# Attentive Linear Transformation for Image Captioning

Senmao Ye, Junwei Han, *Senior Member, IEEE*, and Nian Liu

***Abstract*— We propose a novel attention framework called attentive linear transformation (ALT) for automatic generation of image captions. Instead of learning the spatial or channel-wise attention in existing models, ALT learns to attend to the high-dimensional transformation matrix from the image feature space to the context vector space. Thus ALT can learn various relevant feature abstractions, including spatial attention, channel-wise attention, and visual dependence. Besides, we propose a soft threshold regression to predict the spatial attention probabilities. It preserves more relevant local regions than popular softmax regression. Extensive experiments on the MS COCO and the Flickr30k data sets all demonstrate the superiority of our model compared with other state-of-the-art models.**

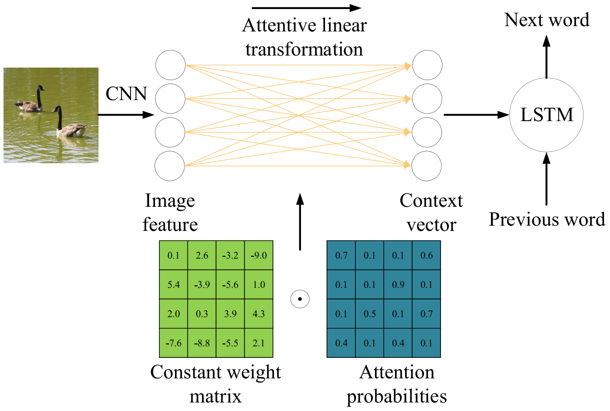
***Index Terms*— Image captioning, attention, linear transformation, CNN, LSTM.**

## I. Introduction

Automatic generation of image captions is a fundamental task to build a bridge between visual system and language system. On the one hand, image caption models need to determine what objects and events appear in an image. On the other hand, they need to describe the visual relationships properly in a natural language. Image captioning is one of the very challenging computer vision tasks, but it has great significance such as helping visual impaired people and building intelligent robots. Despite the challenging nature, it has attracted increasing research interests [1]–[6].

Due to the advancement of deep neural networks, the encoder-decoder framework shows promising results in image captioning [1]–[6]. In this framework, CNN is used to encode an image input to a vector and RNN is used to decode the vector to a English sentence.

Spatial attention models are usually used to boost the encoder-decoder framework by focusing on relevant image regions [6]–[8]. When generating a target word, only a small local region is involved and other parts of the image are irrelevant, which would mislead the training and the inference process. Spatial attention models treat each image as a set of local regions and predict an attention probability for each local region. Then they pick out regions with large attention probabilities to calculate a context feature and subsequently feed it into RNN to predict the next word.



| | | | |
|---|---|---|---|
| 0.1 | 2.6 | -3.2 | -9.0 |
| 5.4 | -3.9 | -5.6 | 1.0 |
| 2.0 | 0.3 | 3.9 | 4.3 |
| -7.6 | -8.8 | -5.5 | 2.1 |

| | | | |
|---|---|---|---|
| 0.7 | 0.1 | 0.1 | 0.6 |
| 0.1 | 0.1 | 0.9 | 0.1 |
| 0.1 | 0.5 | 0.1 | 0.7 |
| 0.4 | 0.1 | 0.4 | 0.1 |

Fig. 1. Illustration of the attentive linear transformation. The constant weights convey information from the image feature to the context vector. Each weight is multiplied by an attention probability. Irrelevant visual information will be suppressed by small attention probabilities.

However, a single region still contains both relevant and irrelevant details such as color, quantity and category. Although spatial attention models can attend to relevant regions, they can not attend to the details within each region. Previous work in [9] and [10] address this problem by applying channel-wise attention models on image features. Nevertheless, these models usually consider spatial attention and channel-wise attention as two independent modules and generate them separately. This choice is straightforward but suboptimal for captioning models, since image features are usually in a high-dimensional feature space while these models decompose it into low-dimensional spaces (the spatial dimension and the channel dimension) and generate attention probabilities on each of them, which will lose high-dimensional semantic information. Thus, a unified attention model should not only attend to spatial regions and feature channels, but also take into account other abstract semantic concepts.

To address the above problems, we propose a novel attention model called attentive linear transformation (ALT). ALT attends to the high-dimensional transformation weight matrix from an image feature space to a context vector space rather than spatial regions or feature channels. As depicted in Figure 1, ALT first learns a constant transformation weight matrix. Then it learns another attention matrix and multiplies

Manuscript received February 6, 2018; revised June 21, 2018; accepted July 3, 2018. Date of publication July 12, 2018; date of current version August 14, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Vishal Monga. *(Corresponding author: Junwei Han.)*

The authors are with the School of Automation, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: junweihan2010@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2018.2855406

both matrices to modulate the activeness of the transformation weights. Thus, irrelevant visual information can be suppressed by inactive transformation weights with small attention probabilities.

The advantage of ALT is that it can attend to subtler and more abstract visual concepts than previous models. The transformation from the image feature space to the context space learns high-level contextual feature abstractions between the two spaces. Since ALT directly attends to this high-dimensional transformation matrix, it can flexibly catch relevant visual semantics. When we deploy the proposed ALT in the caption model, it can learn to select relevant and informative feature abstractions instead of a concrete form like spatial region or feature channel. This property can help the caption model to explore more useful concepts for image captioning besides spatial regions and feature channels.

One significant learned concept of ALT is that it can control the adaptive visual dependence of the predicted word. The attention probabilities in previous spatial attention models have a fixed summation of 1. Thus, spatial attention models have to feed the same amount of visual information into RNN, even when generating words without visual meaning. On the contrary, ALT doesn't have such restrains for the learned attention. It can adaptively choose to attend to more or less visual feature abstractions. When the target word has no relevance with visual information, ALT chooses to suppress all the transformation weights. This property is important since irrelevant visual information could mislead the word prediction.

Besides, we propose a soft threshold regression which is more suitable than softmax regression to compute attention probabilities of image regions. Softmax regression follows the *winner-take-all* theory. It highlights the most relevant region while suppresses other regions. However, this property may neglect other subordinate clues to predict the right word. Soft threshold regression implicitly sets a threshold for the relevance degree. Regions above this threshold will be assigned close attention probabilities so that more relevant visual regions can be selected. We further combine soft threshold regression with ALT to handle the spatial structure in images.

Our overall contributions are:

- We propose an attentive linear transformation to extract relevant information from an image feature space to a context vector space. By using our proposed ALT, our caption model can incorporate the spatial attention, the channel-wise attention, the visual dependence, and other informative high-level semantics together for image captioning.
- We propose a soft threshold regression for computing attention probabilities over image regions, and it shows better performance than softmax regression.
- Our model outperforms other state-of-the-art image caption models on the MS COCO dataset and the Flickr30k datasets. Extensive experimental results demonstrate the effectiveness of our ALT model.

## II. Related Work

Generating image captions has been long studied as the bridge between computer vision and natural language processing. Traditionally, heavily hand-designed systems are built to solve this problem. The work in [4] and [11] first detect objects, attributes and prepositions in images and then use probabilistic graphic models to infer corresponding captions. In [12]–[16], authors use object proposals and attributes to represent images, then they use powerful language parsing models to generate captions. However, the above systems are limited by the predefined templates and language grammars. The winner of 2015 COCO Captioning Challenge [17], first uses multi-instance learning to directly detect words in images and then uses the maximum entropy language model to predict captions. Some work such as [18]–[20] address the problem via retrieving the most relevant descriptions. However, because of lacking the capability of describing previously unseen compositions of objects, these methods have weak generalizatibility.

