\otimes} h_{t-1} + b_z^{\otimes}) + f_t \odot c_{t-1} \\
h_t &= o_t \odot \tanh(c_t) \\
s_t &= W_s h_t,
\end{aligned}$$

where ϕ is a maxout non-linearity with 2 units ($\otimes$ denotes the units) and σ is the sigmoid function. We initialize h_0 and c_0 to zero. The LSTM outputs a distribution over the next word w_t using the softmax function:

$$w_t \sim \text{softmax}\left(\text{s}_\text{t}\right) \tag{1}$$

In our architecture, the hidden states and word and image embeddings have dimension 512. Let θ denote the parameters of the model. Traditionally the parameters θ are learned by maximizing the likelihood of the observed sequence. Specifically, given a target ground truth sequence $(w_1^*, \ldots, w_T^*)$, the objective is to minimize the cross entropy loss (XE):

$$L(\theta) = -\sum_{t=1}^{T} \log(p_\theta(w_t^* | w_1^*, \ldots w_{t-1}^*)), \tag{2}$$

where $p_\theta(w_t | w_1, \ldots w_{t-1})$ is given by the parametric model in Equation (1).

Attention Model (Att2in). Rather than utilizing a static, spatially pooled representation of the image, attention models dynamically re-weight the input spatial (CNN) features to focus on specific regions of the image at each time step. In this paper we consider a modification of the architecture of the attention model for captioning given in [6], and input the attention-derived image feature only to the cell node of the LSTM.

$$
\begin{aligned}
x_t &= E1_{w_{t-1}} \text{ for } t \geq 1 \; w_0 = BOS \\
i_t &= \sigma\left(W_{ix}x_t + W_{ih}h_{t-1} + b_i\right) \quad \text{(Input Gate)} \\
f_t &= \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + b_f\right) \quad \text{(Forget Gate)} \\
o_t &= \sigma\left(W_{ox}x_t + W_{oh}h_{t-1} + b_o\right) \quad \text{(Output Gate)} \\
c_t &= i_t \odot \phi(W^{\otimes}_{zx}x_t + W^{\otimes}_{zI}I_t + W^{\otimes}_{zh}h_{t-1} + b^{\otimes}_z) + f_t \odot c_{t-1} \\
h_t &= o_t \odot \tanh(c_t) \\
s_t &= W_s h_t,
\end{aligned}
$$

where I_t is the attention-derived image feature. This feature is derived as in [6] as follows: given CNN features at N locations $\{I_1, \dots I_N\}$, $I_t = \sum_{i=1}^{N} \alpha_t^i I_i$, where $\alpha_t = \text{softmax}(a_t + b_\alpha)$, and $a_t^i = W \tanh(W_{aI}I_i + W_{ah}h_{t-1} + b_a)$. In this work we set the dimension of W to 1×512, and set c_0 and h_0 to zero. Let θ denote the parameters of the model. Then $p_\theta(w_t|w_1, \dots w_{t-1})$ is again defined by (1). The parameters θ of attention models are also traditionally learned by optimizing the XE loss (2).

Attention Model (Att2all). The standard attention model presented in [6] also feeds then attention signal I_t as an input into all gates of the LSTM, and the output posterior. In our experiments feeding I_t to all gates in addition to the input did not boost performance, but feeding I_t to both the gates and the outputs resulted in significant gains when ADAM [20] was used.

3. Reinforcement Learning

Sequence Generation as an RL problem. As described in the previous section, captioning systems are traditionally trained using the cross entropy loss. To directly optimize NLP metrics and address the exposure bias issue, we can cast our generative models in the Reinforcement Learning terminology as in [7]. Our recurrent models (LSTMs) introduced above can be viewed as an "agent" that interacts with an external "environment" (words and image features). The parameters of the network, θ, define a policy p_θ, that results in an "action" that is the prediction of the next word. After each action, the agent (the LSTM) updates its internal "state" (cells and hidden states of the LSTM, attention weights etc). Upon generating the end-of-sequence (EOS) token, the agent observes a "reward" that is, for instance, the CIDEr score of the generated sentence—we denote this reward by r. The reward is computed by an evaluation metric by comparing the generated sequence to corresponding ground-truth sequences. The goal of training is to minimize the negative expected reward:

$$L(\theta) = -\mathbb{E}_{w^s \sim p_\theta}\left[r(w^s)\right], \tag{3}$$

where $w^s = (w_1^s, \dots w_T^s)$ and w_t^s is the word sampled from the model at the time step t. In practice $L(\theta)$ is typically estimated with a single sample from p_θ:

$$L(\theta) \approx -r(w^s), \; w^s \sim p_\theta.$$

Policy Gradient with REINFORCE. In order to compute the gradient $\nabla_\theta L(\theta)$, we use the REINFORCE algorithm [15](See also Chapter 13 in [14]). REINFORCE is based on the observation that the expected gradient of a non-differentiable reward function can be computed as follows:

$$\nabla_\theta L(\theta) = -\mathbb{E}_{w^s \sim p_\theta}\left[r(w^s)\nabla_\theta \log p_\theta(w^s)\right]. \tag{4}$$

In practice the expected gradient can be approximated using a single Monte-Carlo sample $w^s = (w_1^s \dots w_T^s)$ from p_θ, for each training example in the minibatch:

$$\nabla_\theta L(\theta) \approx -r(w^s)\nabla_\theta \log p_\theta(w^s).$$

REINFORCE with a Baseline. The policy gradient given by REINFORCE can be generalized to compute the reward associated with an action value *relative* to a reference reward or *baseline* b:

$$\nabla_\theta L(\theta) = -\mathbb{E}_{w^s \sim p_\theta}\left[(r(w^s) - b)\nabla_\theta \log p_\theta(w^s)\right]. \tag{5}$$

The baseline can be an arbitrary function, as long as it does not depend on the "action" w^s [14], since in this case:

$$
\begin{aligned}
\mathbb{E}_{w^s \sim p_\theta}\left[b\nabla_\theta \log p_\theta(w^s)\right] &= b\sum_{w_s} \nabla_\theta p_\theta(w^s) \\
&= b\nabla_\theta \sum_{w_s} p_\theta(w^s) \\
&= b\nabla_\theta 1 = 0.
\end{aligned} \tag{6}
$$

This shows that the baseline does not change the expected gradient, but importantly, it can reduce the variance of the gradient estimate. For each training case, we again approximate the expected gradient with a single sample $w^s \sim p_\theta$:

$$\nabla_\theta L(\theta) \approx -(r(w^s) - b)\nabla_\theta \log p_\theta(w^s). \tag{7}$$

Note that if b is function of θ or t as in [7], equation (6) still holds and $b(\theta)$ is a valid baseline.

Final Gradient Expression. Using the chain rule, and the parametric model of p_θ given in Section 2 we have:

$$\nabla_\theta L(\theta) = \sum_{t=1}^{T} \frac{\partial L(\theta)}{\partial s_t}\frac{\partial s_t}{\partial \theta},$$

where s_t is the input to the softmax function. Using REINFORCE with a baseline b the estimate of the gradient of $\frac{\partial L(\theta)}{\partial s_t}$ is given by [17]:

$$\frac{\partial L(\theta)}{\partial s_t} \approx (r(w^s) - b)(p_\theta(w_t|h_t) - 1_{w_t^s}). \tag{8}$$

4. Self-critical sequence training (SCST)

The central idea of the self-critical sequence training (SCST) approach is to baseline the REINFORCE algorithm with the reward obtained by the current model under the inference algorithm used at test time. The gradient of the negative reward of a sample w^s from the model w.r.t. to the softmax activations at time-step t then becomes:

$$\frac{\partial L(\theta)}{\partial s_t} = (r(w^s) - r(\hat{w}))(p_\theta(w_t|h_t) - 1_{w^s_t}). \quad (9)$$

where $r(\hat{w})$ again is the reward obtained by the current model under the inference algorithm used at test time. Accordingly, samples from the model that return higher reward than $\hat{w}$ will be "pushed up", or increased in probability, while samples which result in lower reward will be suppressed. Like MIXER [7], SCST has all the advantages of REINFORCE algorithms, as it directly optimizes the true, sequence-level, evaluation metric, but avoids the usual scenario of having to learn a (context-dependent) *estimate* of expected future rewards as a baseline. In practice we have found that SCST has much lower variance, and can be more effectively trained on mini-batches of samples using SGD. Since the SCST baseline is based on the test-time estimate under the current model, SCST is forced to improve the performance of the model under the inference algorithm used at test time. This encourages training/test time consistency like the maximum likelihood-based approaches "Data as Demonstrator" [8], "Professor Forcing" [9], and E2E [7], but importantly, it can directly optimize sequence metrics. Finally, SCST is self-critical, and so avoids all the inherent training difficulties associated with actor-critic methods, where a second "critic" network must be trained to estimate value functions, and the actor must be trained on *estimated* value functions rather than actual rewards.

In this paper we focus on scenario of greedy decoding, where:

$$\hat{w}_t = \arg\max_{w_t} p(w_t \mid h_t) \quad (10)$$

This choice, depicted in Figure 1, minimizes the impact of baselining with the test-time inference algorithm on training time, since it requires only one additional forward pass, and trains the system to be optimized for fast, greedy decoding at test-time.

