Image Captioning With End-to-End Attribute Detection and Subsequent Attributes Prediction

Yiqing Huang, *Student Member, IEEE*, Jiansheng Chen, *Senior Member, IEEE*, Wanli Ouyang, *Senior Member, IEEE*, Weitao Wan, *Student Member, IEEE*, and Youze Xue, *Student Member, IEEE*

***Abstract*— Semantic attention has been shown to be effective in improving the performance of image captioning. The core of semantic attention based methods is to drive the model to attend to semantically important words, or attributes. In previous works, the attribute detector and the captioning network are usually independent, leading to the insufficient usage of the semantic information. Also, all the detected attributes, no matter whether they are appropriate for the linguistic context at the current step, are attended to through the whole caption generation process. This may sometimes disrupt the captioning model to attend to incorrect visual concepts. To solve these problems, we introduce two end-to-end trainable modules to closely couple attribute detection with image captioning as well as prompt the effective uses of attributes by predicting appropriate attributes at each time step. The *multimodal attribute detector* (MAD) module improves the attribute detection accuracy by using not only the image features but also the word embedding of attributes already existing in most captioning models. MAD models the similarity between the semantics of attributes and the image object features to facilitate accurate detection. The *subsequent attribute predictor* (SAP) module dynamically predicts a concise attribute subset at each time step to mitigate the diversity of image attributes. Compared to previous attribute based methods, our approach enhances the explainability in how the attributes affect the generated words and achieves a state-of-the-art single model performance of 128.8 CIDEr-D on the MSCOCO dataset. Extensive experiments on the MSCOCO dataset show that our proposal actually improves the performances in both image captioning and attribute detection simultaneously. The codes are available at: https://github.com/RubickH/Image-Captioning-with-MAD-and-SAP.**

***Index Terms*— Image captioning, semantic attention, end-to-end training, multimodal attribute detector, subsequent attribute predictor.**

Manuscript received September 26, 2019; revised December 26, 2019; accepted January 20, 2020. Date of publication January 30, 2020; date of current version February 6, 2020. This work was supported by the National Natural Science Foundation of China under Grant 61673234. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Pavan Turaga. *(Corresponding author: Jiansheng Chen.)*

Yiqing Huang, Weitao Wan, and Youze Xue are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: huang-yq17@mails.tsinghua.edu.cn; wwt16@mails.tsinghua.edu.cn; xueyz19@mails.tsinghua.edu.cn).

Jiansheng Chen is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China, and also with the Beijing National Research Center for Information Science and Technology, Beijing 100084, China (e-mail: jschenthu@mail.tsinghua.edu.cn).

Wanli Ouyang is with the School of Electrical and Information Engineering, The University of Sydney, Sydney, NSW 2006, Australia (e-mail: wanli.ouyang@sydney.edu.au).

Digital Object Identifier 10.1109/TIP.2020.2969330

I. Introduction

IMAGE captioning serves as a bridge to link computer vision and natural language processing. It has attracted considerable attention in the artificial intelligence field. It aims at generating descriptions for input images using natural language. Image captioning is practically useful in applications such as human-machine interaction and content based image retrieval, and in systems helping visually impaired people to perceive the world.

In the past several years, image captioning methods have been evolving from vanilla encoder-decoder to including attention, attributes, and reinforcement learning. The widely adopted encoder-decoder framework [1], [2] has shown excellent performance in image captioning. Researchers have later proposed visual attention [3], [4] and semantic attention [5], [6] to further boost the performance of the encoder-decoder network. Generally speaking, visual attention [3], [4] utilizes the extracted spatial feature or object feature to describe the image comprehensively, while semantic attention [5], [6] focuses more on image details by utilizing image attributes. Apart from the introduction of attentions, Rennie *et al.* [7] proposed the Self-Critical-Sequence-Training (SCST) which is a reinforcement learning based approach to directly optimize captioning metrics such as CIDEr-D [8] or Bleu [9].

In this work, we mainly focus on improving the effectiveness of using the image attribute based semantic attention, considering that attributes contain both the high-level knowledge of the image content and specific semantics of corresponding captioning words. Image attributes are commonly chosen from a data-driven vocabulary, which is established by selecting the most commonly used words in the ground truth training captions. Previous image attribute based methods [5], [6], [10]–[12] usually leverage pre-trained deep networks to predict image attributes which are then input to the image captioning network. However, these approaches only use the classification results from attribute classification/detection, leaving the rich semantics of attributes unused.

To effectively utilize the rich semantics of attributes, we introduce a multimodal attribute detector (MAD). The key of MAD is the utilization of attribute embedding, which is trained along with the image captioning network. Benefiting from the rich semantic information contained in the

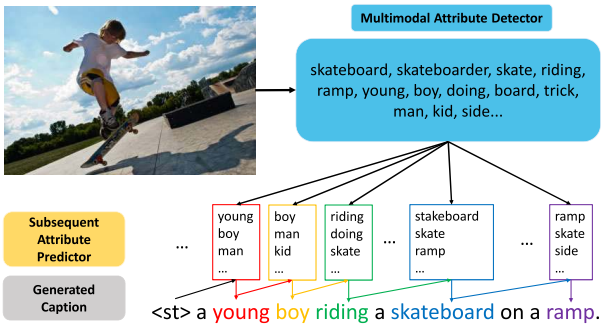


Fig. 1. Our model learns to predict the most appropriate subsequent attributes with the image attribute prior and the previous attribute at each time step for plausible and explainable caption generation. '<*st*>' is short for '<*start token*>', which is the beginning of all the sentences.

embedding of attribute words, better image captioning results can be achieved. Besides, our inclusion of attribute embedding enables MAD to be jointly trained with the image captioning model in an end-to-end manner. The attribute detector and the captioning network are closely coupled and collaborate through the attribute embedding, boosting the performance of both tasks.

Furthermore, other than merely treating the attribute detection as a multi-label classification task, MAD models the similarity between the object feature of the image and the attribute embedding to predict the image attributes. As the attribute embedding contains specific semantic information of corresponding attribute, the image is more likely to contain the attribute of which the embedding is more relevant to the object feature. MAD not only facilitates efficient and accurate image attribute prediction, but also boosts the accuracy of image captioning.

For an input image, multiple attributes reflecting different aspects of the semantic content can be detected. However, at a specific time step of the caption generation, most of these attributes are actually irrelevant to the linguistic context based on which the next word is to be generated. For example in Fig. 1, when the word '*boy*' is to be generated, detected attributes such as '*skateboard*', '*doing*' and '*ramp*' are actually not very helpful. You *et al.* [5] pointed out that these irrelevant attributes may even be harmful by disrupting the model to attend to incorrect visual concepts. Therefore, it is appropriate to exclude these irrelevant attributes when constructing the context. Therefore, an ideal way of utilizing attribute words is to include only those attributes which are closely related to the next word to be generated in the context. To implement this idea, we explore the possibility of estimating a concise attribute subset, in which the attributes are closely related to the current linguistic context with high probabilities. These selected attributes are called subsequent attributes, and the process estimating this subset is called subsequent attribute prediction (SAP). Generally, the proposed SAP is a language model that predicts the most relevant attributes at each time step with the image attribute prior generated by MAD to prevent the attention module from attending to irrelevant attributes. More specifically, at each time step, we record the previous attribute in the generated partial caption and predict the subsequent attributes via a Graph Convolutional Network (GCN). In the GCN, the subsequent attributes are selected according to their transition probability given the previous attribute. As shown in Fig. 1, the proposed SAP module selects the most appropriate subsequent attributes at each time step, leading to more precise and explainable generation results. For example, the word '*boy*' is generated using the semantically relevant subsequent attributes like '*boy*', '*man*', and '*kid*', while the word '*skateboard*' is generated with '*skateboard*', '*skate*' and '*ramp*'. Our proposed SAP only keeps the most appropriate attributes at each time step to benefit the attention module and further boost the performance of caption generation.

By combining MAD and SAP, more precise and appropriate attributes are detected and utilized to provide explicit attribute information in word generation. Our proposal is validated through extensive experiments in which a new state-of-the-art single model performance of 128.8 CIDEr-D on the Karpathy split [13] of MSCOCO dataset [14] is achieved. The main contributions of our work are as follows:

- We model the similarity between the attribute embedding and the object features to facilitate accurate attribute detection.
- We jointly train a multimodal attribute detector with image captioning. As such, the two tasks help each other during the training. Our model achieves state-of-the-art performances with the object feature as the only input.
- We propose SAP for predicting attributes that are highly related to the word to be generated. The predicted subsequent attributes are highly relevant to the current linguistic context, leading to enhancements in both the captioning performance and the semantic explainability of the generation process.