In recent years, neural language models have been introduced into image caption by [21]–[23]. Kiros *et al.* [21] use a multimodel log-bilinear model that is biased by image features. Motivated by the advancement of sequence generation in machine translation, Mao *et al.* [22] and Vinyals *et al.* [6] propose to model caption generation as a seq2seq task. Specifically, they use a CNN to encode the input image into a feature vector, and then use a RNN to decode the feature vector into a caption. The difference is that, in [22], a global image feature is input into the RNN decoder at every time step while it is only input at the first time step in [6]. Different from [6], [22], Lu *et al.* [8] feeds in a global image feature and a local context vector at every time step. Karpathy and Fei-Fei [3] simultaneously generate image captions and locate their corresponding locations.

Further, many methods have been proposed to boost neural caption models. On one hand, some work try to build more powerful RNN decoders. Bengio *et al.* [1] use scheduled sampling to handle the bias problem in RNN. Gu *et al.* [24] use highway RNN to exploit the hierarchical and temporal structure of history words. Chen *et al.* [25] enhance the RNN decoder by assigning larger weights to the keywords. On the other hand, some work try to enhance caption models with more powerful image representation. Liu *et al.* [26] formulate image captioning as a translation task from region proposals to captions. Wu *et al.* [27] incorporate attributes with high-level concepts into the caption model. Pu *et al.* [28] propose a variational convolutional auto-encoder for image captioning.

Attention mechanism has achieved considerable success in image captioning [5], [7], [8], [29]. Irrelevant image regions has been long studied in computer vision such as [30]–[33]. Xu *et al.* [5] propose two attention models for caption generation, which are called soft attention and hard attention, respectively. The former samples a single relevant region according to the attention probability distribution, and the latter uses the feature expectation of the sampled regions as the context vector. Following this work, the soft attention model is widely adopted by later work [7], [8], [29] for its efficiency and stability. Reference [34] also observes that contiguous models are more stable than discrete models. Review network in [7] produces several review steps to catch global relationships and generates captions by attending to the review steps. Semantic attention in [29] first produces a set of image attributes and attends to these top-down semantic concept proposals. Adaptive attention in [8] proposes to attend to both image regions and a RNN memory called visual sentinel. By attending more to the visual sentinel, the adaptive attention model can let in less visual information. Our model can also adaptively control the activeness of attention. The attention probabilities in our model don't have a fixed sum. If little visual information is needed, the sum of the attention probabilities will be small which makes our model take in less visual information.

To address the limitation of vanilla attention models, some other work [26], [35] try to predict flexible spatial regions. MAT in [26] uses pre-trained object detection methods to locate objects. Area attention in [35] locates objects with a model that can be trained without additional bounding box annotations. Both models provide a better way to represent image information while ours provides a better way to extract relevant image information. We think it's interesting to combine both works in the future.

The work in [9] and [10] apply channel-wise attention models to image features. These models and our model all try to pay attention to details in the image feature representation. However, visual information and language information are quite different. Directly feeding attended visual features into the RNN decoder ignores the semantic gap between the two modalities and leads to suboptimal modeling. In our work, we define two separate feature spaces for the image encoder outputs and the language decoder inputs, and then use a linear transformation to connect them. Our model attends to the high-dimensional transformation matrix from an image feature space to a language feature space. Comparing with [9] and [10] which only attend to channels in image features, our model can attend to finer image feature details.

## III. Attentive Linear Transformation for Image Captioning

We first make an introduction of the encoder-decoder framework to predict captions, and then introduce our attentive linear transformation.

### A. Encoder-Decoder Framework for Image Captioning

We encode an input image $I$ into a feature representation with deep convolutional neural network (CNN). Then we decode the feature representation into a caption with recurrent neural network (RNN). The network is trained to minimize the cross entropy loss between the true caption distribution $q$ and the predicted caption distribution $p$:

$$H(q, p) = E_q[-\log\ p]. \tag{1}$$

According to the chain rule, log $p$ of a caption is factorized into ordered conditionals at different time steps:

$$\log\ p = \sum_{t=1}^{N} \log\ p(y_t|y_1, \dots, y_{t-1}, I), \tag{2}$$

$$p(y_t|y_1, \dots, y_{t-1}, I) = \phi(\mathbf{h}_t + \mathbf{l}_t), \tag{3}$$

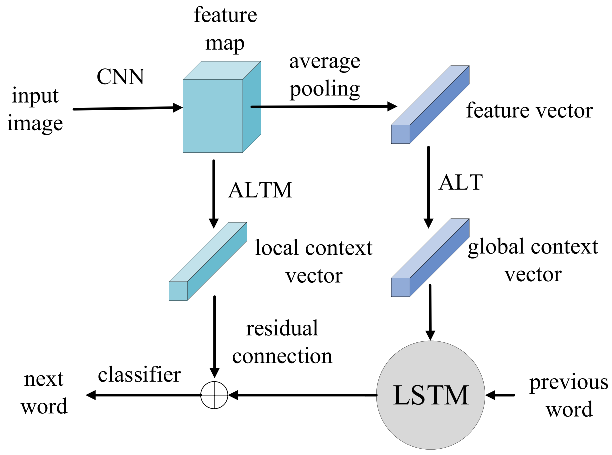


Fig. 2. Illustration of the overall framework. ALT controls LSTM unit with global information. ALTM provides the word classifier with precise local information.

where $y_t$ is the predicted word in the caption at time t. $N$ is the quantity of time steps. The conditional probability of $y_t$ is modeled by a nonlinear mapping $\phi$. $\mathbf{l}_t$ denotes the local context vector at time $t$ providing local visual information. $\mathbf{h}_t$ means the RNN hidden state at time $t$. Instead of a vanilla RNN, we use Long Short Term Memory (LSTM) shown below because LSTM has been demonstrated to handle long sequences better:

$$\mathbf{h}_t = \text{LSTM}(\mathbf{w}_t, \mathbf{g}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}), \tag{4}$$

where $\mathbf{w}_t$ is the input word vector, $\mathbf{g}_t$ is the global context vector at time $t$ providing global visual information. $\mathbf{c}_{t-1}$ is the memory cell vector of LSTM at $t - 1$.

Following [8], the context vectors $\mathbf{g}_t$ and $\mathbf{l}_t$ are important, since they convey visual information from the image representation to the decoder. $\mathbf{g}_t$ controls LSTM with the global information. $\mathbf{l}_t$ provides the word classifier with precise local information. In the next section, we will introduce the attentive linear transformations to produce the global context vector $\mathbf{g}_t$ and the local context vector $\mathbf{l}_t$, respectively. At last, we will provide a precise formulation of the LSTM decoder in Section III-E. The overall framework is visualized in Figure 2.

### B. Attentive Linear Transformation

We first define two finite-dimensional vector spaces $V$ and $G$ for the global image vector representation $\mathbf{v} \in \mathbb{R}^n$ and the global context vector $\mathbf{g}_t \in \mathbb{R}^m$,respectively. We aim to use a linear transformation $\mathbf{P}_t \in \mathbb{R}^{m\times n}$ to extract relevant information from $\mathbf{v}$:

$$\mathbf{g}_t = \mathbf{P}_t\mathbf{v}. \tag{5}$$

We model $\mathbf{P}_t$ as a element-wise multiplication of a constant matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$ and a variable matrix $\mathbf{A}_t \in \mathbb{R}^{m\times n}$.

$$\mathbf{P}_t = \mathbf{W} \odot \mathbf{A}_t, \tag{6}$$

where $\odot$ indicates the element-wise multiplication. $\mathbf{W}$ is the transformation weights from $V$ to $G$. However, $\mathbf{W}$ keeps identical for all the images and words, which is not the case since different images contain different visual contents and different words for a specific image correspond to different visual information. Thus, we propose to use attention probabilities $\mathbf{A}_t$ to adaptively control the activeness of each weight in $\mathbf{W}$. As a result, irrelevant information will be suppressed by small attention probabilities.