Generalizations. The basic SCST approach described above can be generalized in several ways.

One generalization is to condition the baseline on what has been generated (i.e. sampled) so far, which makes the baseline *word-dependent*, and further reduces the variance of the reward signal by making it dependent only on *future rewards*. This is achieved by baselining the reward for word w^s_t at timestep t with the reward obtained by the word sequence $\bar{w} = \{w^s_{1:t-1}, \hat{w}_{t:T}\}$, which is generated by sampling tokens for timesteps $1 : t-1$, and then executing the inference algorithm to complete the sequence. The resulting reward signal, $r(w^s) - r(\bar{w})$, is a baselined future reward (advantage) signal that conditions on both the input image and the sequence $w^s_{1:t-1}$, and remains unbiased. We call this variant *time-dependent SCST (TD-SCST)*.

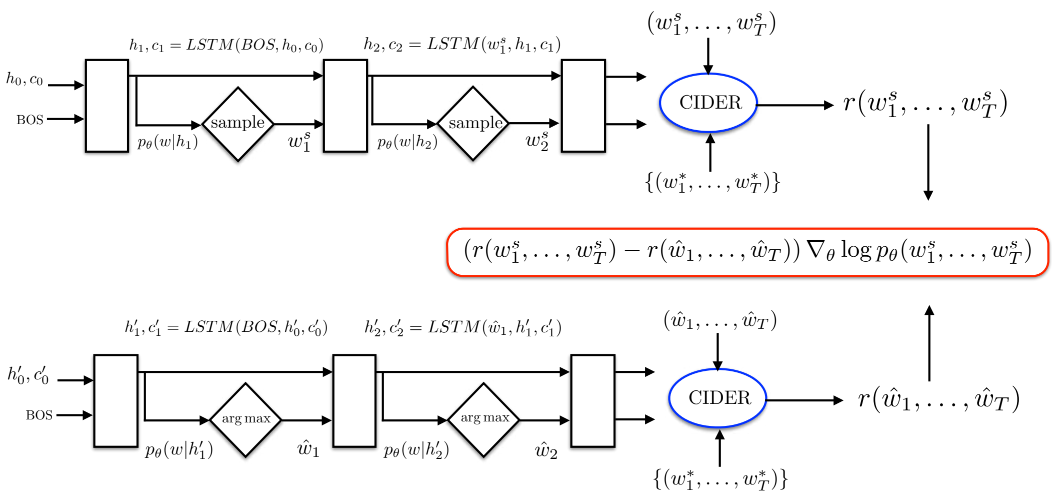
Another important generalization is to utilize the inference algorithm as a critic to replace the learned critic of traditional actor-critic approaches. Like for traditional actor-critic methods, this biases the learning procedure, but can be used to trade off variance for bias. Specifically, the *primary* reward signal at time t can be based on a sequence that samples only n future tokens, and then executes the inference algorithm to complete the sequence. The primary reward is then based on $\tilde{w} = \{w^s_{1:t+n}, \hat{w}_{t+n+1:T}\}$, and can further be baselined in a time-dependent manner using TD-SCST. The resulting reward signal in this case is $r(\tilde{w}) - r(\bar{w})$. We call this variant *True SCST*.

We have experimented with both TD-SCST and "True" SCST as described above on the MSCOCO task, but found that they did not lead to significant additional gain. We have also experimented with learning a control-variate for the SCST baseline on MSCOCO to no avail. Nevertheless, we anticipate that these generalizations will be important for other sequence modeling tasks, and policy-gradient-based RL more generally.

5. Experiments

Dataset. We evaluate our proposed method on the MSCOCO dataset [1]. For offline evaluation purposes we used the data splits from [21]. The training set contains $113,287$ images, along with 5 captions each. We use a set of $5K$ image for validation and report results on a test set of $5K$ images as well, as given in [21]. We report four widely used automatic evaluation metrics, BLEU-4, ROUGEL, METEOR, and CIDEr. We prune the vocabulary and drop any word that has count less then five, we end up with a vocabulary of size 10096 words.

Image Features 1) *FC Models.* We use two type of Features: a) (FC-2k) features, where we encode each image with Resnet-101 (101 layers) [22]. Note that we do not rescale or crop each image. Instead we encode the full image with the final convolutional layer of resnet, and apply average pooling, which results in a vector of dimension 2048. b) (FC-15K) features where we stack average pooled 13 layers of Resnet-101 (11×1024 and 2×2048). These 13 layers are the odd layers of conv4 and conv5, with the exception of the 23rd layer of conv4, which was omitted. This results in a feature vector of dimension 15360.
2) *Spatial CNN features for Attention models:* (Att2in) We encode each image using the residual convolutional neural network Resnet-101 [22]. Note that we do not rescale or crop the image. Instead we encode the full



$h_1, c_1 = LSTM(BOS, h_0, c_0)$
$h_2, c_2 = LSTM(w_1^s, h_1, c_1)$
h_0, c_0
BOS
sample
$p_\theta(w|h_1)$
w_1^s
$p_\theta(w|h_2)$
w_2^s
$(w_1^s, \ldots, w_T^s)$
CIDER
$r(w_1^s, \ldots, w_T^s)$
$\{(w_1^*, \ldots, w_T^*)\}$
$(r(w_1^s, \ldots, w_T^s) - r(\hat{w}_1, \ldots, \hat{w}_T)) \nabla_\theta \log p_\theta(w_1^s, \ldots, w_T^s)$
$h'_1, c'_1 = LSTM(BOS, h'_0, c'_0)$
$h'_2, c'_2 = LSTM(\hat{w}_1, h'_1, c'_1)$
h'_0, c'_0
arg max
$p_\theta(w|h'_1)$
$\hat{w}_1$
$p_\theta(w|h'_2)$
$\hat{w}_2$
$(\hat{w}_1, \ldots, \hat{w}_T)$
$r(\hat{w}_1, \ldots, \hat{w}_T)$

Figure 1: Self-critical sequence training (SCST). The weight put on words of a sampled sentence from the model is determined by the difference between the reward for the sampled sentence and the reward obtained by the estimated sentence under the test-time inference procedure (greedy inference depicted). This harmonizes learning with the inference procedure, and lowers the variance of the gradients, improving the training procedure.

image with the final convolutional layer of Resnet-101, and apply spatially adaptive max-pooling so that the output has a fixed size of $14 \times 14 \times 2048$. At each time step the attention model produces an attention mask over the 196 spatial locations. This mask is applied and then the result is spatially averaged to produce a 2048 dimension representation of the attended portion of the image.

Implementation Details. The LSTM hidden, image, word and attention embeddings dimension are fixed to 512 for all of the models discussed herein. All of our models are trained according to the following recipe, except where otherwise noted. We initialize all models by training the model under the XE objective using ADAM [20] optimizer with an initial learning rate of 5×10^{-4}. We anneal the learning rate by a factor of 0.8 every three epochs, and increase the probability of feeding back a sample of the word posterior by 0.05 every 5 epochs until we reach a feedback probability 0.25 [8]. We evaluate at each epoch the model on the development set and select the model with best CIDEr score as an initialization for SCST training. We then run SCST training initialized with the XE model to optimize the CIDEr metric (specifically, the CIDEr-D metric) using ADAM with a learning rate 5×10^{-5} [1]. Initially when experimenting with FC-2k and FC-15k models we utilized curriculum learning (CL) during training, as proposed in [7], by increasing the number of words that are sampled and trained under CIDEr by one each epoch (the prefix of the sentence remains under the XE criterion until eventually being subsumed). We have since realized that for the MSCOCO task CL is not required, and provides little to no boost in performance. The results reported here for the FC-2K and FC-15K models are trained with CL, while the attention models were trained directly on the entire sentence for all epochs after being initialized by the XE seed models.

[1]In the case of the Att2all models, the XE model was trained for only 20 epochs, and the learning rate was also annealed during RL training.

Training	Evaluation Metric			
Metric	CIDEr	BLEU4	ROUGEL	METEOR
XE	90.9	28.6	52.3	24.1
XE (beam)	94.0	29.6	52.6	25.2
MIXER	101.9	30.9	53.8	24.9
SCST	**106.3**	**31.9**	**54.3**	**25.5**

Table 1: Performance of self-critical sequence training (SCST) versus MIXER [7] on the test portion of the Karpathy splits when trained to optimize the CIDEr metric (FC-2K models). Both improve the seed cross-entropy trained model, but SCST significantly outperforms MIXER.

5.1. Offline Evaluation

Evaluating different RL training strategies.

Table 1 compares the performance of SCST to MIXER [7] (test set, Karpathy splits). In this experiment, we utilize "curriculum learning" (CL) by optimizing the expected reward of the metric on the last n words of each training sentence, optimizing XE on the remaining sentence prefix, and slowly increasing n. The results reported were generated with the optimized CL schedule reported in [7].

Metric	MIXER less CL	SCST
CIDEr	110.4 ± 0.5	**113.8** ± 0.3
BLEU4	32.8 ± 0.1	**34.1** ± 0.1
ROUGE	55.2 ± 0.1	**55.7** ± 0.1
METEOR	26.0 ± 0.04	**26.6** ± 0.04

Table 2: mean/std. performance of SCST versus REINFORCE with learned baseline (MIXER less CL; Att2all models, 4 seeds; Karpathy test set; CIDEr optimized).