II. Related Work

Image captioning methods have been evolving along with the development of deep neural networks in recent years. Inspired by the success of encoder-decoder framework in the Machine Translation, Vinyals *et al.* [1] firstly proposed the CNN plus RNN encoder-decoder framework in image captioning, which has become dominant in the image captioning field recently. The visual attention mechanism was later introduced to improve the accuracy of the generated captions by effectively utilizing the visual features. Xu *et al.* [3] proposed soft/hard attention mechanism to automatically focus on the salient regions when generating corresponding words. Lu *et al.* [15] argued that adaptively attend to the visual information and the linguistic information is beneficial for image captioning models. Anderson *et al.* [4] combined Bottom-Up attention with Top-Down attention to construct a novel attention based encoder-decoder model. They firstly leveraged the Faster-RCNN [16] to detect a set of object proposals then attended to them in the sentence generation.

Besides utilizing the visual features, the methods leveraging image attributes have also been proved to be effective in improving the quality of the generated sentences [5], [6], [11], [12]. Image attributes are defined as the most important

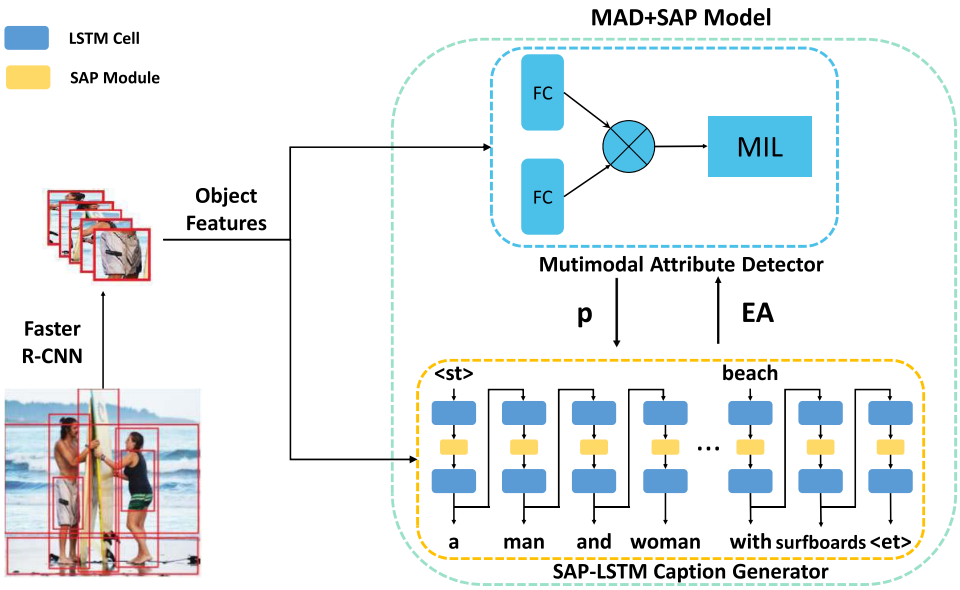


Fig. 2. The overall framework of our proposal. The framework is composed of an end-to-end trained multimodal attribute detector (MAD) and SAP-LSTM caption generator. The SAP module and the LSTM cells are in yellow and blue respectively. The faster R-CNN firstly extracts the object features and then sends them to MAD to predict the probability of image attributes **p** with the help of attribute embedding **EA** (*in Section* III.A). The SAP-LSTM takes in both the object features and **p** to generate plausible image captions (*in Section* III.B *and* III.C). '<*st*>' and '<*et*>' are abbreviations for '<*start token*>' and '<*end token*>' respectively.

words for expressing image semantic information. For example on the MSCOCO dataset [14], the attribute vocabulary is usually established as the 1000 most frequently used words except articles and prepositions in the training captions. This vocabulary covers over 92% of the word occurrences in the training captions [10]. In previous works, the image attributes are usually detected using a pre-trained attribute detector such as a CNN in [10], a FCN [17] in [5], [11] or a modified Faster-RCNN in [4]. These attribute detectors are all trained independently using images and attribute labels, leaving the semantic information embedded in attributes unused. Our proposed MAD additionally exploits the attribute embedding and models the similarity between it and the object features for better detection performance.

Various methods of utilizing image attributes has been proposed to boost the performance of image captioning models. Wu *et al.* [12] sent the attribute confidence vector along with the image feature in the first time step to incorporate attribute information in image captioning. Gan *et al.* [6] introduced a new SCN-LSTM decoder by integrating semantic information with the LSTM cell. You *et al.* [5] firstly proposed semantic attention with the formulation of dot-product to study the similarity between previous predicted word and the image attributes. However, a common problem is that these approaches use all attributes for each time step, in which most attributes are not related to the word to be generated. Utilizing all the detected attributes, as most previous methods did, may induce the model to focus on incorrect semantic information, leading to performance degradation as well as relatively low explainability in captioning. To address this common problem in existing image attribute based models, we propose the SAP module to enable linguistically appropriate attribute subset prediction, leading to more precise and explainable semantic attention and caption generation.

III. Methodology

As is shown in Fig. 2, our model mainly consists of two parts: a multimodal attribute detector (MAD) which can be jointly trained with the captioning model, and a two-layer LSTM caption generator equipped with subsequent attribute predictor (SAP), or SAP-LSTM in short. In the caption generation process, we firstly leverage a ResNet-101 [18] based Faster-RCNN [16] to extract the object features $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N\}, \mathbf{v}_i \in \mathbb{R}^h, \mathbf{V} \in \mathbb{R}^{N \times h}$ from the input image. Then, MAD predicts the attribute probabilities **p** for the input images by using the object features **V** and the attribute embedding **EA** generated by the LSTM layers (*in Section* III.A). Finally, the SAP module (*in Section* III.B) is inserted into the two-layer LSTM caption generator to generates plausible image captions using **p** and **V** (*in Section* III.C).

While previous attribute based captioning models usually take both image features and attribute detection results as inputs, our model only leverages the object features since the proposed MAD is jointly trained end-to-end with the captioning network. The attribute information is effectively exploited from these object features through joint training to achieve state-of-the-art performances for both tasks. Detailed implementation of these our method will be elaborated in the rest of this section.

A. Multimodal Attribute Detector

Previous image captioning methods incorporating attributes usually demand the training of a CNN based attribute detector independently outside the image captioning network [5], [10], [11]. As such, the attribute detection is usually modeled as a multi-source domain generalization task [19] or a multi-label classification task and only the visual features can be exploited as the input of the attribute

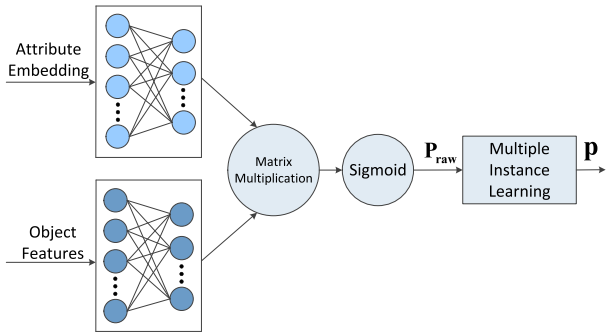


Fig. 3. The proposed multimodal attribute detector models the similarity between the attribute embedding **EA** and object features **V** via matrix multiplication to predict the attribute probability distribution.

detector. However, unlike class labels in the classification task, image attributes are essentially words containing explicit semantic information. We therefore argue that leveraging these semantic information will probably benefit attribute detection. Based on this understanding, we additionally utilize the attribute embedding **EA**, which is generated as a byproduct by the decoding LSTM in most captioning networks, to model the similarity between the object feature and the attributes in the detection process. As is illustrated in Fig. 3, the probability of whether an attribute is contained in the input image is predicted in two steps by the proposed MAD module. In the first step, the attribute embedding and the object features are mapped to the same space using two fully connected layers, of which the outputs are then merged using matrix multiplication to compute the similarity. The product is then sent to a sigmoid layer to generate the raw probability matrix $\mathbf{P}_{raw} \in \mathbb{R}^{1000\times N}$, where P^{ij}_{raw} denotes the probability that the j^{th} object contains the i^{th} attribute a_i. The mathematical formulation of the first step is in Eq. 1, where $\mathbf{W}_{attr} \in \mathbb{R}^{d\times g}$ and $\mathbf{W}_v \in \mathbb{R}^{d\times h}$ are trainable parameters, $\mathbf{E} \in \mathbb{R}^{g\times k}$ denotes the embedding of all the words in the vocabulary, $\mathbf{A} \in \mathbb{R}^{k\times 1000}$ is the one-hot index matrix of the 1000 attributes, $\otimes$ denotes the matrix multiplication, and the superscript T is the transpose operation.