Previous attention models [5], [9], [10] are also linear transformations. The spatial attention model is the weighted summation of a number of local feature vectors. The channel-wise attention model multiplies the image feature with a diagonal matrix. In this work, we first directly model attention mechanism as a general linear transformation.

In practice, predicting a full-rank probability matrix with the same parameter number as $\mathbf{W}$ is prohibitive because of too much parameters to learn, which are naturally redundant in neural networks. There are several methods such as [36] and [37] to compress the size of deep neural networks. Here, we simply factorize $\mathbf{A}_t$ as the multiplication of two low rank matrices $\mathbf{A}_t^1 \in \mathbb{R}^{m\times r}, \mathbf{A}_t^2 \in \mathbb{R}^{n\times r}$:

$$\mathbf{A}_t = \mathbf{A}_t^1(\mathbf{A}_t^2)^T. \tag{7}$$

By substituting Equation 7 and Equation 6 into Equation 5, we can obtain:

$$\mathbf{g}_t = (\mathbf{W} \odot (\mathbf{A}_t^1(\mathbf{A}_t^2)^T))\mathbf{v}. \tag{8}$$

Equation 8 is still infeasible in practice, because directly storing $\mathbf{A}_t$ of all the RNN time steps will occupy a lot of GPU memory. As $\mathbf{A}_t$ has a very small rank compared with its size, we can use much fewer elements to store $\mathbf{A}_t$. Here, we use a small trick to avoid the direct presence of $\mathbf{A}_t$. We rewrite $\mathbf{A}_t^1, \mathbf{A}_t^2$ as:

$$\mathbf{A}_{\rm t}^1 = \left[\mathbf{a}_1^1\ \mathbf{a}_2^1\ \cdots\ \mathbf{a}_j^1\ \cdots\ \mathbf{a}_r^1\right], \quad \mathbf{a}_j^1 \in \mathbb{R}^{m\times 1}, \tag{9}$$

$$\mathbf{A}_{\rm t}^2 = \left[\mathbf{a}_1^2\ \mathbf{a}_2^2\ \cdots\ \mathbf{a}_j^2\ \cdots\ \mathbf{a}_r^2\right], \quad \mathbf{a}_j^2 \in \mathbb{R}^{n\times 1}, \tag{10}$$

where we omit the subscript $t$ of $\mathbf{a}_j^1$ and $\mathbf{a}_j^2$ for convenience. Then

$$\mathbf{A}_t^1(\mathbf{A}_t^2)^T = \sum_{j=1}^{r} \mathbf{a}_j^1(\mathbf{a}_j^2)^T. \tag{11}$$

By substituting Equation 11 into Equation 8, we can obtain:

$$\mathbf{g}_t = \sum_{j=1}^{r} \mathbf{a}_j^1 \odot (\mathbf{W}^T(\mathbf{v} \odot \mathbf{a}_j^2)). \tag{12}$$

Compared with Equation 8, Equation 12 avoids the direct presence of $\mathbf{A}_t$. We use multi-layer perceptions (MLP) to predict $\mathbf{a}_j^1$ and $\mathbf{a}_j^2$:

$$\mathbf{a}_j^i = \text{Logistic}(\text{MLP}_j^i(\mathbf{v}, \mathbf{h}_{t-1})), \quad i = 1, 2, \tag{13}$$

where $\mathbf{h}_{t-1}$ is the hidden state of the LSTM at time $t-1$ and we use the standard logistic regression as the activation function:

$$\text{Logistic}(x) = \frac{1}{1+e^{-x}}. \tag{14}$$

When the rank $r > 1$, ALT can learn multiple attention mechanisms of the feature abstractions from the image feature space to the context vector space, instead of only learning one feature scaling mechanism in previous channel-wise attention models.

Unlike spatial attention models, the sum of all the attention probabilities in $\mathbf{A}_t$ is not fixed. $\mathbf{A}_t$ can be very large when generating words with significant visual meaning and a null matrix when generating visually-unrelated words. Thus, when to attend to visual information can be adaptively controlled. In other word, our model is aware of the visual dependence of generated words.

### *C. Soft Threshold Regression*

As ALT is proposed to attend to global image feature vectors, we also generalize it to attend to spatial image features. In this section, we first introduce the soft threshold regression for computing attention probabilities in local regions.

In previous work, attention probabilities in local regions are predicted by the softmax regression:

$$p(x_k) = \frac{e^{x_k}}{\sum_{j=1}^{K} e^{x_j}}. \tag{15}$$

Usually, $x_k$ is interpreted as the negative energy value in the $k$th region. Softmax regression can also be expressed as:

$$p(x_k) = \frac{1}{1+\sum_{j=1, j\neq k}^{K} e^{x_j - x_k}}. \tag{16}$$

We see that the attention probabilities only depend on the difference values of negative energies. As the gap between the largest $x_k$ and the others grows gradually, it becomes a form of *winner-take-all* (every output is nearly 0 except the largest output 1) [38]. This property leads to excellent single-class classification performance. However, in image captioning, there may be more than one relevant local regions for each word generation. *Winner-take-all* tends to only notice the most relevant region and neglects all other regions, even if some of them are indispensable.

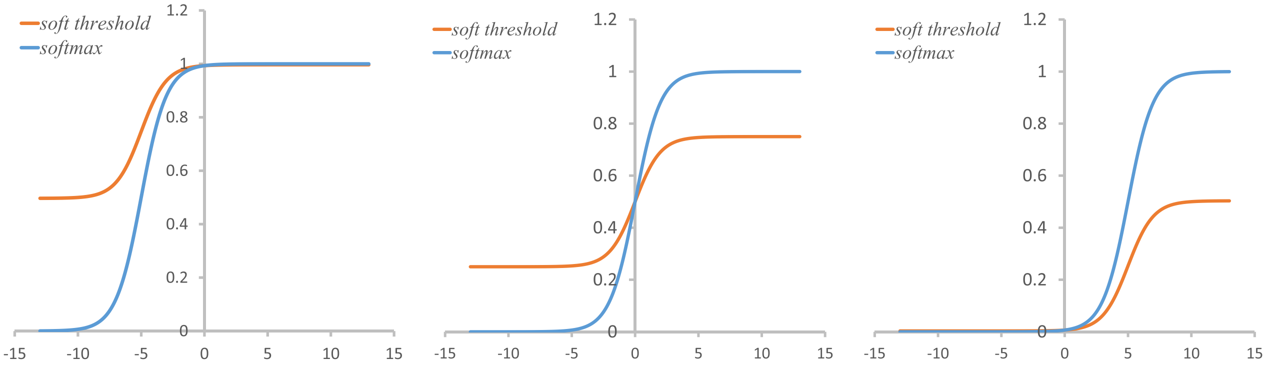
Essentially, the property of *winner-take-all* is caused by the positive second-order derivative of $e^{x_k}$ w.r.t $x_k$. Thus we replace $e^{x_k}$ with $\frac{1}{1+e^{-x_k}}$, whose second-order derivative is negative. The new regression function is as follow:

$$p(x_k) = \frac{\frac{1}{1+e^{-x_k}}}{\sum_{j=1}^{K} \frac{1}{1+e^{-x_j}}}. \tag{17}$$

Specifically, we use a standard logistic function to squash the negative energy values into $(0, 1)$ before normalization. We illustrate the difference between the proposed soft threshold regression and the softmax regression in Figure 3. As we can see, soft threshold regression is a form of thresholding. When $x_k$ is large enough, the $k_{th}$ region will also have a large attention probability, even if other regions have much larger negative energies.