We found that CL was not necessary to train both SCST and REINFORCE with a learned baseline on MSCOCO—turning off CL sped up training and yielded equal or better results. The gain of SCST over learned baselines was consistently >3 CIDEr points, regardless of the CL schedule and the initial seed (c.f. table 2 and graph in supp. material).

Training Metric	Evaluation Metric CIDEr	BLEU4	ROUGEL	METEOR
XE	90.9	28.6	52.3	24.1
XE (beam)	94.0	29.6	52.6	25.2
CIDEr	**106.3**	31.9	54.3	25.5
BLEU	94.4	**33.2**	53.9	24.6
ROUGEL	97.7	31.6	**55.4**	24.5
METEOR	80.5	25.3	51.3	**25.9**

Table 3: Performance on the test portion of the Karpathy splits [21] as a function of training metric (FC-2K models). Optimizing the CIDEr metric increases the overall performance under the evaluation metrics the most significantly. The performance of the seed cross-entropy (XE) model is also depicted. All models were decoded greedily, with the exception of the XE beam search result, which was optimized to beam 3 on the validation set.

Training on different metrics.

We experimented with training directly on the evaluation metrics of the MSCOCO challenge. Results for FC-2K models are depicted in table 3. In general we can see that optimizing for a given metric during training leads to the best performance on that same metric at test time, an expected result. We experimented with training on multiple test metrics, and found that we were unable to outperform the overall performance of the model trained only on the CIDEr metric, which lifts the performance of all other metrics considerably. For this reason most of our experimentation has since focused on optimizing CIDEr.

Single FC-Models Versus Attention Models. We trained FC models (2K and 15 K), as well as attention models (Att2in and Att2all) using SCST with the CIDEr metric. We trained 4 different models for each FC and attention type, starting the optimization from four different random seeds [2]. We report in Table 4, the system with best performance for each family of models on the test portion of Karpathy splits [21]. We see that the FC-15K models outperform the FC-2K models. Both FC models are outperformed by the attention models, which establish a new state of the art for a single model performance on Karpathy splits. Note that this quantitative evaluation favors attention models is inline with our observation that attention models tend to generalize better and compose outside of the context of the training of MSCOCO, as we will see in Section 6.

Single Best Models (XE)

Model Type	Search Method	Evaluation Metric CIDEr	BLEU4	ROUGEL	METEOR
FC-2K	greedy	90.9	28.6	52.3	24.1
	beam	94.0	29.6	52.6	25.2
FC-15K	greedy	94.1	29.5	52.9	24.4
	beam	96.1	30.0	52.9	25.2
Att2in	greedy	99.0	30.6	53.8	25.2
	beam	**101.3**	**31.3**	**54.3**	**26.0**
Att2all (RL seed)	greedy	97.9	29.3	53.4	25.4
	beam	99.4	30.0	53.4	25.9

Single Best Models (SCST unless noted o.w.)

Model Type	Search Method	Evaluation Metric CIDEr	BLEU4	ROUGEL	METEOR
FC-2K	greedy	106.3	31.9	54.3	25.5
	beam	106.3	31.9	54.3	25.5
FC-15K	greedy	106.4	32.2	54.6	25.5
	beam	106.6	32.4	54.7	25.6
Att2in	greedy	111.3	33.3	55.2	26.3
	beam	111.4	33.3	55.3	26.3
Att2all	greedy	113.7	34.1	55.6	26.6
	beam	**114.0**	**34.2**	**55.7**	**26.7**
4 Att2all (MIXER-)	greedy	110.2	32.7	55.1	26.0
	beam	110.5	32.8	55.2	26.1

Table 4: Performance of the best XE and corr. SCST-trained single models on the Karpathy test split (best of 4 random seeds). The results obtained via the greedy decoding and optimized beam search are depicted. Models learned using SCST were trained to directly optimize the CIDEr metric. MIXER less CL results (MIXER-) are also included.

Model Ensembling. In this section we investigate the performance of ensembling over 4 random seeds of the XE and SCST-trained FC models and attention models. We see in Table 5 that ensembling improves performance and confirms the supremacy of attention modeling, and establishes

[2]please consult the supplementary material for additional details regarding how the models were trained.

Ensembled Models (XE)

Model Type	Search Method	Evaluation Metric CIDEr	BLEU4	ROUGEL	METEOR
4 FC-2K	greedy	96.3	30.1	53.5	24.8
	beam	99.2	31.2	53.9	25.8
4 FC-15K	greedy	97.7	30.7	53.8	25.0
	beam	100.7	31.7	54.2	26.0
4 Att2in	greedy	102.8	32.0	54.7	25.7
	beam	**106.5**	**32.8**	**55.1**	**26.7**
Att2all (RL seeds)	greedy	102.0	31.2	54.4	26.0
	beam	104.7	32.2	54.8	26.7

Ensembled Models (SCST unless o.w. noted)

Model Type	Search Method	Evaluation Metric CIDEr	BLEU4	ROUGEL	METEOR
4 FC-2K	greedy	108.9	33.1	54.9	25.7
	beam	108.9	33.2	54.9	25.7
4 FC-15K	greedy	110.4	33.4	55.4	26.1
	beam	110.4	33.4	55.4	26.2
4 Att2in	greedy	114.7	34.6	56.2	26.8
	beam	115.2	34.8	56.3	26.9
4 Att2all	greedy	116.8	35.2	56.5	27.0
	beam	**117.5**	**35.4**	**56.6**	**27.1**
4 Att2all (MIXER-)	greedy	113.8	34.2	56.0	26.5
	beam	113.6	33.9	55.9	26.5

Table 5: Performance of Ensembled XE and SCST-trained models on the Karpathy test split (ensembled over 4 random seeds). The models learned using self-critical sequence training (SCST) were trained to optimize the CIDEr metric. MIXER less CL results (MIXER-) are also included.

yet another state of the art result on Karpathy splits [21]. Note that in our case we ensemble only 4 models and we don't do any fine-tuning of the Resnet. NIC [23], in contrast, used an ensemble of 15 models with fine-tuned CNNs.

5.2. Online Evaluation on MS-COCO Server

Table 6 reports the performance of two variants of 4 ensembled attention models trained with self-critical sequence training (SCST) on the official MSCOCO evaluation server. The previous best result on the leaderboard (as of April 10, 2017) is also depicted. We outperform the previous best system on all evaluation metrics.

6. Example of Generated Captions

Here we provide a qualitative example of the captions generated by our systems for the image in figure 2. This picture is taken from the objects out-of-context (OOOC) dataset of images [24]. It depicts a boat situated in an unusual context, and tests the ability of our models to compose descriptions of images that differ from those seen during

Ensemble SCST models	Evaluation Metric CIDEr	BLEU4	ROUGEL	METEOR
Ens. 4 (Att2all)	**114.7**	**35.2**	**56.3**	**27.0**
Ens. 4 (Att2in)	112.3	34.4	55.9	26.8
Previous best	104.9	34.3	55.2	26.6

Table 6: Performance of 4 ensembled attention models trained with self-critical sequence training (SCST) on the official MSCOCO evaluation server (5 reference captions). The previous best result on the leaderboard (as of 04/10/2017) is also depicted (http://mscoco.org/dataset/#captions-leaderboard, Table C5, Watson Multimodal).

training. The top 5 captions returned by the XE and SCST-trained FC-2K, FC-15K, and attention model ensembles when deployed with a decoding "beam" of 5 are depicted in figure 3 [3]. On this image the FC models fail completely, and the SCST-trained ensemble of attention models is the only system that is able to correctly describe the image. In general we found that the performance of all captioning systems on MSCOCO data is qualitatively similar, while on images containing objects situated in an uncommon context [24] (i.e. unlike the MSCOCO training set) our attention models perform much better, and SCST-trained attention models output yet more accurate and descriptive captions. In general we qualitatively found that SCST-trained attention models describe images more accurately, and with higher confidence, as reflected in Figure 3, where the average of the log-likelihoods of the words in each generated caption are also depicted. Additional examples, including an example with the corresponding heat-maps for the SCST-trained Att2in ensemble can be found in the supplementary material (figure 8 of section D). We found that Att2in attention models performed better than our Att2all models when applied to images "from the wild", so here we focus on demonstrating the performance of our Att2in systems.

7. Discussion and Future Work

In this paper we have presented a simple and efficient approach to more effectively baselining the REINFORCE algorithm for policy-gradient based RL, which allows us to more effectively train on non-differentiable metrics, and leads to significant improvements in captioning performance on MSCOCO—our results on the MSCOCO evaluation sever establish a new state-of-the-art on the task. The self-critical approach, which normalizes the reward obtained by sampled sentences with the reward obtained by the model under the test-time inference algorithm is intuitive, and avoids having to estimate both action-dependent and action-independent reward functions.

[3]pls. consult the the supp. material for further details on beam search.