$$\mathbf{P}_{raw} = sigmoid((\mathbf{W}_{attr}\mathbf{EA})^T \otimes \mathbf{W}_v\mathbf{V}^T) \tag{1}$$

In the second step, the probability values in each row of P_{raw} are merged using the noisy-OR Multiple Instance Learning (MIL) model proposed in [10] to predict the final probability p_i that the input image contains the i^{th} attribute a_i. The above procedure is illustrated as Eq. 2.

$$p_i = 1 - \prod_{j=0}^{N}(1 - P^{ij}_{raw}) \tag{2}$$

The proposed MAD module shows remarkable applicability for it only leverages the visual features and the attribute embedding, both of which are already included in most image captioning networks. As such, it can be trained end-to-end along with the image captioning network. Moreover, compared to the pre-trained attribute detectors in previous works [3], [11], [12], MAD shares the image feature with the image captioning network so that the overall number of model parameters can be efficiently reduced.

The focal loss proposed in [20] is leveraged for training MAD as is shown in Eq. 3, where l_i denotes whether a_i is in the ground-truth captions or not, and δ and γ are empirically set to 0.95 and 2.

$$\begin{aligned} &loss_i^{fl} \\ &= -l_i\delta(1-p_i)^\gamma log(p_i) - (1-l_i)(1-\delta)p_i^\gamma log(1-p_i) \end{aligned} \tag{3}$$

Similar to [20], the loss generated for each attribute is then normalized by the number of positive attributes N_{pos} to supervise the training of MAD as in Eq. 4. The probabilities predicted by this detector is later input as the prior probability to the subsequent attribute preditor (SAP) module for predicting subsequent attributes at each time step. Details are elaborated in the next section.

$$loss_{MAD} = \frac{1}{N_{pos}}\Sigma_{i=1}^{1000} loss_i^{fl} \tag{4}$$

B. GCN Based Subsequent Attribute Predictor

Another essential component in our proposal is the subsequent attribute predictor (SAP), which is introduced to predict an appropriate subset of attributes, or subsequent attributes, to attend to at each time step. Fig. 4 shows the difference between the traditional way of utilizing image attributes and the proposed subsequent attributes based method. Without using SAP, the model attends to a fixed attribute set at all time steps as is shown on the top of Fig. 4. However, the attributes in this set may be diverse significantly in semantics. For example, *'young'* is not relevant to *'giraffe'*, *'standing'* or *'zoo'* according to common human understanding. Therefore, it is quite ambiguous how these irrelevant attributes may affect the network in the generation of *'young'*. SAP is proposed to deal with such ambiguity. It can be observed that the predicted subsequent attributes vary with the context at each time step and most of the subsequent attributes are closely related to the word to be generated. Hence, *'young'* is generated with relevant attributes like *'small'*, *'boy'*, and *'child'*. Such phenomenon indicates that utilizing subsequent attributes leads to better explainability than utilizing the fixed attribute set. The framework of SAP is shown in Fig. 5 of which details are illustrated as follows.

1) Previous Attribute: In this paper, we define the previous attribute as the last attribute word in the partial caption that has been already generated. Specifically, '$\langle st\rangle$' is used as the initial previous attribute for all the sentences. In the inference stage, as is shown in Fig. 4, '$\langle st\rangle$' is the previous attribute when generating *'a'* and *'young'*. Once the attribute word *'young'* is generated, it replaces '$\langle st\rangle$' as the new previous attribute. The definition of the previous attribute is slightly different in the training stage when using the cross-entropy loss since only word probabilities are generated during training. As such, previous attribute changes if the input word $\mathbf{y}_{t-1}$ in the ground truth caption is an attribute word.

2) Graph Convolutional Network: We leverage a Graph Convolutional Network (GCN) to update the embedding of the attributes by propagating the transition probability among all attributes. The GCN takes the embedding of all attributes **EA**) $\in \mathbb{R}^{g\times 1000}$ and the transition probability matrix

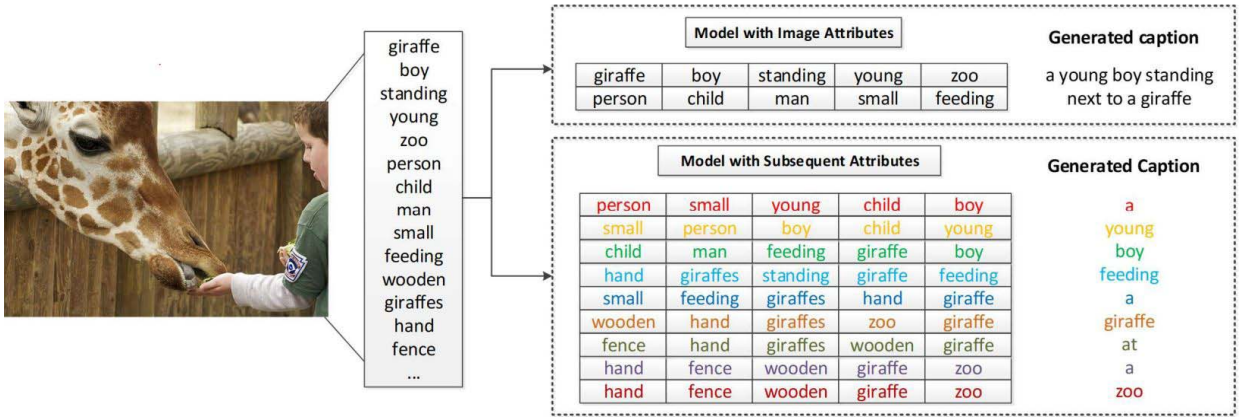


Fig. 4. Comparison of models exploiting image attributes and subsequent attributes. The traditional image attribute based methods attend to the same attribute set at all time steps. While the subsequent attributes vary at each time step. The subsequent attributes and the word generated at the same time step are shown in the same color. The subsequent attributes are arranged from right to left in descending order of probability.

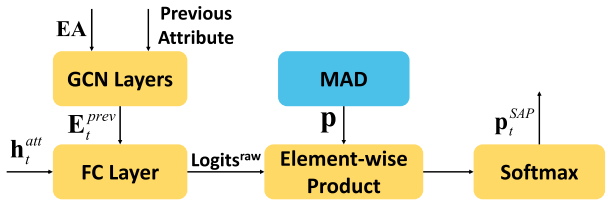


Fig. 5. SAP predicts subsequent attributes using the attribute embedding **EA**, previous attribute in the partial caption, the output of the attention LSTM ($\mathbf{h}_t^{att}$), and the attribute probability distribution $\mathbf{p}$ predicted by MAD.

$\mathbf{U} \in \mathbb{R}^{1000\times1000}$ as inputs, and updates the embedding to $\tilde{\mathbf{E}_A} \in \mathbb{R}^{1000\times g}$. We follow [21] to construct the GCN layer, of which the mathematical formulation is shown in Eq. 5, where $\mathbf{W}^l \in \mathbb{R}^{g\times g}$ is the parameter of the fully connected layer, and $f(\cdot)$ is the LeakyRelu [22] activation function.

$$\tilde{\mathbf{E}_A} = f(\mathbf{U}(\mathbf{EA})^{\mathbf{T}}\mathbf{W}^{\mathbf{l}}) \tag{5}$$

A crucial part of the GCN is the transition probability matrix $\mathbf{U}$, which is established in a data-driven way in this work. We count the neighboring attributes pairs in the ground truth training captions to get the transition frequency matrix $\mathbf{U}^{fre}$. For example, given a ground truth caption '$\langle$ *st* $\rangle$ *a man is driving a car.*' and the corresponding attributes '$\langle st \rangle$', '*man*', '*driving*', and '*car*', the neighboring attribute pairs are ('$\langle st \rangle$' and '*man*), ('*man*' and '*driving*'), ('*man*' and '*car*'). To mitigate the influence of frequent attributes, we normalize the $\mathbf{U}^{fre}$ with the occurrence of each attribute to calculate the normalized matrix $\mathbf{U}^{norm}$ as is shown in Eq. 6, where N_i is the occurrence of the i^{th} attribute in the training set.

$$U_{ij}^{norm} = U_{ij}^{fre}/N_i \tag{6}$$

The final transition probability matrix $\mathbf{U}$ is achieved by normalizing each row of $\mathbf{U}^{norm}$ as is shown in Eq. 7.

$$\mathbf{U}_i = \mathbf{U}_i^{norm}/(\Sigma_{j=1}^{1000}U_{ij}^{norm}) \tag{7}$$

The i^{th} row of $\mathbf{U}$ indicates the conditional probability distribution of the subsequent attributes when the i^{th} attribute is the current previous attribute. To transit the attribute information, we adopt two GCN layers, of which the output can be viewed as the refined features for the attributes. These features contain the subsequent information for each attribute. Consequently, the transited information of the previous attribute at the time step t, denoted as $\mathbf{E}_t^{prev}$, is selected for predicting the subsequent attributes.