### *D. Attentive Linear Transformation for Image Feature Matrix*

Attentive linear transformation for image matrix (ALTM) is a generalization of ALT for handling spatial structures. Obviously, compressing the entire image into a global vector



(a) (b) (c)

Fig. 3. Comparison between the soft threshold regression and the softmax regression. For visualization convenience, we only show two input negative energy values ($x_1$ and $x_2$). The vertical axis shows the probability of $x_1$ and the horizontal axis shows the value of $x_1$. We set $x_2$ to be a small value $-5$ in (a), 0 in (b) and a large value 5 in (c).

will lose explicit spatial visual information which could be useful for richer and descriptive captions. Thus, as same as [5], [7], and [8], we represent an image with a feature matrix:

$$\mathbf{M} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \cdots \ \mathbf{v}_K], \quad \mathbf{v}_k \in \mathbb{R}^n, \tag{18}$$

where $K$ is the quantity of regions. $\mathbf{v}_k$ is the local feature for the $k$-th region. For convenience of expression, we transform the image matrix into a vector $\hat{\mathbf{v}} \in \mathbb{R}^{(n\times K)}$:

$$\hat{\mathbf{v}} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_K \end{bmatrix}. \tag{19}$$

Then we use an attentive linear transformation to get a local context vector $\mathbf{l}_t$:

$$\mathbf{l}_t = \mathbf{P}_t \hat{\mathbf{v}}, \tag{20}$$

where $\mathbf{P}_t \in \mathbb{R}^{m\times(n\times K)}$ is factorized into:

$$\mathbf{P}_t = \mathbf{A}_t \odot \mathbf{W}. \tag{21}$$

$\mathbf{W} \in \mathbb{R}^{m\times(n\times K)}$ is the transformation weights. To further handle the spatial structure in the image representation, the attention probability matrix $\mathbf{A}_t \in \mathbb{R}^{m\times(n\times K)}$ is factorized into:

$$\mathbf{A}_t = \mathbf{A}_t^3 \bigotimes (\mathbf{A}_t^1 (\mathbf{A}_t^2)^T), \tag{22}$$

where $\bigotimes$ indicates the Kronecker product. $\mathbf{A}_t^3 \in \mathbb{R}^{1\times K}$ stands for the the spatial attention probabilities of local regions, which are predicted by the proposed soft threshold regression. The negative energy $x_k$ of each element in $\mathbf{A}_t^3$ is predicted by a multi-layer perception:

$$x_k = \mathrm{MLP}^3(\mathbf{v}_k, \mathbf{h}_t), \tag{23}$$

where $\mathbf{h}_t$ is the hidden state of the LSTM decoder at the current time step $t$. $\mathbf{a}_j^1$ and $\mathbf{a}_j^2$ of $\mathbf{A}_t^1$ and $\mathbf{A}_t^2$ are predicted by:

$$\mathbf{a}_j^i = \mathrm{Logistic}(\mathrm{MLP}_j^i(\mathbf{s}_t, \mathbf{h}_t)), \quad i = 1, 2, \tag{24}$$

where $\mathbf{s}_t$ is a weighted summation of local features:

$$\mathbf{s}_t = \mathbf{M}(\mathbf{A}_t^3)^T. \tag{25}$$

Compared to ALT, ALTM can explicitly attend to spatial regions with spatial attention probabilities in $\mathbf{A}_t^3$. As deep neural networks have too much parameters to train, it's beneficial to regularize parameters with specific domain knowledge. By factorizing the transformation matrix into different factors, the ALT framework is able to incorporate different domain knowledges. Specifically, it incorporates the spatial attention, the channel-wise attention, the visual dependence and other informative high-level semantics together for image captioning

### E. LSTM Decoder With ALT

In this section, we provide a precise formulation of the proposed model. Different from previous models, we feed in the the global context vector $\mathbf{g}_t$ and the local context vector $\mathbf{l}_t$ produced by ALT and ALTM, respectively, to provide attended global and local context information into the decoder:

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{u}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \left( T \begin{pmatrix} \mathbf{w}_t \\ \mathbf{g}_t \\ \mathbf{h}_{t-1} \end{pmatrix} \right), \tag{26}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{u}_t, \tag{27}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \tag{28}$$

$$\mathbf{p}_t = \mathrm{Softmax}(\mathbf{W}_p(\mathbf{h}_t + \mathbf{l}_t)), \tag{29}$$

where $\sigma$ indicates the sigmoid activation function, and $\mathbf{i}_t, \mathbf{f}_t, \mathbf{o}_t$ means the input gate, forget gate, output gate of the LSTM, respectively. $\mathbf{w}_t$ is the input word vector. $T : \mathbb{R}^{D+n+d} \rightarrow \mathbb{R}^{4d}$ is a linear transformation, where $D$ is the dimensionality of the word embedding and $d$ is the quantity of the LSTM cell state units. $\mathbf{W}_p$ is the weights of the last classifier.

## IV. Experiments

### A. Datasets and Evaluation Metrics

We evaluate the proposed method on the widely used MS COCO [2] benchmark dataset and the Flickr30k [18] benchmark dataset.

The **MS COCO** dataset totally consists of 123,287 images, where every image has five sentence annotations. For batch

TABLE I
PERFORMANCE COMPARISONS ON THE MS COCO DATASET. BEST SCORES ARE SHOWN IN **BOLD FACE**

| model | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | METEOR | ROUGE | CIDER | SPICE |
|---|---|---|---|---|---|---|---|---|
| Soft-Att [5] | 70.7 | 49.2 | 34.4 | 24.3 | 23.9 | - | - | - |
| Hard-Att [5] | 71.8 | 50.4 | 35.7 | 25.0 | 23.4 | - | - | - |
| Semantic-Att [29] | 70.9 | 53.7 | 40.2 | 30.4 | 24.3 | - | - | - |
| NIC [39]v2 | - | - | - | 32.1 | 25.7 | - | 99.8 | - |
| Attributes [27] | 74 | 56 | 42 | 31 | 26 | - | 94 | - |
| VAE [24] | 72 | 52 | 37 | 28 | 24 | - | 90 | - |
| Area Attention [35] | - | - | - | 30.7 | 24.5 | - | 93.8 | - |
| SCA-CNN [9] | 71.9 | 54.8 | 41.1 | 31.1 | 25.0 | 53.1 | 95.2 | - |
| Reference LSTM [25] | **76.1** | **59.6** | 45.0 | 33.7 | 25.7 | 55.0 | 102.9 | - |
| MAT [26] | 73.1 | 56.8 | 42.7 | 32.3 | 25.8 | 54.1 | 105.8 | 18.9 |
| Adaptive-Att [8] | 74.2 | 58.0 | 43.9 | 33.2 | 26.6 | 54.9 | 108.5 | 19.4 |
| PG-BCMR [40] | 75.4 | 59.1 | 44.5 | 33.2 | 25.7 | 55.0 | 101.3 | - |
| ALT-ALTM | 75.1 | 59.0 | **45.7** | **35.5** | **27.4** | **55.9** | **110.7** | **20.3** |

training, captions longer than 16 words are truncated. Then, we build a vocabulary of 9487 words which appear at least 5 times in the training data. Those words appeared less than 5 times are replaced by the unknown token **UNK**. We evaluate our model with using the publicly available test splits in [3], which adopts 5000 images for testing and another 5000 for validation. We further test on the MS COCO test set, which has 40775 images with no human captions provided. We also use the COCO evaluation server to evaluate our model.

The **Flickr30k** dataset contains of 31,783 images in total depicting humans in daily activities. Each image is annotated with 5 sentences. Captions longer than 30 words are truncated. Our vocabulary is formed by 8511 words that appear more than 5 times in the training split. Words that appeared less that 5 times are replaced by the token **UNK**. We use the publicly available splits of [3] consisting of 1000 images for testing and 1014 images for validation.

For quantitative analysis and comparison, we utilize a number of widely used evaluation metrics:BLEU@N [41], SPICE [42], METEOR [43], CIDER-D [44] and ROUGE-L [45]. Coco-caption code is used to compute these metrics.