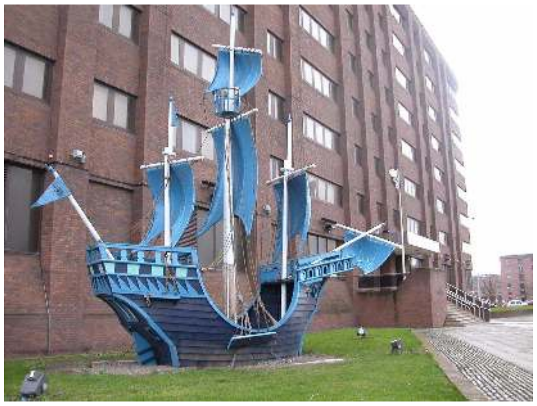


Figure 2: An image from the objects out-of-context (OOOC) dataset of images from [24].

1. a blue of a building with a blue umbrella on it -1.234499
2. a blue of a building with a blue and blue umbrella -1.253700
3. a blue of a building with a blue umbrella -1.261105
4. a blue of a building with a blue and a blue umbrella on top of it -1.277339
5. a blue of a building with a blue and a blue umbrella -1.280045

(a) Ensemble of 4 Attention models (Att2in) trained with XE.

1. a blue boat is sitting on the side of a building -0.194627
2. a blue street sign on the side of a building -0.224760
3. a blue umbrella sitting on top of a building -0.243250
4. a blue boat sitting on the side of a building -0.248849
5. a blue boat is sitting on the side of a city street -0.265613

(b) Ensemble of 4 Attention models (Att2in) trained with SCST.

1. a couple of bikes parked next to each other -1.005856
2. a couple of bikes parked next to each other on a of a building -1.039497
3. a couple of bikes parked next to each other on a building -1.050528
4. a couple of bikes parked next to each other on a street -1.055674
5. a couple of bikes parked next to a building -1.070224

(c) Ensemble of 4 FC-2K models trained with XE.

1. a statue of a man on a bike with a building -0.376297
2. a statue of a building with a bicycle on a street -0.414397
3. a statue of a bicycle on a building with a surfboard -0.423379
4. a statue of a bicycle on a building with a umbrella -0.430222
5. a statue of a building with a umbrella -0.435535

(d) Ensemble of 4 FC-2K models trained with SCST.

1. a couple of bikes that are next to a building -0.898723
2. a couple of bikes parked next to a building -0.932335
3. a row of bikes parked next to a building -0.950412
4. a row of bicycles parked next to a building -0.971651
5. a couple of bikes parked next to each other -0.985120

(e) Ensemble of 4 FC-15K models trained with XE.

1. a scooter parked in front of a building -0.326988
2. a group of a motorcycle parked in front of a building -0.366700
3. a group of surfboards in front of a building -0.386932
4. a scooter parked in front of a building with a clock -0.429441
5. a scooter parked in front of a building with a building -0.433893

(f) Ensemble of 4 FC-15K models trained with SCST.

Figure 3: Captions generated for the image depicted in Figure 2 by the various models discussed in the paper. Beside each caption we report the average log probability of the words in the caption. On this image, which presents an object situated in an atypical context [24], the FC models fail to give an accurate description, while the attention models handle the previously unseen image composition well. The models trained with SCST return a more accurate and more detailed summary of the image.

References

[1] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *EECV*, 2014. 1, 4

[2] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP*, 2014. 1

[3] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *CVPR*, 2015. 1, 2

[4] Andrej Karpathy and Fei-Fei Li. Deep visual-semantic alignments for generating image descriptions. *CVPR*, 2015. 1, 2

[5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 1997. 1, 2

[6] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 1, 2, 3

[7] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *ICLR*, 2015. 1, 2, 3, 4, 5, 10

[8] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *NIPS*, 2015. 1, 4, 5, 10

[9] Alex Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. *Neural Information Processing Systems (NIPS) 2016*, 2016. 1, 4

[10] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *ACL*, 2002. 2

[11] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain, 2004. 2

[12] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, volume 29, pages 65–72, 2005. 2

[13] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 2, 10

[14] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT Press, 1998. 2, 3

[15] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Machine Learning*, pages 229–256, 1992. 2, 3

[16] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015. 2

[17] Wojciech Zaremba and Ilya Sutskever. Reinforcement learning neural turing machines. *Arxiv*, 2015. 2, 3

[18] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014. 2

[19] Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. An actor-critic algorithm for sequence prediction. *Arxiv*, 2016. 2

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 3, 5

[21] Andrej Karpathy and Fei-Fei Li. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 4, 6, 7

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[23] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge. *PAMI*, 2016. 7

[24] Myung Jinchoi, Antonio Torralba, and Alan S. Willsky. Context models and out-of-context objects. 7, 8, 10, 13, 16, 17

Self-critical Sequence Training for Image Captioning: Supplementary Material

A. Beam search

Throughout the paper and in this supplementary material we often refer to caption results and evaluation metric results obtained using "beam search". This section briefly summarizes our beam search procedure. While decoding the image to generate captions that describe it, rather than greedily selecting the most probable word ($N = 1$), we can maintain a list of the N most probable sub-sequences generated so far, generate posterior probabilities for the next word of each of these sub-sequences, and then again prune down to the N-best sub-sequences. This approach is widely referred to as a *beam search*, where N is the width of the decoding "beam". In our experiments we additionally prune away hypotheses within the N-best list that have a log probability that is below that of the maximally probable partial sentence by more than $\Delta_{\log} = 5$. For all reported results, the value of N is tuned on a per-model basis on the validation set (of the Karpathy splits). On MSCOCO data, $N = 2$ is typically optimal for cross-entropy (XE) trained models and SCST-trained models, but in the latter case beam search provides only a very small boost in performance. For our captioning demonstrations we set $N = 5$ for all models for illustrative purposes, and because we have qualitatively observed that for test images that are substantially different from those encountered during training, beam search is important.

B. Performance of XE versus SCST trained models

In tables 4 and 5 of the main text we compared the performance of models trained to optimize the CIDEr metric with self-critical sequence training (SCST) with that of their corresponding bootstrap models, which were trained under the cross-entropy (XE) criterion using scheduled sampling [8]. We provide some additional details about these experiments here. For all XE models, the probability p_f of feeding forward the maximally probable word rather than the ground-truth word was increased by 0.05 every 5 epochs until reaching a maximum value of 0.25. The XE model with the best performance on the validation set of the Karpathy splits was then selected as the bootstrap model for SCST (with the exception of the Att2all attention models, where CE training was intentionally terminated prematurely to encourage more exploration during early epochs of RL training).

For all models, the performance of greedily decoding each word at test time is reported, as is the performance of *beam search* as described in the previous section. As reported in [7], we found that beam search using RL-trained models resulted in very little performance gain. Figure 5 depicts the performance of our best Att2in model, which is trained to directly optimize the CIDEr metric, as a function of training epoch and evaluation metric, on the validation portion of the Karpathy splits. Optimizing CIDEr clearly improves all of the MSCOCO evaluation metrics substantially.

C. Performance of MIXER versus SCST trained models

SCST consistently outperforms MIXER by more than four CIDER points (c.f. figure 4 and table 1).

D. Examples of Generated Captions

Figures 6-14 depict demonstrations of the captioning performance of all systems. In general we found that the performance of all captioning systems on MSCOCO data is qualitatively similar, while on images containing objects situated in an uncommon context [24] (i.e. unlike the MSCOCO training set) our attention models perform much better, and SCST-trained attention models output yet more accurate and descriptive captions. Attention heat-maps for the image and corresponding captions depicted in figure 6 and 7 are given in figure 8. The heatmaps of the attention weights are reasonably inline with the predicted words in both cases, and the SCST attention weights are spatially sharper here, and in general.

E. Further details and analysis of SCST training

One detail that was crucial to optimizing CIDEr to produce better models was to include the EOS tag as a word. When the EOS word was omitted, trivial sentence fragments such as "with a" and "and a" were dominating the metric gains, despite the "gaming" counter-measures (sentence length and precision clipping) that are included in CIDEr-D [13], which is what we optimized. Including the EOS tag substantially lowers the reward allocated to incomplete sentences, and completely resolved this issue. Another more obvious detail that is important is to associate the reward for the sentence with the first

EOS encountered. Omitting the reward from the first EOS fails to reward sentence completion which leads to run-on, and rewarding any words that follow the first EOS token is inconsistent with the decoding procedure.