3) Implementation of SAP: We concatenate $\mathbf{E}_t^{prev}$ with the output of attention LSTM, namely $\mathbf{h}_t^{att}$, and fed the concatenation result to a fully connected layer to get the raw logits as is shown in Eq. 8, where $\mathbf{W}_s \in \mathbb{R}^{(g+g)\times1000}$ and $\mathbf{b}_s \in \mathbb{R}^{1000}$ are trainable parameters.

$$\mathbf{logits}^{raw} = \mathbf{W}_s[\mathbf{E}_t^{prev}; \mathbf{h}_t^{att}] + \mathbf{b}_s \tag{8}$$

The attribute detection probability $\mathbf{p}$, namely the output of MAD, is input to SAP as the prior of image attributes. In practice, $\mathbf{p}$ is leveraged as the weight of each corresponding logit. As shown in Eq. 9, the element-wise product between $\mathbf{p}$ and the raw logits are input to the softmax layer to generate $\mathbf{p}_t^{SAP}$, which is the probability distribution of the subsequent attributes. Thus, the transited information of the previous attribute, the current linguistic context, and the prior of image attribute are all taken into account in the prediction of subsequent attributes. Cross-entropy loss is adopted to supervise the training of SAP module as shown in Eq. 10, where a_t^s is the index of subsequent attribute at time step t.

$$\mathbf{p}_t^{SAP} = softmax(\mathbf{p} \odot \mathbf{logits}^{raw}) \tag{9}$$

$$loss_{SAP} = \frac{1}{T}\sum_{t=1}^{T} -log(\mathbf{p}_t^{SAP}(a_t^s|a_1^s, a_2^s \ldots a_{t-1}^s) \tag{10}$$

Intuitively, the attribute with the highest probability can be directly selected as the subsequent attribute. However, we notice in the experiments that most top-ranked attributes at each time step are also closely related to the current linguistic state of the sentence. Consequently, rather than selecting only the top-1 attribute, we select top-K attribute as subsequent attributes $\mathbf{A}_t$. Selecting more appropriate attributes also prevents our model from attending to the wrong attribute when the top-1 subsequent attribute is erroneous.

Our proposed SAP assists the semantic attention module to generate precise and explainable attention as it dynamically selects the appropriate subsequent attributes at each time step. Consequently, the semantic attention in our framework can be viewed as a trade-off between hard attention and soft attention. Comparing with hard attention, which only exploits the information of one attribute at each time step, we feed more relevant information to the LSTM to enrich the semantic information. Comparing with soft attention, which takes all the detected attributes into consideration at all time steps, we selectively choose a concise attribute set at each time step to mitigate linguistic ambiguity caused by irrelevant attributes.

C. SAP-LSTM

A two-layer LSTM is leveraged as the decoder, where the first LSTM layer and the second LSTM layer are called the attention LSTM and language LSTM respectively. The implementation of these two LSTM layers are similar to that proposed in [4]. In our proposal, the attention LSTM not only helps to establish top-down attention, but also helps the subsequent attributes prediction. The mean pooled object feature $\bar{\mathbf{v}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{v_i}$, the embedding of the input word $\mathbf{E}\mathbf{y}_{t-1}$, and the output of the language LSTM at the last time step $\mathbf{h}_{t-1}^{lan}$ are firstly concatenated and then input to the attention LSTM. The output of the attention LSTM, namely $\mathbf{h}_t^{att}$, is employed to distinguish the importance of different object features $\mathbf{v}_i$ to generate the visual context $\hat{\mathbf{v}}_t$. The implementation of visual attention is similar to that in [4], the details are illustrated in Eq. 11, where $\mathbf{w}_a \in \mathbb{R}^d$, $\mathbf{W}_{vh} \in \mathbb{R}^{d\times g}$, $\mathbf{W}_v \in \mathbb{R}^{d\times h}$ are parameters to be learned. The i^{th} element of the vector $\mathbf{n}_t$ and $\boldsymbol{\alpha}_t$ are denoted as n_t^i and α_t^i respectively.

$$\begin{aligned} n_t^i &= \mathbf{w}_a tanh(\mathbf{W}_{vh}\mathbf{h}_t^{att} + \mathbf{W}_v\mathbf{v}_i) \\ \boldsymbol{\alpha}_t &= softmax(\mathbf{n}_t) \\ \hat{\mathbf{v}}_t &= \sum_{i=1}^{N}(\alpha_t^i * \mathbf{v}_i) \end{aligned} \tag{11}$$

The above procedure is shown in the left part of Fig. 6. The rest of Fig. 6 is illustrated as follows. The output of the attention LSTM, namely $\mathbf{h}_t^{att}$, is also fed to the subsequent attribute predictor along with the previous attribute in the partial caption to predict the most appropriate subsequent attributes. These attributes are then input to the semantic attention module to generate the explicit semantic context $\hat{\mathbf{a}}_t$. Eq. 12 describes such a process, where $\mathbf{A}_t = \{\mathbf{A}_t^1, \mathbf{A}_t^2, \ldots \mathbf{A}_t^K\}$ is the set of one-hot index vectors of the subsequent attributes predicted by SAP, and $\mathbf{E}$ is the word embedding matrix shared with MAD. Other notations are similar to that in Eq. 11.

$$\begin{aligned} m_t^j &= \mathbf{w}_b tanh(\mathbf{W}_{ah}\mathbf{h}_t^{lan} + \mathbf{W}_a\mathbf{E}\mathbf{A}_t^j)) \\ \boldsymbol{\beta}_t &= softmax(\mathbf{m}_t) \\ \hat{\mathbf{a}}_t &= \sum_{j=1}^{K}(\beta_t^j * \mathbf{E}\mathbf{A}_t^j) \end{aligned} \tag{12}$$

The concatenation of $\hat{\mathbf{v}}_t$, $\hat{\mathbf{a}}_t$, and h_t^{att} is then fed to the language LSTM to generate h_t^{lan}, which is input to a fully connected layer to generate logits. Finally, the output logits

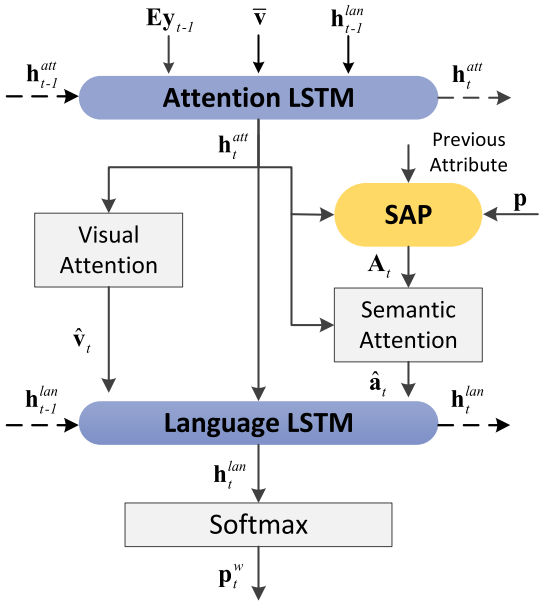


Fig. 6. The proposed SAP module is employed in the two-layer LSTM to predict appropriate subsequent attributes $\mathbf{A}_t$ at each time step to provide explicit semantic context $\hat{\mathbf{a}}_t$ in the word generation process.