### *B. Implementation Details*

*1) Image Feature:* There are two kinds of image features used in our full model: the global feature vector $\mathbf{v}$ and the feature matrix $\mathbf{M}$. $\mathbf{v}$ is the output of the global averaging pooling of ResNet-152 [46] with a dimension of 2048. We use the output of the last convolutional layer of ResNet-152 as the feature matrix. To reduce computational complexity, we use a $1 \times 1$ convolution followed by ReLU to reduce the feature map channels from 2048 to 512.

*2) Decoder:* We set the rank of $\mathbf{A}_t$ in ALT and ALTM to 30. The decoder is a single layer LSTM with 512 hidden units, while the dimension of the word embedding is also set to 512. All the MLPs have one hidden layer with 512 units followed by hyperbolic activation functions. We use a single model with beam size of 3 for offline evaluation on both the MS COCO dataset and the Flicker30k dataset. We ensemble 5 models trained with different initializations for online evaluation. We also include the result of our best single model on MSCOCO server, since different ensembling strategies makes the comparisons unfair. We use the same training data for offline evaluation and online evaluation on the MS COCO dataset.

*3) Training Strategy:* The LSTM hidden state and memory state are initialized as zero. Except for the pre-trained CNN encoder, all the weights are randomly initialized. Adam optimizer are used with base learning rates of 4e-4 for the LSTM decoder and 1e-5 for the CNN encoder. We set the weight-decay and momentum to be 0.999 and 0.8. We first train the network with the CNN parameters fixed, and then finetune the whole network. The training of our model can be done within 90 hours.

### *C. Comparison With Other State-of-the-Art Methods*

**Hard-Att** and **Soft-Att** in [5] are the first two attention-based image caption models. **Semantic-Att** [29] and **Adaptive-Att** [8] also incorporate attention mechanisms into their caption models. The encoder CNN has a big influence on the overall performance. **Attributes** [27], **Reference LSTM** [25], **Area Attention** [35], **PG-BCMR** [40], **Hard-Att** and **Soft-Att** in [5] use the VGG-16 Net [47] as the image encoder. **NICv2** [39], **Semantic-Att** [29] use the GoogleNet [48] as the image encoder. **Adaptive-Att** [8], **MAT** [26], **SCA-CNN** [9] and our model use the ResNet-152 [46] as the image encoder. We denote our full model as **ALT-ALTM**.

In Table I, we report the performance comparisons of our model and other state-of-the-art models on the MS COCO dataset. We observe that our model outperforms other methods in terms of BLEU@3, BLEU@4, CIDER-D, ROUGE-L, METEOR and SPICE. For BLEU@1 and BLEU@2, our model only performs slightly worse than **Reference LSTM** [25]. However, [25] and some other models [24], [27] are good at predicting accurate words. As BLEU@N relies on word matching, these models overfit to the BLEU@N. The overfitting is more obvious in Table II on the Flickr30k dataset which contains less training data than the MS COCO dataset. We also compare our performance on the MS COCO online server, where we selected top ranked methods up to the date we submitted the paper and two recent work with similar intuition [9], [26].

TABLE II

PERFORMANCE COMPARISONS ON FLICKR 30K DATASET

| model | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | METEOR | ROUGE | CIDER | SPICE |
|---|---|---|---|---|---|---|---|---|
| Hard-Att [5] | 66.9 | 43.9 | 29.6 | 19.9 | 18.46 | - | - | - |
| Semantic-Att [29] | 64.7 | 46.0 | 32.4 | 23.0 | 18.9 | - | - | - |
| VAE [28] | **73** | **53** | **38** | 25 | - | - | - | - |
| SCA-CNN [9] | 66.2 | 46.8 | 32.5 | 22.3 | 19.5 | - | - | - |
| Adaptive-Att [8] | 67.7 | 49.4 | 35.4 | 25.1 | 20.4 | 46.7 | 53.1 | 14.5 |
| ALT-ALTM | 68.5 | 50.7 | 37.0 | **27.0** | **21.2** | **48.0** | **56.2** | **15.5** |

TABLE III

PERFORMANCE COMPARISONS THE MS COCO SERVER

| model | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | METEOR | ROUGE | CIDER |
|---|---|---|---|---|---|---|---|
| 5-Refs | | | | | | | |
| Attributes [27] | 73 | 56 | 41 | 31 | 25 | 53 | 92 |
| SCA-CNN [9] | 71.2 | 54.2 | 40.4 | 30.2 | 24.4 | 52.4 | 91.2 |
| MAT [26] | 73.4 | 56.8 | 42.7 | 32.0 | 25.8 | 54.0 | 102.9 |
| Area Attention [35] | - | - | - | 31.9 | 25.4 | - | 98.1 |
| Reference LSTM [25] | **75.1** | **58.3** | 43.6 | 32.3 | 25.1 | 54.1 | 96.9 |
| Adaptive attention [8] | 74.6 | 58.2 | 44.3 | 33.5 | 26.4 | 55.0 | 103.7 |
| ALT-ALTM(single model) | 74.3 | 57.8 | 44.1 | 33.7 | 26.8 | 55.1 | 104.6 |
| ALT-ALTM(ensemble) | 74.2 | 57.7 | **44.3** | **34.1** | **27.0** | **55.2** | **105.3** |
| 40-Refs | | | | | | | |
| Attributes [27] | 89 | 80 | 69 | 58 | 33 | 67 | 93 |
| SCA-CNN [9] | 89.4 | 80.2 | 69.1 | 57.9 | 33.1 | 67.4 | 91.1 |
| Reference LSTM [25] | 91.3 | 83.3 | 72.7 | 61.6 | 33.6 | 68.8 | 98.8 |
| Review Net [7] | - | - | - | 59.7 | 34.7 | 68.6 | 96.9 |
| Adaptive attention [8] | 91.8 | 84.2 | 74.0 | 63.3 | 35.9 | 70.6 | 105.1 |
| ALT-ALTM(single model) | 92.0 | 84.0 | 73.8 | 63.1 | 36.4 | 70.7 | 104.4 |
| ALT-ALTM(ensemble) | **92.2** | **84.3** | **74.3** | **63.9** | **37.0** | **71.2** | **105.9** |

TABLE IV

PERFORMANCE COMPARISONS OF MODEL VARIANTS ON THE MS COCO DATASETS. (FULL) MEANS FINE-TUNING CNN ENCODER AND SAMPLING WITH BEAM SEARCH

| Model | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | METEOR | ROUGE | CIDER |
|---|---|---|---|---|---|---|---|
| CNN-LSTM | 65.3 | 47.1 | 34.0 | 25.1 | 21.4 | 48.2 | 73.2 |
| ALT | 68.3 | 50.7 | 37.3 | 27.9 | 23.1 | 50.4 | 84.2 |
| Soft Attention | 68.9 | 51.0 | 36.7 | 26.3 | 23.1 | 50.4 | 82.8 |
| ALTM | 69.2 | 51.5 | 38.1 | 28.2 | 23.7 | 51.0 | 86.1 |
| ALT(full) | **71.5** | **54.6** | **41.3** | **31.3** | **25.1** | **53.0** | **95.3** |
| ALTM(full) | 70.0 | 52.5 | 38.9 | 29.0 | 24.4 | 51.8 | 89.5 |

### *D. Component Analysis*

We compare different model variants in Table IV to validate the effectiveness of different components. For comparison convenience, we adopt the VGG-16 Net [47] which is the most widely used encoder CNN for image captioning. What's more, VGG can be trained with a small batch size which allows for larger ranks. Furthermore, we don't use ensemble or CNN fine tuning. We set the beam size to be 3 for every model. We use the same training strategy in every model variant.