This work has focused on optimizing the CIDEr metric, since, as discussed in the paper, optimizing CIDER substantially improves all MSCOCO evaluation metrics, as was shown in tables 4 and 5 and is depicted in figure 5. Nevertheless, directly optimizing another metric does lead to higher evaluation scores on that same metric as shown, and so we have started to experiment with including models trained on Bleu, Rouge-L, and METEOR in our Att2in ensemble to attempt to improve it further. So far we have not been able to substantially improve performance w.r.t. the other metrics without more substantially

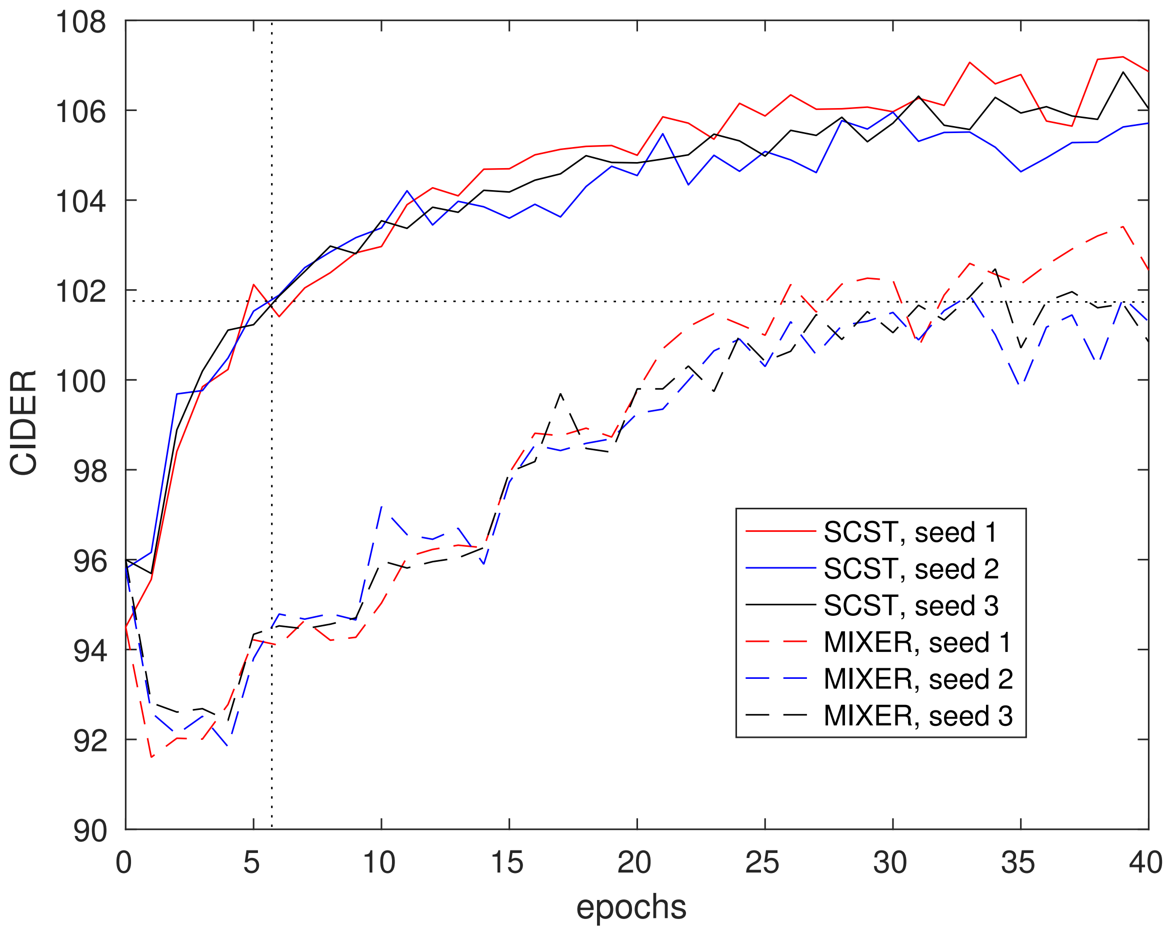


Figure 4: SCST vs. MIXER (FC models) over multiple random seeds.

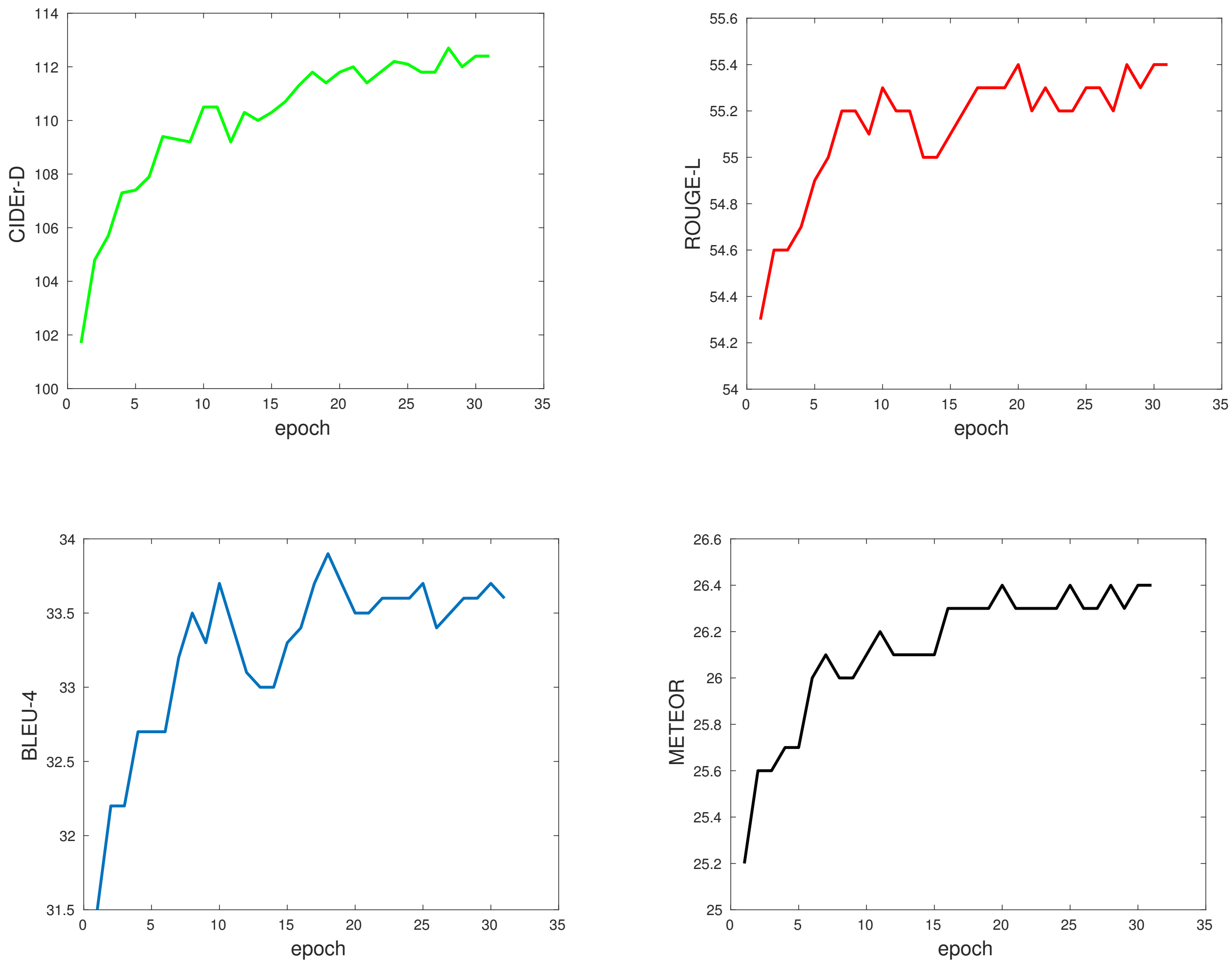


Figure 5: Performance of our best Att2in model, which is trained to directly optimize the CIDEr metric, as a function of training epoch on the validation portion of the Karpathy splits, for the CIDEr, BLEU-4, ROUGE-L, and METEOR MSCOCO evaluation metrics. Optimizing CIDEr improves all of these evaluation metrics substantially.