are sent to a softmax layer to generate the probability for each word in the vocabulary as is in Eq. 13, where $\mathbf{W}_p \in \mathbb{R}^{k\times g}$ and $\mathbf{b}_p \in \mathbb{R}^k$ are trainable parameters. The overall working flow of the LSTMs can be formulated by Eq. 14 and Eq. 15, in which $LSTM_{att}$ and $LSTM_{lan}$ denote the attention LSTM and the language LSTM respectively, different $\mathbf{h}$s are the outputs of different LSTM layers at different time steps.

$$\mathbf{p}_t^w = softmax(\mathbf{W}_p\mathbf{h}_t^{lan} + \mathbf{b}_p) \tag{13}$$

$$\mathbf{h}_t^{att} = LSTM_{att}(\mathbf{h}_{t-1}^{att}, [\mathbf{h}_{t-1}^{lan}; \bar{\mathbf{v}}; \mathbf{E}\mathbf{y}_{t-1}]) \tag{14}$$

$$\mathbf{h}_t^{lan} = LSTM_{lan}(\mathbf{h}_{t-1}^{lan}, [\mathbf{h}_t^{att}; \hat{\mathbf{v}}_t; \hat{\mathbf{a}}_t]) \tag{15}$$

D. Loss Functions

Our model is trained in two phases. In the first phase, the traditional cross-entropy (XE) loss between the ground truth caption and word probability output is adopted as in Eq. 16.

$$loss_{XE}^w = \frac{1}{T}\sum_{t=1}^{T} -log(\mathbf{p}_t^w(y_t|y_{1:t-1}, \mathbf{V})) \tag{16}$$

In the second phase, the model is optimized for the CIDEr-D metric using the Self-Critical Sequence Training (SCST) [7]. The target is to minimize the negative expected score corresponding to the model parameter θ as shown in Eq. 17, where r is the CIDEr-D score of the sampled sentence $w_{1:T}$.

$$loss_{RL}^w = -E_{w_{1:T}\sim\theta}[r(w_{1:T})] \tag{17}$$

The gradient of $loss_{RL}^w$ can be approximated as Eq. 18, where $r(w_{1:T}^s)$ and $r(\hat{w}_{1:T})$ are the CIDEr rewards for the random sampled sentence and the max sampled sentence.

$$\nabla_\theta loss_{RL}^w = -(r(w_{1:T}^s) - r(\hat{w}_{1:T}))\nabla_\theta log(\mathbf{p}^w(w_{1:T}^s)) \tag{18}$$

The overall loss function is a linear combination of the word loss $loss^w$, $loss_{MAD}$, and $loss_{SAP}$ as shown in Eq. 19. $loss_{MAD}$ and $loss_{SAP}$ are defined in Eq. 4 and Eq. 10 respectively. The word loss $loss^w$ equals $loss^w_{XE}$ for the first training phase, and $loss^w_{RL}$ for the second.

$$loss = loss^w + 0.2 * loss_{MAD} + 0.2 * loss_{SAP} \quad (19)$$

IV. Experiments

A. Dataset and Experimental Settings

1) Dataset and Evaluation Metrics: We conduct our experiments and evaluated model performances on the MSCOCO captioning dataset [14]. Beam search strategy is adopted in the inference state and we empirically set the beam size to 2. For offline evaluation, we use 5000 images for validation and 5000 images for testing from the 40504 validation set following the widely adopted Karpathy's data split [13].

To evaluate the image captioning performance, the following metrics are used: Bleu [9], Meteor [23], Rouge-L [24], CIDEr-D [8] and SPICE [25]. These metrics are computed with the code[1] released by the MSCOCO test server. Besides image captioning, we also evaluate the attribute detection performance of our proposal using the the $F1$ score.

2) Features and Parameter Settings: We adopt ResNet-101 based faster R-CNN to extract the object features with the size of 36×2048 as was practiced in [4]. We select the words that appear in the training set for over 5 times to form our vocabulary which is finally of the size 9487. Each word is then embedded to a 1000 dimensional word embedding space. Similar to [6], we select the top-ranked 1000 words to establish our attribute vocabulary. Attributes share the same embedding with the corresponding words. The number of subsequent attributes is empirically set to K = 10 in this work. Contexts and outputs of LSTMs are mapped to 1000 dimensional spaces for both the visual attention and the semantic attention. The hidden size of the two-layer LSTM is set to 1000. The aforementioned constants d,g,h,k in the equations are set to 512, 1000, 2048 and 9487 respectively.

3) Training Details: The models are implemented with the PyTorch[2] framework. Adam optimizer [26] is utilized to minimize the loss function in Eq. 19. The models are trained with cross-entropy (XE) loss for 50 epochs in the first phase and with SCST for another 100 epochs in the second phase. For both training phases, the batch size is set to 200. The rate of scheduled sampling [27] is linearly increased by 5% every 5 epochs until 25% in cross-entropy training. The learning rate is initially set to 5e-4 and decays by a factor of 0.8 every 5 epochs in the first phase and every 10 epochs in the second phase. We set the dropout ratio to 0.5 in our models and clip the gradients by the maximum absolute value of 0.1.

B. Attribute Detection Results

We evaluate the attribute detection performance of the jointly trained MAD on the MSCOCO test split. The precision and recall of attribute detection for each image are defined in Eq. 20, in which $\mathbf{A}_{gt}$ is the set of ground truth attributes of an image, $\mathbf{A}_d$ is the set of top-M detected attributes, and $|\cdot|$ represents the cardinality of a set. The $F1$ score is defined as the harmonic mean of precision and recall as in Eq. 21. Higher value of $F1$ indicates better detection performance.

$$precision = \frac{|\mathbf{A}_{gt} \bigcap \mathbf{A}_d|}{|\mathbf{A}_{gt}|}, recall = \frac{|\mathbf{A}_{gt} \bigcap \mathbf{A}_d|}{|\mathbf{A}_d|} \quad (20)$$

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \quad (21)$$

Average $F1$ scores over all the images in Karpathy's MSCOCO test split are reported. Table I compares the detection performance of MAD with the state-of-the-art pre-trained attribute detector proposed in [6]. It can be seen that MAD outperforms the pre-trained detector for different values of M, suggesting that through joint training with the captioning model, our proposed attribute detector is not only efficient but also accurate. It can be noticed that the $F1$ score of MAD is less sensitive with regard to the value of M, indicating the robustness of our proposal.

TABLE I
$F1$ Scores of Two Attribute Detectors With Top-M Attributes

M	5	10	15	20	25
SCN-LSTM [6]	0.278	0.411	0.432	0.412	0.372
MAD+SAP	0.338	0.423	**0.446**	0.442	0.426

C. Image Captioning Results

We compare our method with recent state-of-the-arts including SCN-LSTM [6], SCST [7], LSTM-A [11], Stack-Cap [28], Up-Down [4], CAVP [29] and SGAE [30]. Among these methods, SCST [7] proposes the advanced reinforcement learning (RL) based training methods for image captioning. SCN-LSTM [6] and LSTM-A [11] are attribute based methods, in which pre-trained attribute detectors are utilized to generate the high-level semantic information. Stack-Cap [28] introduces a more complex RL-based reward and utilizes the layerwise information between the multiple LSTM layers. Up-Down [4] proposes the two-layer LSTM architecture which is adopted in our work as the baseline decoder. CAVP [29] proposes to learn pairwise relationships in the decoder as well as a new RL-based reward. SGAE [30] incorporates the language inductive bias into the encoder-decoder framework by leveraging the scene graph of the image and a trained dictionary. The offline performances of our proposed MAD+SAP method and other compared methods are listed in Table II. Our model outperforms all the compared methods in most metrics, especially in terms of CIDEr-D and Meteor, indicating the effectiveness of our proposed method. Comparing with SCST and Up-Down, our model utilizes high-level semantic information to generate more fine-grained captions. While SCN-LSTM and LSTM-A require the pre-training of an attribute detector before implementing the image captioning network, our proposed MAD is closely coupled with the image captioning model to achieve better image captioning performance. Stack-Cap refines the image descriptions output

[1]https://github.com/tylin/coco-caption
[2]https://pytorch.org/

TABLE II

SINGLE MODEL IMAGE CAPTIONING PERFORMANCE (%) ON THE COCO TEST SPLIT, '–' MEANS THE AUTHORS DID NOT EVALUATE UNDER SUCH METRIC. THE IMAGE FEATURES THAT EACH METHOD UTILIZES ARE SHOWN IN THE PARENTHESES, WHERE I-VN, R-N AND F DENOTES THE N^{th} VERSION OF INCEPTION NETWORK, N-LAYER RESNET, RESNET-101 BASED FASTER R-CNN AND SCENE GRAPH FEATURES RESPECTIVELY. THE LARGEST NUMBER IN EACH COLUMN IS MARKED IN BOLDFACE