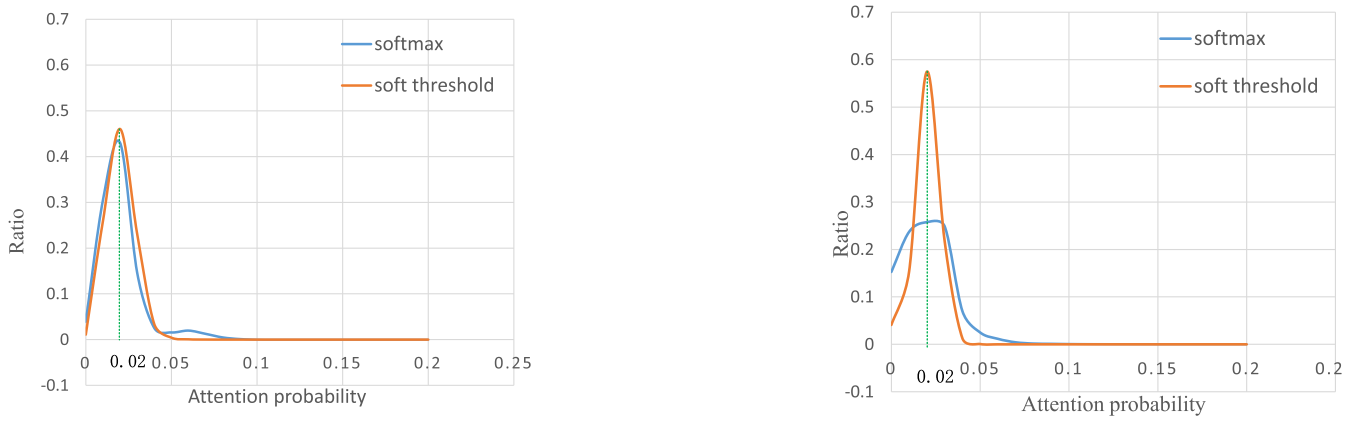
**CNN-LSTM** is the reimplementation of [6]. In **CNN-LSTM**, the image feature is the output of the fc7 layer in the VGG-16 Net. The same as [6], the image feature is only fed into the first RNN time step.

**ALT** means we adopt the proposed **ALT** in the **CNN-LSTM** model. Specifically, we use the **ALT** to transform the global image feature vector into a context vector and feed it into every LSTM time step. The rank of **ALT** is set to 40.

**Soft Attention** is the reimplementation of the soft attention model proposed by [5]. In this model, the image feature is the output of the conv5-3 layer of the VGG-16 Net. Local context vectors produced by Soft Attention are input into the LSTM unit at every step.

**ALTM** is built on top of the above **Soft Attention** model. Specifically, at each step, we feed the image feature map and previous LSTM hidden states into **ALTM** to produce the attended local context vector, which is subsequently fed into the LSTM decoder. Here, we use the softmax regression instead of the proposed soft threshold regression for a fair comparison with **Soft Attention** to demonstrate the effective of the attentive linear transformation. The rank of **ALTM** is set to 40.

From Table IV, we can see that **ALT** significantly improves the conventional **CNN-LSTM** caption model. On the one hand, **ALT** provides the LSTM decoder with more accurate and relevant context information than directly feeding in the global image feature. On the other hand, feeding in relevant image information at each time step remedies the long-term-dependency problem. We can also see that **ALTM** improves



(a)

(b)

Fig. 4. Comparison of the spatial attention probability distributions predicted by the proposed soft threshold regression and the softmax regression. (a) MS COCO. (b) Flickr 30k.

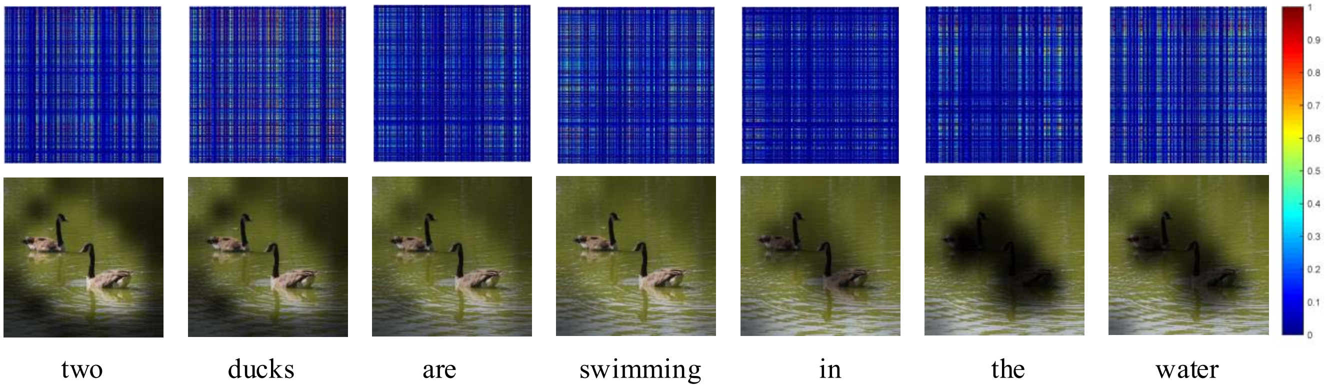


Fig. 5. Visualization of attention probabilities. The first row visualizes the attention probability matrices $\mathbf{A}_t$ at different time steps as heat maps. Blue and red indicate smaller and larger attention probabilities, respectively. In the second row, we visualize spatial attention probability $\mathbf{A}_t^3$ in ALTM directly on raw images. The brighter regions indicate higher spatial attention probabilities.

TABLE V

Performance Comparisons of Models With Different Regression Functions on the MS COCO Dataset

| Model | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | METEOR | ROUGE | CIDER |
|---|---|---|---|---|---|---|---|
| Softmax | 67.7 | 49.7 | 35.3 | 24.9 | **22.9** | 49.5 | 80.2 |
| Rectifier | 67.7 | 49.8 | 35.5 | 25.3 | 22.7 | 49.7 | 80.5 |
| Soft Threshold | **68.4** | **50.6** | **36.3** | **26.9** | **22.9** | **50.2** | **81.2** |

the conventional **Soft Attention** caption model since **ALTM** can further attend to details in a single region and learn visual dependence.

For better comparison with other methods, we also add the results of **ALT** and **ALTM** with beam search and CNN fine-tuning in Table IV. We observe that when using VGG encoder, our full model is even worse than **ALT** alone. We think the reason is that the last 4096-d image feature vector in the fc7 layer and the feature matrix in the Conv5_3 layer of VGG net are too different after two full-connected layers. This makes the residual connection (see Figure 2) not working because it's not an identity mapping [46]. Without fine tuning, ALTM performs better than ALT. But after fine tuning CNN, ALTM performs worse than ALT.We think it's because the gradients of the Conv5_3 layer in ALTM are from the LSTM instead of from the fc layers in ALT, which may make the training unstable.

To analyze the influence of the rank of the transformation weight matrix, we show different results of **ALTM** with different ranks in Figure 7. The beam size is set to 1 in every model. Rank 0 means the **ALTM** model degrades to a vanilla spatial attention model. We can see that a larger rank leads to higher CIDER scores when the rank is smaller than 40. However, when the rank is larger than 40, its increase degrades CIDER score. The probable reason is that too large ranks require too much parameters, which will make it difficult for training. Thus, there is a trade-off between the training effectiveness and the model complexity.