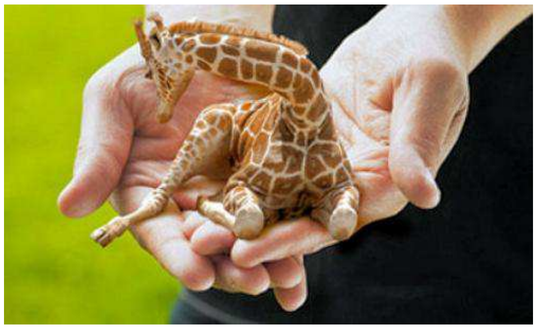
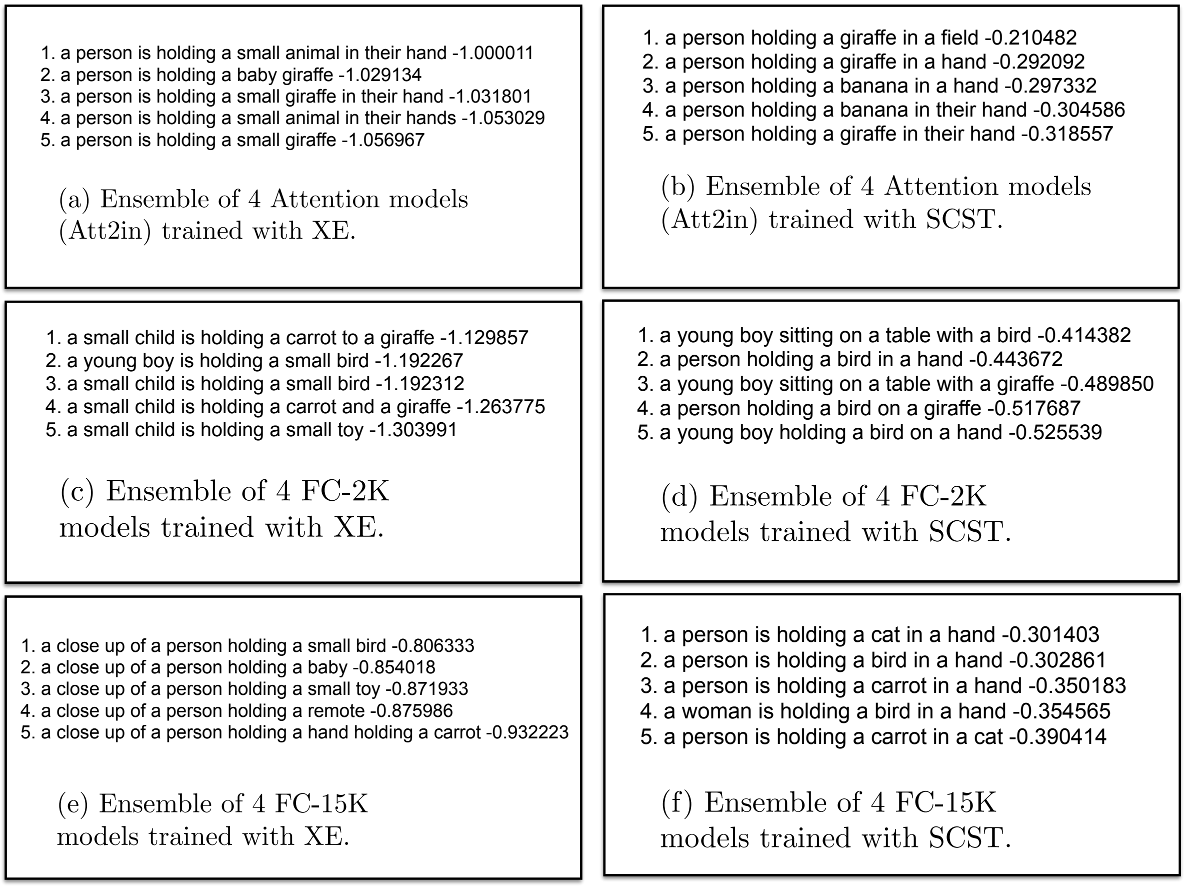


Figure 6: Picture of a common object in MSCOCO (a giraffe) situated in an *uncommon* context (out of COCO domain) [24].



1. a person is holding a small animal in their hand -1.000011
2. a person is holding a baby giraffe -1.029134
3. a person is holding a small giraffe in their hand -1.031801
4. a person is holding a small animal in their hands -1.053029
5. a person is holding a small giraffe -1.056967

(a) Ensemble of 4 Attention models (Att2in) trained with XE.

1. a person holding a giraffe in a field -0.210482
2. a person holding a giraffe in a hand -0.292092
3. a person holding a banana in a hand -0.297332
4. a person holding a banana in their hand -0.304586
5. a person holding a giraffe in their hand -0.318557

(b) Ensemble of 4 Attention models (Att2in) trained with SCST.

1. a small child is holding a carrot to a giraffe -1.129857
2. a young boy is holding a small bird -1.192267
3. a small child is holding a small bird -1.192312
4. a small child is holding a carrot and a giraffe -1.263775
5. a small child is holding a small toy -1.303991

(c) Ensemble of 4 FC-2K models trained with XE.

1. a young boy sitting on a table with a bird -0.414382
2. a person holding a bird in a hand -0.443672
3. a young boy sitting on a table with a giraffe -0.489850
4. a person holding a bird on a giraffe -0.517687
5. a young boy holding a bird on a hand -0.525539

(d) Ensemble of 4 FC-2K models trained with SCST.

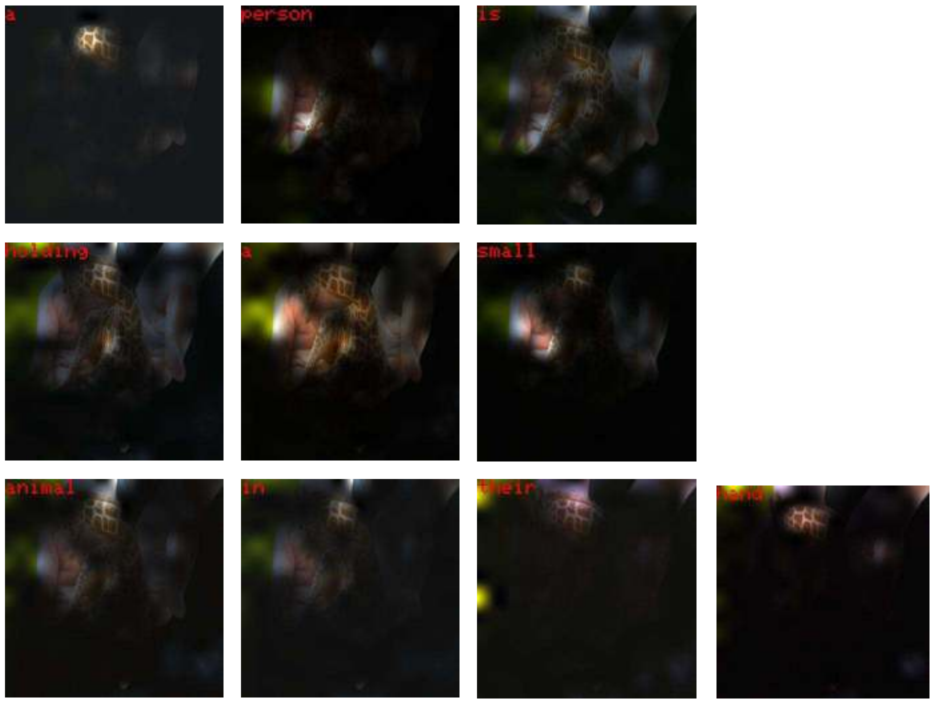
1. a close up of a person holding a small bird -0.806333
2. a close up of a person holding a baby -0.854018
3. a close up of a person holding a small toy -0.871933
4. a close up of a person holding a remote -0.875986
5. a close up of a person holding a hand holding a carrot -0.932223

(e) Ensemble of 4 FC-15K models trained with XE.

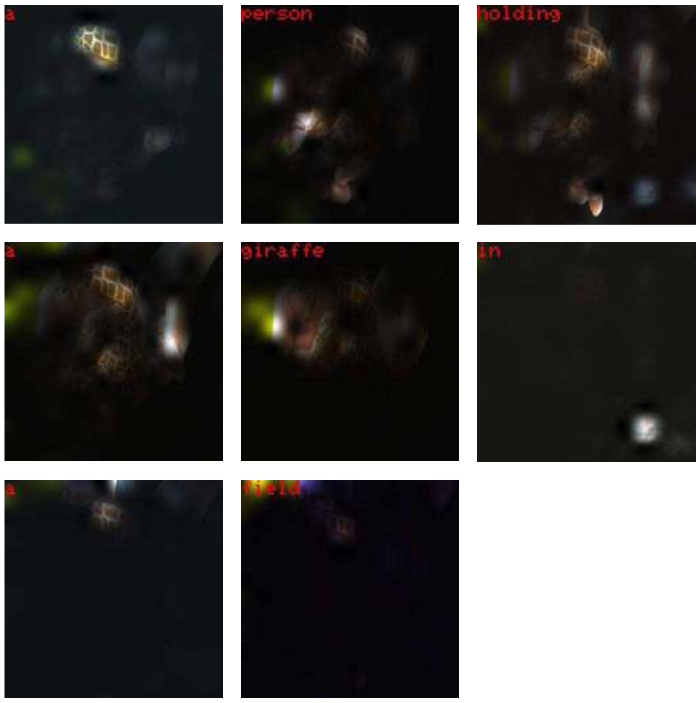
1. a person is holding a cat in a hand -0.301403
2. a person is holding a bird in a hand -0.302861
3. a person is holding a carrot in a hand -0.350183
4. a woman is holding a bird in a hand -0.354565
5. a person is holding a carrot in a cat -0.390414

(f) Ensemble of 4 FC-15K models trained with SCST.

Figure 7: Captions generated by various models discussed in the paper to describe the image depicted in figure 6. Beside each caption we report the average of the log probabilities of each word, normalized by the sentence length. Notice that the attention models trained with SCST give an accurate description of this image with high confidence. Attention models trained with XE are less confident about the correct description. FC models trained with CE or SCST fail at giving an accurate description.



(a) Attention heat-maps for the best model in the XE-trained ensemble of attention models, for the image depicted in figure 6.



(b) Attention heat-maps for the best model in the SCST-trained ensemble of attention models, for the image depicted in figure 6.

.

Figure 8: Attention heat-maps.