Methods	Cross-Entropy (XE) Loss					CIDEr-D Optimization				
Metric	Bleu-4	Meteor	Rouge-L	CIDEr-D	SPICE	Bleu-4	Meteor	Rouge-L	CIDEr-D	SPICE
SCN-LSTM (R-152) [6]	34.1	26.1	-	104.1	-	-	-	-	-	-
SCST (R-101) [7]	30.0	25.9	53.4	99.4	-	34.2	26.7	55.7	114.0	-
LSTM-A (I-V3) [11]	35.2	26.9	55.8	108.8	20.0	35.5	27.3	56.8	118.3	20.8
Stack-Cap (R-101) [27]	35.2	26.5	-	109.1	20.3	36.1	27.4	56.9	120.4	20.9
Up-Down (F) [4]	36.2	27.0	56.4	113.5	20.3	36.3	27.7	56.9	120.1	21.4
CAVP (F) [28]	-	-	-	-	-	**38.6**	28.3	58.5	126.3	21.6
SGAE (F + SG) [29]	36.9	27.7	**57.2**	116.7	20.9	38.4	28.4	**58.6**	127.8	22.1
MAD+SAP (F) (ours)	**37.0**	**28.1**	**57.2**	**117.3**	**21.3**	**38.6**	**28.7**	58.5	**128.8**	**22.2**

TABLE III

PERFORMANCE (%) ON THE ONLINE MSCOCO EVALUATION SERVER. TOP-2 RANKINGS ARE INDICATED BY RED SUPERSCRIPT FOR EACH METRIC

Methods	Bleu-1		Bleu-2		Bleu-3		Bleu-4		Meteor		Rouge-L		CIDEr-D	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
SCN-LSTM [6]	74.0	91.7	57.5	83.9	43.6	73.9	33.1	63.1	25.7	34.8	54.3	69.6	100.3	101.3
SCST [7]	78.1	93.7	61.9	86.0	47.0	75.9	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.7
LSTM-A [11]	78.7	93.7	62.7	86.7	47.6	76.5	35.6	65.2	27.0	35.4	56.4	70.5	116.0	118.0
Stack-Cap [27]	77.8	93.2	61.6	86.1	46.8	76.0	34.9	64.6	27.0	35.6	56.2	70.6	114.8	118.3
Up-Down [4]	80.2[2]	95.2[1]	64.1	88.8[2]	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
CAVP [28]	80.1	94.9[2]	64.7[2]	88.8[2]	50.0[2]	79.7[2]	37.9	69.0	28.1	37.0	58.2	73.1	121.6	123.8
SGAE [29]	-	-	-	-	-	-	38.5[1]	69.7[1]	28.2[2]	37.2[2]	58.6[2]	73.6[1]	123.8[2]	126.5[2]
MAD+SAP	80.5[1]	94.9[2]	65.1[1]	89.1[1]	50.4[1]	80.0[1]	38.4[2]	69.4[2]	28.6[1]	37.7[1]	58.7[1]	73.3[2]	125.1[1]	127.0[1]

from the shallow LSTM layer to generate better caption, while SAP leverages the output of attention LSTM to predict the appropriate subsequent attributes to boost the performance of language LSTM. CAVP proposes a novel RL-based training strategy which is likely to be complementary with our method. Comparing with SGAE, which requires the pre-training of a scene graph generator and a dictionary, our method outperforms it in most metrics yet takes the visual feature as the only input. Our improvements over these compared methods are largely due to the more accurate attribute detection as well as the more proper selection of attributes to attend to. As such, more explicit semantic context can be constructed to facilitate the generation of more precise image captions.

We notice that our MAD+SAP model outperforms the backbone Up-Down model more significantly in the CIDEr-D optimization phase than that in the cross-entropy training phase. Such a phenomenon indicates the effectiveness of incorporating appropriate semantic information into CIDEr-D optimized models. Image attribute words are generally more semantically important than non-attributes such as 'the' and 'is'. Hence they are usually assigned with higher weights than non-attributes in the TF-IDF based CIDEr-D metric. When CIDEr-D is directly optimized, the gain of selecting proper subsequent attributes is much more prominent than that in the cross-entropy based training in which the weights of attribute and non-attribute are essentially equal. Fig. 7 compares the semantic attention weights of MAD+SAP models in XE phase and SCST phase. In the SCST phase, the weights of the attributes are more concentrated than that in XE phase. For instance, the weights of attributes *'holding'* and *'rackets'* are much larger when generating corresponding words in the SCST model than in the XE model.

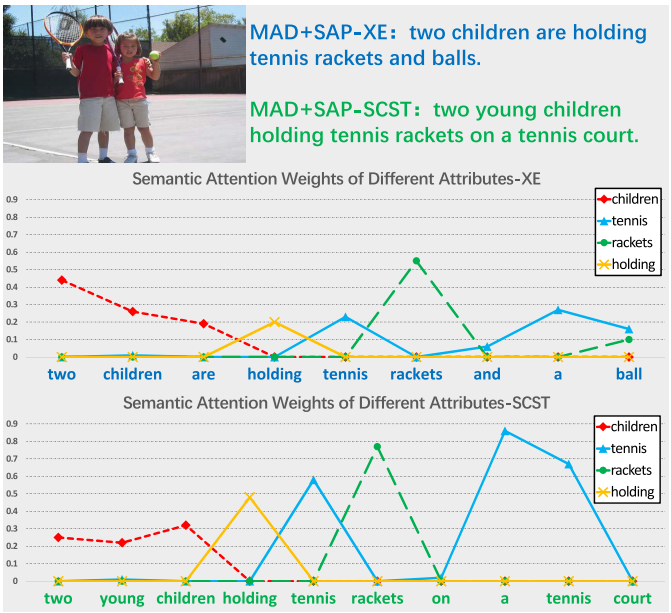


Fig. 7. Comparison of semantic attention weights of MAD+SAP model in XE phase and SCST phase.

We also evaluate the ensembled MAD+SAP model on the online MSCOCO test server.[3] Table III reports the performance of MAD+SAP and other state-of-the-art methods.

[3]https://competitions.codalab.org/competitions/3221/#results

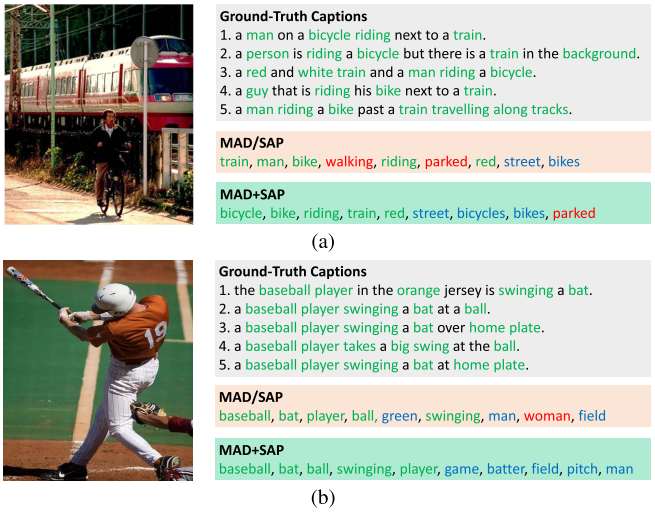


Fig. 8. Two image samples with attribute detection results of MAD/SAP (not end-to-end) and MAD+SAP (end-to-end). Green words are the positive attributes. Blue words are reasonable attributes. Red words are the inappropriate attributes. The attributes are arranged from left to right in a descending order of probability. Both sub-figure (a) and (b) show that the attributes can be better detected with end-to-end training.

We can see that our proposed method also achieve the best performance in most metrics, especially in terms of Meteor and CIDEr-D. It is worth noticing that our model outperforms the baseline Up-Down models significantly in terms of the CIDEr-D metric, and achieves relatively better performance than the recently proposed SGAE method.

The above statistical results suggest that our model generally out performs all the compared methods in both offline and online evaluations, indicating the effectiveness of leveraging appropriate subsequent attributes.

D. Ablation Study

We conduct extensive experiments to verify the influences of different modules in our proposal.