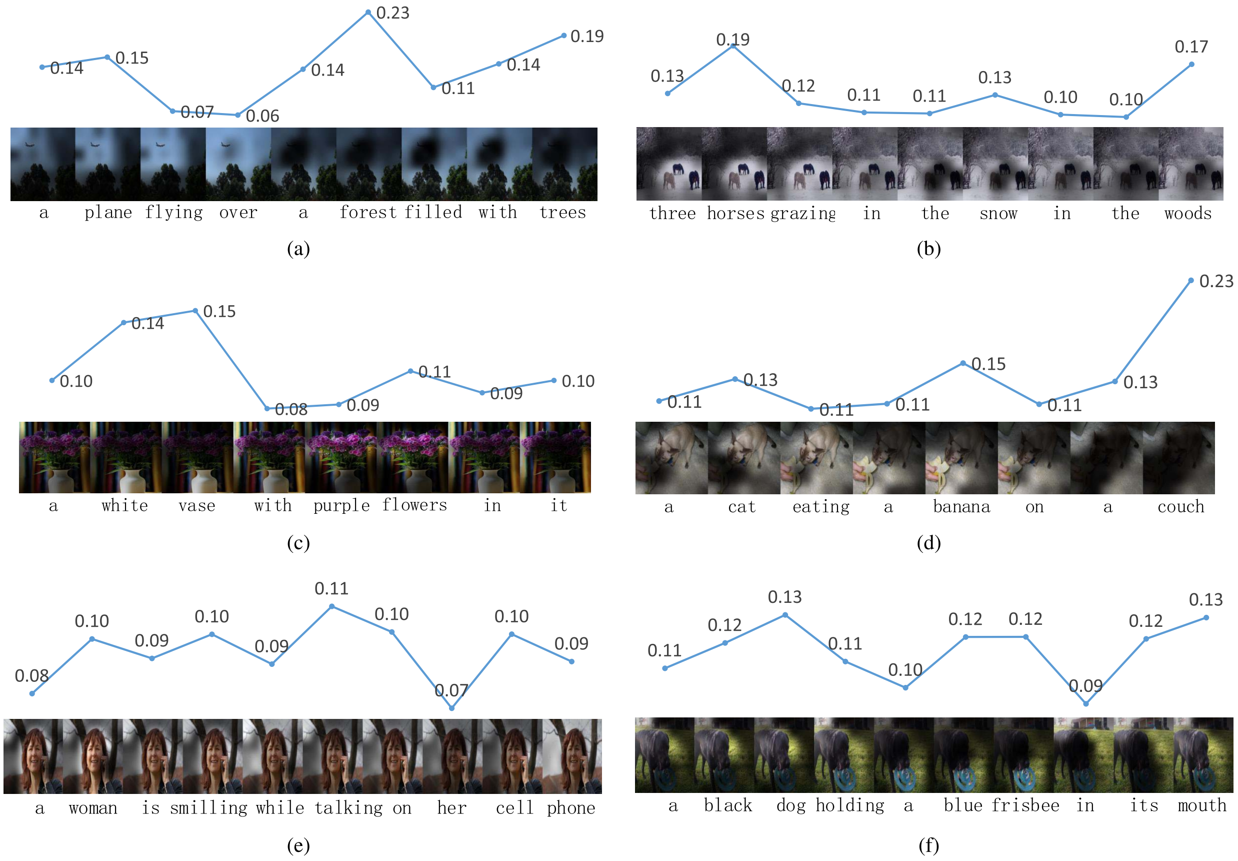


Fig. 6. Visualization of the dependence on visual information and corresponding relevant regions for every generated word. For simplicity, we only keep two places of decimal.

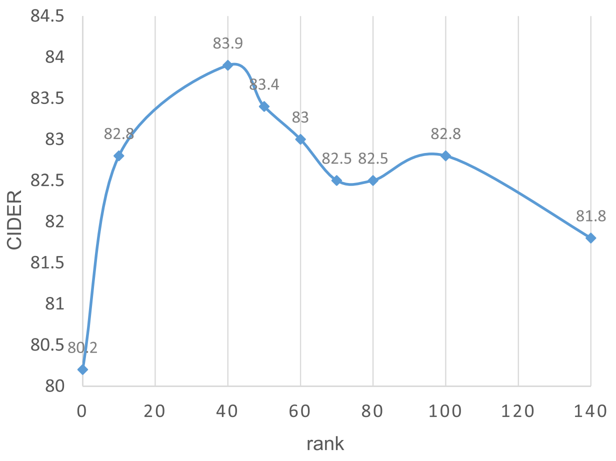


Fig. 7. CIDER scores of ALTM with different ranks.

### *E. The Effectiveness of the Soft Threshold Regression*

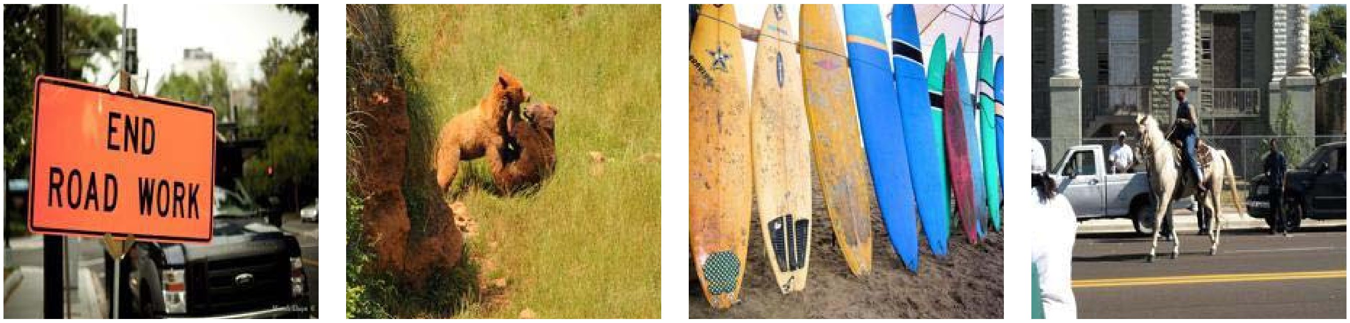
To explore the influence of the activation function on generating spatial attention probabilities, we compare three models with different regression functions in Table V. These models share the same experimental settings with the **Soft Attention** model in Section IV-D. **Softmax** uses the softmax regression which has a positive second-order derivative of $p(x_k)$ w.r.t $x_k$. **Rectifier** uses the following regression function to compute $\mathbf{A}_t^3$:

$$p(x_k) = \frac{max(0, x_k)}{\sum_{j=1}^{K} max(0, x_j)}. \tag{30}$$

Here the second-order derivative of $max(0, x_k)$ w.r.t $x_k, x_k \neq 0$ is zero. **Soft Threshold** uses the soft threshold regression.

In Table V, we can see that the CIDER score increases when the second-order derivatives decrease from **Softmax** to **Rectifier**, to **Soft Threshold**. **Rectifier** performs slightly better than **Softmax** with a zero second-order derivative while **Soft Threshold** leads to large performance gains with a negative second-order derivative.

Figure 4 shows the attention probability distributions produced by the softmax regression and the soft threshold regression. The shown attention probabilities are from 1000 test images in the MS COCO dataset and another 1000 test images from the Flickr30k dataset, respectively. We can see that most attention probabilities are around 0.02 which is the mean value of 49 regions. It's obvious that softmax regression enforces a small portion of probabilities to be very large. As a result, the majority of attention probabilities are smaller than the mean value due to being suppressed by very large ones. On the contrary, probabilities produced by soft threshold have similar quantities on both sides of the mean value. Since indispensable visual clues sometimes scatter over several



(a) (b) (c) (d)

Fig. 8. Some typical failed cases of our model. (a) a sign that says UNK UNK UNK on it. (b) a bear laying on its back in the grass. (c) a row of five surfboards on the beach. (d) a woman riding a horse down a street.

regions, it's important to protect clues in less salient regions from being neglected.

### *F. Visualization of Attention Probabilities*

We visualize the attention probabilities in Figure 5 and Figure 6. We simply upsample $\mathbf{A}_t^3$ of ALTM to the input image size with bilinear interpolation. All samples are from the test split of the MS COCO dataset.