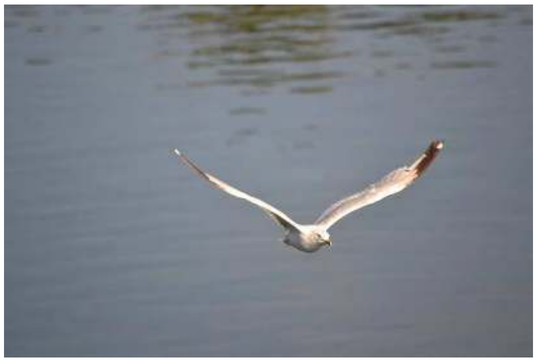
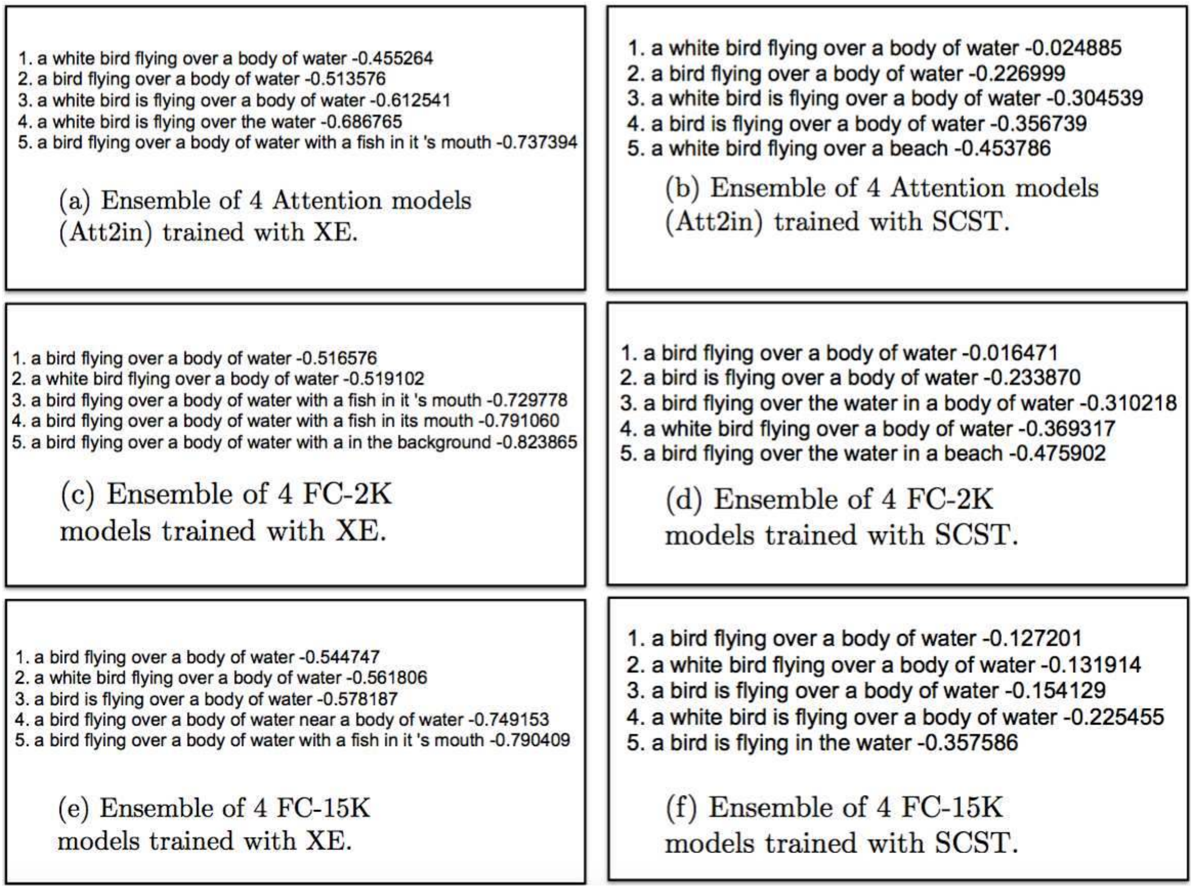


Figure 9: An image from the MSCOCO test set (Karpathy splits).



1. a white bird flying over a body of water -0.455264
2. a bird flying over a body of water -0.513576
3. a white bird is flying over a body of water -0.612541
4. a white bird is flying over the water -0.686765
5. a bird flying over a body of water with a fish in it 's mouth -0.737394

(a) Ensemble of 4 Attention models (Att2in) trained with XE.

1. a white bird flying over a body of water -0.024885
2. a bird flying over a body of water -0.226999
3. a white bird is flying over a body of water -0.304539
4. a bird is flying over a body of water -0.356739
5. a white bird flying over a beach -0.453786

(b) Ensemble of 4 Attention models (Att2in) trained with SCST.

1. a bird flying over a body of water -0.516576
2. a white bird flying over a body of water -0.519102
3. a bird flying over a body of water with a fish in it 's mouth -0.729778
4. a bird flying over a body of water with a fish in its mouth -0.791060
5. a bird flying over a body of water with a in the background -0.823865

(c) Ensemble of 4 FC-2K models trained with XE.

1. a bird flying over a body of water -0.016471
2. a bird is flying over a body of water -0.233870
3. a bird flying over the water in a body of water -0.310218
4. a white bird flying over a body of water -0.369317
5. a bird flying over the water in a beach -0.475902

(d) Ensemble of 4 FC-2K models trained with SCST.

1. a bird flying over a body of water -0.544747
2. a white bird flying over a body of water -0.561806
3. a bird is flying over a body of water -0.578187
4. a bird flying over a body of water near a body of water -0.749153
5. a bird flying over a body of water with a fish in it 's mouth -0.790409

(e) Ensemble of 4 FC-15K models trained with XE.

1. a bird flying over a body of water -0.127201
2. a white bird flying over a body of water -0.131914
3. a bird is flying over a body of water -0.154129
4. a white bird is flying over a body of water -0.225455
5. a bird is flying in the water -0.357586

(f) Ensemble of 4 FC-15K models trained with SCST.

Figure 10: Captions generated for the image depicted in Figure 9 by various models discussed in the paper. Beside each caption we report the average log probability of the words in the caption. All models perform well on this test image from the MSCOCO distribution. More generally we have observed that qualitatively, all models perform comparably on the MSCOCO test images.

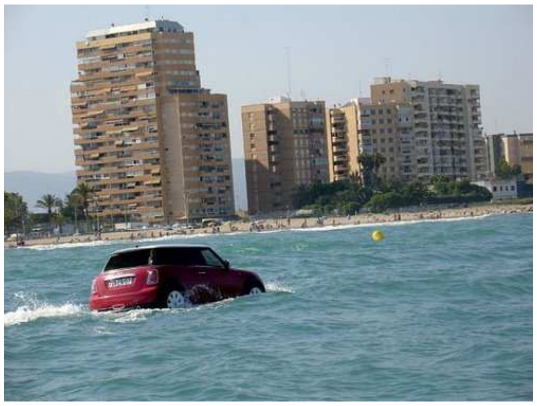
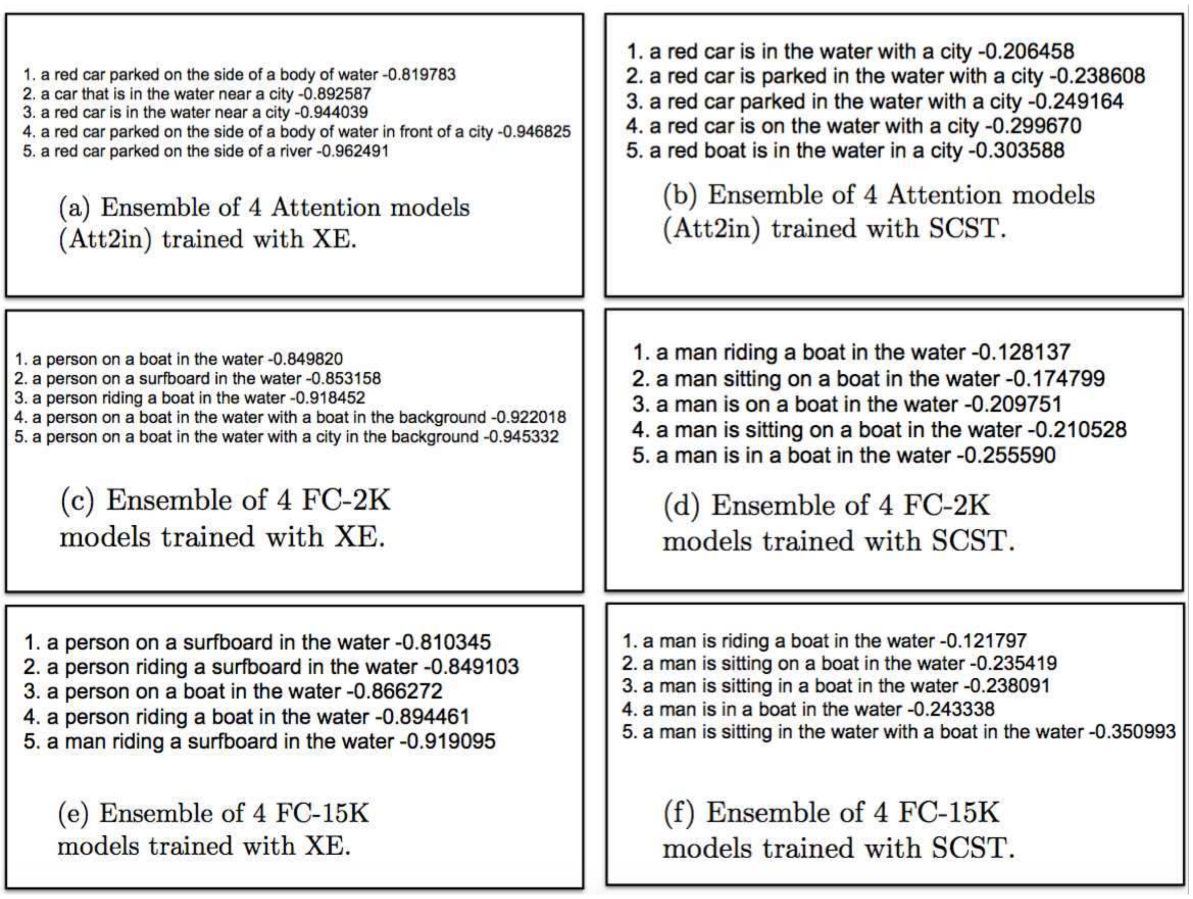


Figure 11: An image from the objects out-of-context (OOOC) dataset of images from [24].