1) Influence of End-to-end Training: A key observation for our method is that end-to-end training the attribute detector with the image captioning network is beneficial for both tasks. Here we present two models, where the first is our proposed MAD+SAP model. Another model (MAD/SAP) follows the traditional way of independently training an attribute detector before training the captioning model. Note that in the model (MAD/SAP), attribute embedding is not shared between MAD and SAP. Fig. 8 shows the attribute detection results of these two models. It can be observed that without explicit attribute embedding in model (MAD/SAP), some of the detected attributes with high confidence are actually not very appropriate for the image content. However, by using end-to-end trained attribute embedding, the proposed MAD module is able to generate a much more reasonable probabilistic prior for the image attributes. Some of the high probability attributes predicted by MAD fit the image content so well that they can even be readily used to replace the corresponding ground truth attribute words in the captions. Actually, leveraging attribute embedding facilitates automatic incorporation of sentence information in attribute detection, so that the semantic information of image attributes can be sufficiently exploited. This probably explains why we have achieved better attribute detection results with a much simpler network than those used in previous works [5], [6], [12].

TABLE IV

F1 SCORE AND IMAGE CAPTIONING PERFORMANCE IN TERMS OF BLEU-4(B@4), METEOR(M), ROUGE-L(R), CIDER-D(C) AND SPICE(S) ON THE COCO TEST SPLIT IN XE TRAINING. † DENOTES THAT THIS MODEL IS NOT END-TO-END TRAINED. THE LARGEST NUMBER IN EACH COLUMN IS MARKED IN BOLDFACE

Methods	F1	B@4	M	R	C	S
MAD/SAP†	0.421	36.7	27.8	57.0	115.6	21.1
MAD/SAP-G†	0.437	36.7	20.0	57.1	116.2	21.2
MAD+SAP	0.446	**37.0**	**28.1**	57.2	117.3	**21.3**
MAD+SAP-G	**0.448**	**37.0**	**28.1**	**57.3**	**117.4**	**21.3**

We then statistically validate the importance of end-to-end training by showing the F1 score and image captioning performance in Table IV. We also ablate our model by utilizing the pre-trained GloVe [31] to initialize the word embedding in MAD to verify the influence of shared embedding. The models with the GloVe word embedding are suffixed with 'G'. Incorporating the pre-trained GloVe is beneficial for both MAD/SAP and MAD+SAP models. However, it is obvious that the shared embedding in the end-to-end training yields much more improvement than utilizing the GloVe. This is because GloVe is pre-trained only on language corpus, it may not be very appropriate to be directly adopted in MAD, which predicts the image attribute by modeling the similarity between the embedding and the visual features. However, the embedding in SAP-LSTM is trained in image captioning, which is the task that thoroughly couples the embedding with the visual features. Thus this end-to-end trained embedding is able to facilitate better detection performance in MAD. Consequently, fine-tuning the GloVe in MAD+SAP-G model yields the best performance in attribute detection and leads to the best image captioning performance. Considering that the other compared methods [4], [29], [30] did not use pre-trained word embedding, we do not use GloVe in any of the other comparison results.

2) Influences of MAD and SAP: For quantifying the influence of our proposed MAD and SAP, we ablate our model with the following parts: **base**: We follow [4] to build the two-layer LSTM, which is the state-of-the-art model employing bottom-up attention mechanisms, to form our baseline. **base+MAD**: We add the end-to-end trained multimodal attribute detector to extract high-level semantic information from images. The semantic attention is applied to the top-15 attributes. **base+SAP**: We add the subsequent attribute predictor to predict the appropriate subsequent attribute. Note that **p** in Eq. 9 is removed since MAD is not utilized in this model. **MAD+SAP**: our full model with both MAD and SAP. The performances of the ablated models are shown in Table V. Our base model shows comparable performance with that proposed in [4]. Both MAD and SAP improves the model performance significantly by incorporating either accurate image attribute or semantically appropriate subsequent attributes to feed semantic

TABLE V

SINGLE MODEL IMAGE CAPTIONING PERFORMANCE ON THE COCO TEST SPLIT. THE LARGEST NUMBER IN EACH COLUMN IS DISPLAYED IN BOLD

Methods	Bleu-4	Meteor	Rouge-L	CIDEr-D	SPICE
base	37.0	27.7	57.5	122.2	21.5
base+MAD	37.6	28.2	58.1	127.9	22.0
base+SAP	38.0	28.5	58.3	128.3	22.0
MAD+SAP	**38.6**	**28.7**	**58.5**	**128.8**	**22.2**

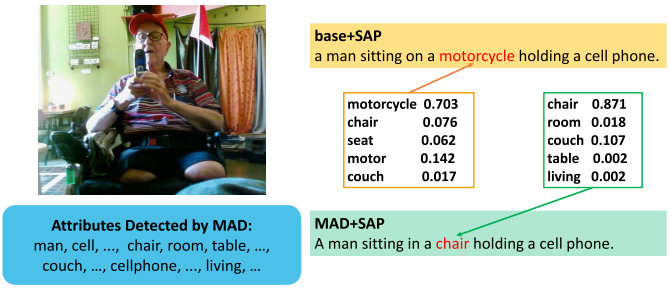


Fig. 9. Example of the predicted subsequent attributes and corresponding attention weights for an input image. K is set to 5 for simplicity.

TABLE VI

THE PERFORMANCE OF SELECTING DIFFERENT NUMBERS OF ATTRIBUTES. THE STANDARD DEVIATION OF EACH COLUMN IS SHOWN IN THE BOTTOM

K	Bleu-4	Meteor	Rouge-L	CIDEr-D	SPICE
base+MAD					
5	37.2	28.1	58.1	127.2	22.0
10	37.5	28.3	58.1	127.6	22.0
15	37.6	28.2	58.1	127.9	22.0
20	37.4	28.3	58.2	127.3	22.0
std	0.171	0.096	0.050	0.316	0
MAD+SAP					
5	38.4	28.6	58.4	128.1	22.2
10	38.6	28.7	58.5	128.8	22.2
15	38.5	28.6	58.5	128.7	22.2
20	38.4	28.6	58.6	128.5	22.2
std	0.096	0.050	0.082	0.310	0

information into the model. We observe that quality of the subsequent attribute prediction can be obviously improved by using both MAD and SAP simultaneously, which further hovers the performance of the image captioning.

As is shown in Fig 9, *'motorcycle'* is predicted as one of the subsequent attributes when only SAP is used. Although the word *'motorcycle'* complies well with the current linguistic context when the previous attribute is *'sitting'*, it is actually not related to the input image. However, by using MAD, more accurate attribute priors for the image can be generated, leading to more reasonable subsequent attribute predictions. As such, our proposed MAD+SAP model predicts appropriate subsequent attributes and assigns them with reasonable attention weights so that more precise captions can be generated.

3) Comparison Between Image Attributes and Subsequent Attribute: The number of selected attributes, denoted as K, may affect the performance of the image attribute based captioning models in the generation of the semantic context. Selecting too many or too few image attributes may downgrade the model's performance by either introducing redundancies or causing lack of semantic information. Table VI shows the influence of the K when adopting conventional image attributes predicted by the aforementioned **base**+**MAD** model and the subsequent attributes predicted by the **MAD**+**SAP** model. Generally, the standard deviations of all the metrics are smaller when adopting the subsequent attributes, indicating the image captioning performance of the proposed MAD+SAP model is not sensitive to the number of selected attributes. Generally speaking, this is because more subsequent attributes appropriate for the current linguistic context are effectively selected by SAP. Therefore, changing the value of K does not greatly affect the semantic information of the semantic context, leading to robustness of the caption generation process w.r.t. the variation of K.

TABLE VII

MODEL ENSEMBLE PERFORMANCE OF BLEU-4(B@4), METEOR(M), ROUGE-L(R), CIDER-D(C) AND SPICE(S) ON THE COCO TEST SPLIT. THE LARGEST NUMBER IN EACH COLUMN IS DISPLAYED IN BOLD

Methods	B@4	M	R	C	S
GCN-LSTM$_{fuse}$ [31]	38.3	28.6	58.5	128.7	22.1
SGAE$_{fuse}$ [29]	**39.0**	28.4	**58.9**	129.1	22.2
MAD+SAP$_{fuse}$	**39.0**	**28.9**	58.8	**129.8**	**22.3**

4) Model Ensemble: Model ensemble has long been adopted in image captioning to boost the model's capacity. The best results reported by recent state-of-the-art works such as GCN-LSTM [32] and SGAE [30] are actually obtained by fusing different models, which are denoted as GCN-LSTM$_{fuse}$, SGAE$_{fuse}$ in Table VII. We therefore report the performance of the ensemble of two MAD+SAP models which is denoted as MAD+SAP$_{fuse}$. The performance of MAD+SAP$_{fuse}$ outperforms GCN-LSTM$_{fuse}$ [32] and SGAE$_{fuse}$ [30] in most metrics. Note that while the training batch size of our method is only 200, GCN-LSTM is trained with an extremely large batch size of 1000, which might be critical in its achieving high performance. Comparing with our end-to-end trained method, SGAE leverages extra scene graph features and requires three steps of training. This comparison shows that our proposed MAD+SAP model is also well coupled with the widely adopted model ensemble strategy.