In Figure 5, we visualize an image with caption "two ducks are swimming in the water". The quantity "two" and the category "ducks" naturally correspond with the same regions. Obviously, by only extracting relevant regions, spatial attention mechanism can not attend to details such as quantity, color and shape. Though vanilla attention mechanism knows which region are relevant, it can't pick out relevant details in a single region. In Figure 5, we can see that $\mathbf{A}_t$ varies over time steps, demonstrating that it attends to different weights in the linear transformation.

In Figure 6, we visualize the learned adaptive visual dependence at different time steps. The dependence is measured by the mean value of all the attention probabilities. We can see that ALTM is more dependent on visual information when predicting words with explicit visual meaning. Although there is no direct supervision of how a word is relevant to visual information, the fluctuation of dependence looks quite reasonable. Spatial attention mechanism selects relevant regions and feeds their features into the decoder for every target word. However, not every word has corresponding visual information. Actually, gradients from non-visual words could misguide the caption model [8].

### *G. Failure Cases*

We also show several typical errors made by our model in Figure 8. In the first case, our model cannot recognize the words on the sign. In the second case, the model fails to distinguish objects with intersection. In the third case, the model count the quantity incorrectly. In the forth case, the model mistakes the gender. These cases give us some insights to build more powerful caption models. For example, in the the first case, we can embed text detector into image captioning models. In the third case, we can embed object detector to count the quantity. We will explore these in our future work.

## V. Conclusion

In this paper, we have proposed a novel attention model called attentive linear transformation. ALT attends to the high-dimensional transformation matrix from an image feature space to a context vector space. The advantage of ALT is that the weights in the linear transformation can capture information without a concrete form like spatial region or feature channel. Thus, ALT is able to attend to subtler and more abstract visual concepts than previous attention models. By using the proposed ALT, our caption model achieves state-of-art result on a few widely used benchmarks. Furthermore, ALT can also be applied to other domains such as visual question answering and neural machine translation.

## References

[1] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1171–1179.

[2] X. Chen *et al.* (2015). "Microsoft coco captions: Data collection and evaluation server." [Online]. Available: https://arxiv.org/abs/1504.00325

[3] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3128–3137.

[4] G. Kulkarni *et al.*, "Babytalk: Understanding and generating simple image descriptions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2891–2903, Dec. 2013.

[5] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1–10.

[6] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3156–3164.

[7] Z. Yang, Y. Yuan, Y. Wu, W. W. Cohen, and R. R. Salakhutdinov, "Review networks for caption generation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2361–2369.

[8] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 375–383.

[9] L. Chen *et al.*, "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 5659–5667.

[10] L. Zhou, C. Xu, P. Koch, and J. J. Corso. (2016). "Watch what you just said: Image captioning with text-conditional attention." [Online]. Available: https://arxiv.org/abs/1606.04621

[11] A. Farhadi *et al.*, "Every picture tells a story: Generating sentences from images," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 15–29.

[12] M. Mitchell *et al.*, "Midge: Generating image descriptions from computer vision detections," in *Proc. 13th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2012, pp. 747–756.

[13] A. Aker and R. Gaizauskas, "Generating image descriptions using dependency relational patterns," in *Proc. 48th Annu. Meeting Assoc. Comput. Linguistics*, 2010, pp. 1250–1258.

[14] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi, "Collective generation of natural image descriptions," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics, Long Papers*, vol. 1, 2012, pp. 359–368.

[15] P. Kuznetsova, V. Ordonez, T. L. Berg, and Y. Choi, "TreeTalk: Composition and compression of trees for image descriptions," *Trans. Assoc. Comput. Linguistics*, vol. 2, no. 10, pp. 351–362, 2014.

[16] D. Elliott and F. Keller, "Image description using visual dependency representations," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1292–1302.

[17] H. Fang *et al.*, "From captions to visual concepts and back," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1473–1482.

[18] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *J. Artif. Intell. Res.*, vol. 47, no. 1, pp. 853–899, 2013.

[19] Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik, "Improving image-sentence embeddings using large weakly annotated photo collections," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 529–545.

[20] V. Ordonez, G. Kulkarni, and T. L. Berg, "Im2text: Describing images using 1 million captioned photographs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1143–1151.

[21] R. Kiros, R. Salakhutdinov, and R. Zemel, "Multimodal neural language models," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 595–603.

[22] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. (2014). "Explain images with multimodal recurrent neural networks." [Online]. Available: https://arxiv.org/abs/1410.1090

[23] R. Kiros, R. Salakhutdinov, and R. S. Zemel. (2014). "Unifying visual-semantic embeddings with multimodal neural language models." [Online]. Available: https://arxiv.org/abs/1411.2539

[24] J. Gu, G. Wang, and T. Chen. (2016). "Recurrent highway networks with language cnn for image captioning." [Online]. Available: https://arxiv.org/abs/1612.07086

[25] M. Chen, G. Ding, S. Zhao, H. Chen, Q. Liu, and J. Han, "Reference based lstm for image captioning," in *Proc. AAAI*, 2017, pp. 3981–3987.

[26] C. Liu, F. Sun, C. Wang, F. Wang, and A. Yuille. (2017). "Mat: A multimodal attentive translator for image captioning." [Online]. Available: https://arxiv.org/abs/1702.05658

[27] Q. Wu, C. Shen, L. Liu, A. Dick, and A. van den Hengel, "What value do explicit high level concepts have in vision to language problems?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 203–212.

[28] Y. Pu *et al.*, "Variational autoencoder for deep learning of images, labels and captions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2352–2360.

[29] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4651–4659.

[30] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, "Advanced deep-learning techniques for salient and category-specific object detection: A survey," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 84–100, Jan. 2018.

[31] J. Han, R. Quan, D. Zhang, and F. Nie, "Robust object co-segmentation using background prior," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1639–1651, Apr. 2018.

[32] N. Liu and J. Han, "A deep spatial contextual long-term recurrent convolutional network for saliency detection," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3264–3274, Jul. 2018.

[33] N. Liu, J. Han, T. Liu, and X. Li, "Learning to predict eye fixations via multiresolution convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 2, pp. 392–404, Feb. 2018.

[34] B. Li, Y. Dai, and M. He, "Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference," *Pattern Recognit.*, vol. 83, pp. 328–339, Nov. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320318302097

[35] M. Pedersoli, T. Lucas, C. Schmid, and J. Verbeek, "Areas of attention for image captioning," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 1–9.

[36] W. Chen, J. Wilson, S. Tyree, K. Weinberger, and Y. Chen, "Compressing neural networks with the hashing trick," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2285–2294.

[37] A. Novikov, D. Podoprikhin, A. Osokin, and D. P. Vetrov, "Tensorizing neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 442–450.

[38] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[39] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 652–663, Apr. 2017.

[40] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy, "Improved image captioning via policy gradient optimization of spider," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 873–881.

[41] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 311–318.

[42] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: Semantic propositional image caption evaluation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 382–398.

[43] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proc. 9th Workshop Stat. Mach. Transl.*, 2014, pp. 376–380.

[44] R. Vedantam, C. L. Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4566–4575.

[45] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proc. Text Summarization Branches Out*, vol. 8, 2004, pp. 1–8.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[47] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[48] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

**Senmao Ye** received the B.E. degree from Northwestern Polytechnical University, Xi'an, China, in 2016, where he is currently pursuing the master's degree with the School of Automation. His research interests include computer vision and natural language processing, especially on image captioning and image synthesis.

**Junwei Han** (M'12–SM'15) is a currently a Full Professor with Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision, multimedia processing, and brain imaging analysis. He is an Associate Editor of the IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, *Neurocomputing*, *Multidimensional Systems and Signal Processing*, and *Machine Vision and Applications*.

**Nian Liu** received the B.E. and M.E. degrees from Northwestern Polytechnical University, Xi'an, China, in 2012 and 2015, respectively, where he is currently pursuing the Ph.D. degree with the School of Automation. His research interests include computer vision and multimedia processing, especially on saliency detection and deep learning.