1. a red car parked on the side of a body of water -0.819783
2. a car that is in the water near a city -0.892587
3. a red car is in the water near a city -0.944039
4. a red car parked on the side of a body of water in front of a city -0.946825
5. a red car parked on the side of a river -0.962491

(a) Ensemble of 4 Attention models (Att2in) trained with XE.

1. a red car is in the water with a city -0.206458
2. a red car is parked in the water with a city -0.238608
3. a red car parked in the water with a city -0.249164
4. a red car is on the water with a city -0.299670
5. a red boat is in the water in a city -0.303588

(b) Ensemble of 4 Attention models (Att2in) trained with SCST.

1. a person on a boat in the water -0.849820
2. a person on a surfboard in the water -0.853158
3. a person riding a boat in the water -0.918452
4. a person on a boat in the water with a boat in the background -0.922018
5. a person on a boat in the water with a city in the background -0.945332

(c) Ensemble of 4 FC-2K models trained with XE.

1. a man riding a boat in the water -0.128137
2. a man sitting on a boat in the water -0.174799
3. a man is on a boat in the water -0.209751
4. a man is sitting on a boat in the water -0.210528
5. a man is in a boat in the water -0.255590

(d) Ensemble of 4 FC-2K models trained with SCST.

1. a person on a surfboard in the water -0.810345
2. a person riding a surfboard in the water -0.849103
3. a person on a boat in the water -0.866272
4. a person riding a boat in the water -0.894461
5. a man riding a surfboard in the water -0.919095

(e) Ensemble of 4 FC-15K models trained with XE.

1. a man is riding a boat in the water -0.121797
2. a man is sitting on a boat in the water -0.235419
3. a man is sitting in a boat in the water -0.238091
4. a man is in a boat in the water -0.243338
5. a man is sitting in the water with a boat in the water -0.350993

(f) Ensemble of 4 FC-15K models trained with SCST.

Figure 12: Captions generated for the image depicted in Figure 11 by the various models discussed in the paper. Beside each caption we report the average log probability of the words in the caption. On this image, which presents an object situated in an atypical context [24], the FC models fail to give an accurate description, while the attention models handle the previously unseen image composition well. The models trained with SCST return a more accurate and more detailed summary of the image.

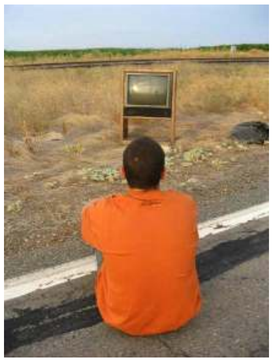
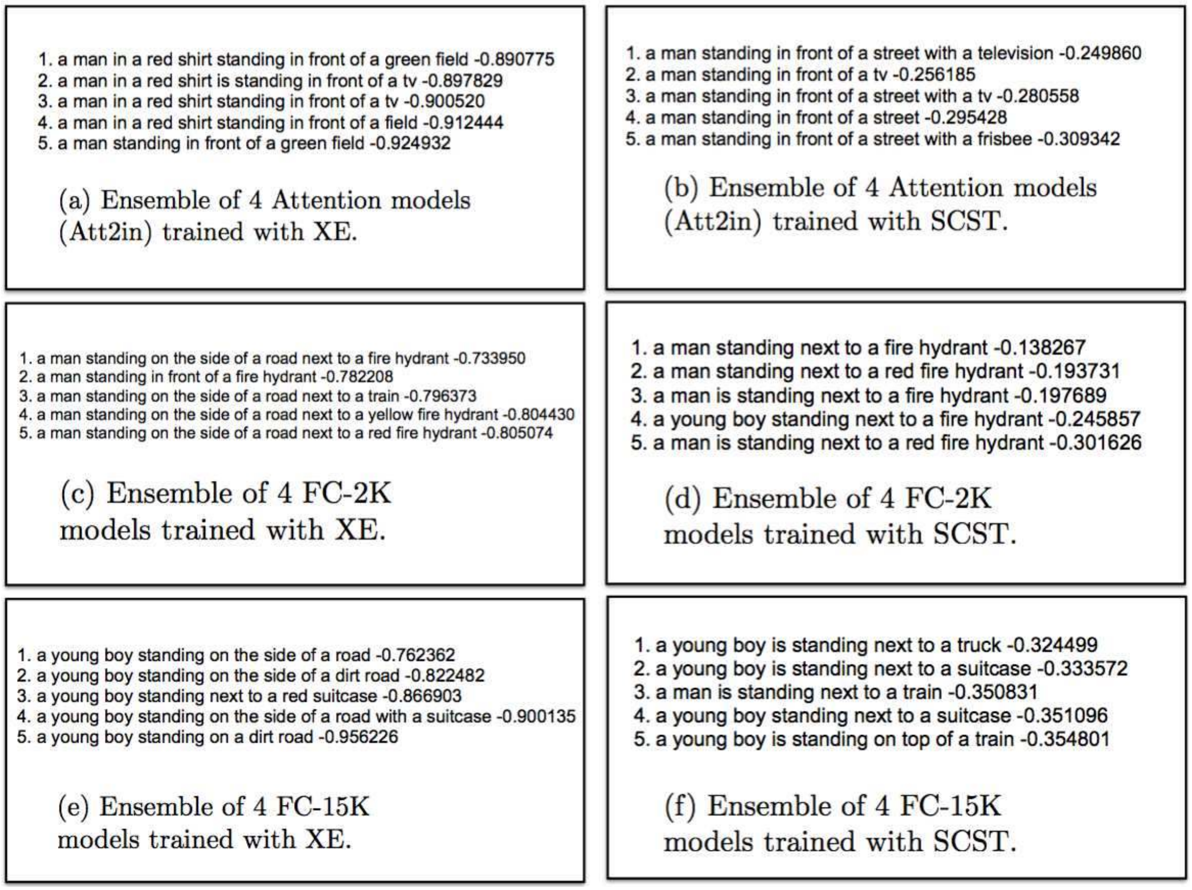


Figure 13: An image from the objects out-of-context (OOOC) dataset of images from [24].



1. a man in a red shirt standing in front of a green field -0.890775
2. a man in a red shirt is standing in front of a tv -0.897829
3. a man in a red shirt standing in front of a tv -0.900520
4. a man in a red shirt standing in front of a field -0.912444
5. a man standing in front of a green field -0.924932

(a) Ensemble of 4 Attention models (Att2in) trained with XE.

1. a man standing in front of a street with a television -0.249860
2. a man standing in front of a tv -0.256185
3. a man standing in front of a street with a tv -0.280558
4. a man standing in front of a street -0.295428
5. a man standing in front of a street with a frisbee -0.309342

(b) Ensemble of 4 Attention models (Att2in) trained with SCST.

1. a man standing on the side of a road next to a fire hydrant -0.733950
2. a man standing in front of a fire hydrant -0.782208
3. a man standing on the side of a road next to a train -0.796373
4. a man standing on the side of a road next to a yellow fire hydrant -0.804430
5. a man standing on the side of a road next to a red fire hydrant -0.805074

(c) Ensemble of 4 FC-2K models trained with XE.

1. a man standing next to a fire hydrant -0.138267
2. a man standing next to a red fire hydrant -0.193731
3. a man is standing next to a fire hydrant -0.197689
4. a young boy standing next to a fire hydrant -0.245857
5. a man is standing next to a red fire hydrant -0.301626

(d) Ensemble of 4 FC-2K models trained with SCST.

1. a young boy standing on the side of a road -0.762362
2. a young boy standing on the side of a dirt road -0.822482
3. a young boy standing next to a red suitcase -0.866903
4. a young boy standing on the side of a road with a suitcase -0.900135
5. a young boy standing on a dirt road -0.956226

(e) Ensemble of 4 FC-15K models trained with XE.

1. a young boy is standing next to a truck -0.324499
2. a young boy is standing next to a suitcase -0.333572
3. a man is standing next to a train -0.350831
4. a young boy standing next to a suitcase -0.351096
5. a young boy is standing on top of a train -0.354801

(f) Ensemble of 4 FC-15K models trained with SCST.

Figure 14: Captions generated for the image depicted in Figure 13 by the various models discussed in the paper. Beside each caption we report the average log probability of the words in the caption. On this image, which presents an object situated in an atypical context [24], the FC models fail to give an accurate description, while the attention models handle the previously unseen image composition well. The models trained with SCST return a more accurate and more detailed summary of the image.