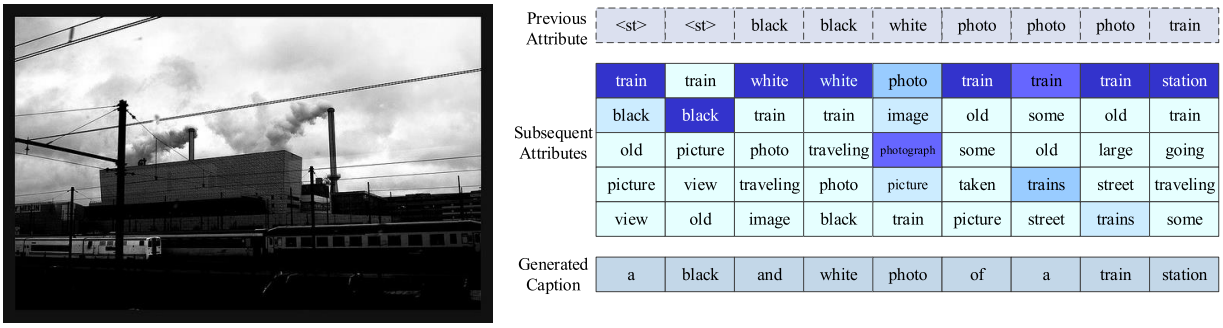
V. EXTENSION TO VIDEO CAPTIONING

Semantic attributes are also very widely adopted in the task of video captioning. For instance, Pan *et al.* [33] fused the image attributes and video attributes to enhance sentence generation. Yan *et al.* [35] proposed multi-faceted attention layer to jointly leverage the visual features and semantic attributes. Just like previous attribute based image captioning methods, these video captioning works leveraged pre-trained attribute detectors and used all detected attributes at each time step. Therefore, it is promising to further verify the effectiveness of our proposal in video captioning. We perform video captioning experiments on two benchmark datasets: MSVD dataset [39] and MSR-VTT dataset [40]. We leverage ResNet-152 to extract 40 2048d visual features and utilize C3D [41] pre-trained on the Kinetics dataset [42] to extract 2048d frame representations every 16 frames for each video

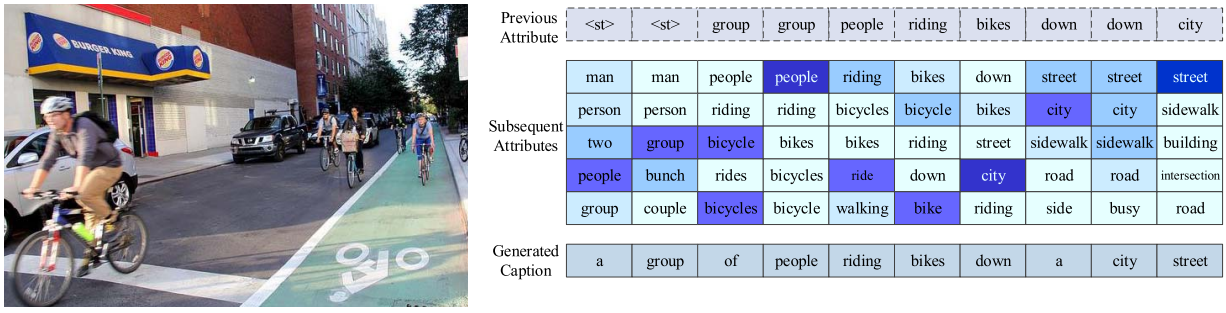
TABLE VIII

VIDEO CAPTIONING PERFORMANCE (%) ON THE MSVD DATASET AND MSR-VTT DATASET. THE INPUT FEATURES THAT EACH METHOD UTILIZES ARE SHOWN IN THE PARENTHESES, WHERE I-VN, R-N AND IR DENOTES THE N^{th} VERSION OF INCEPTION NETWORK, N-LAYER RESNET AND INCEPTION-RESNET-V2 RESPECTIVELY. THE LARGEST NUMBER IN EACH COLUMN IS MARKED IN BOLDFACE

Methods	MSVD dataset				MSR-VTT dataset			
Metric	Bleu-4	Meteor	Rouge-L	CIDEr-D	Bleu-4	Meteor	Rouge-L	CIDEr-D
LSTM-TSA (VGG + C3D) [32]	52.8	33.5	-	74.0	-	-	-	-
RecNet (I-V4) [33]	52.3	34.1	69.8	80.3	39.1	26.6	59.3	42.7
MFATT (R-152 + C3D + Faster R-CNN) [34]	52.0	33.5	-	72.1	39.1	26.7	-	-
STAT (I-V3/R-152 + C3D) [35]	52.0	33.5	-	73.8	39.3	27.1	-	43.9
GRU-EVE (IR + C3D + YOLO) [36]	47.9	35.0	71.5	78.1	38.3	**28.4**	60.7	48.1
EncDec+CG+POS (IR + I3D) [37]	52.5	34.1	71.3	88.7	**41.7**	27.8	61.2	**48.5**
baseline (R-152 + C3D) (ours)	51.5	35.0	71.5	88.0	40.2	27.8	59.9	46.0
MAD+SAP (R-152 + C3D) (ours)	**53.3**	**35.4**	**72.0**	**90.8**	41.3	28.3	**61.4**	**48.5**



(a)



(b)

Fig. 10. Example of the predicted subsequent attributes and the generated caption for two image in MSCOCO test set. The semantic attention weights of corresponding words are also shown in the table, higher color saturation denotes larger weights. K is set to 5 for simplicity. Our model generates plausible image captions for both gray scale image, as sub-figure (a), and colored image, as sub-figure (b).

in both datasets. We set the max length of sentence to 28 and keep the words that appear more than once to form the vocabularies of 7051 words and 16860 words for MSVD and MSR-VTT respectively. Similar to Long *et al.* [35], we firstly remove the meaningless words, such as *'a'*, *'is'*, and then select the most frequent 10 words across captions of each video as the ground truth attributes. We form the baseline model by implementing soft attention to the outputs of the visual feature encoder and the C3D features in the S2VT model [43]. The aforementioned MAD and SAP modules are incorporated into the baseline model to additionally include the semantic information to boost video captioning. Similar to that in image captioning, we utilize Adam [26] to optimize the loss function in Eq. 19 with the batch size of 100 for 450 epochs in MSVD dataset and 800 epochs in MSR-VTT dataset. The learning rate is initialized to $4e$-4 and decays by a factor of 0.8 every 30 epochs on MSVD dataset and every 50 epochs on MSR-VTT dataset.

We evaluate and compare our MAD+SAP method with six state-of-the-art RNN-based techniques on the official test sets, which contains 670 video in MSVD dataset and 2990 videos in MSR-VTT dataset respectively. Table VIII shows the Bleu-4, Meteor, Rouge-L and CIDEr-D metrics of our method and other compared methods. Our proposed MAD+SAP model achieves much better performance, especially in terms of CIDEr-D, over the early approaches such as LSTM-TSA [33], RecNet [34], MFATT [35] and STAT [36]. Compared with the recently proposed methods such as GRU-EVE [37] and EncDec+CG+POS [38], our model achieves better performance on the MSVD dataset and comparative performances on the MSR-VTT dataset. Note that these two compared methods adopt Inception-ResNet-V2 [44] to extract more fine-grained

Fig. 11. More image captioning results of different methods. C1 is the caption generated by the base model. C2 and C3 are the captions generated by base+MAD and MAD+SAP model respectively. C4 is one of the five ground truth sentences.

visual features while our model follows the conventional way like [35], [36] to use the ResNet-152 features. These experimental results on video captioning datasets further verify the effectiveness of our proposed MAD+SAP method.

VI. Qualitative Results

Fig. 10 illustrates the caption generation process of our model by demonstrating the subsequent attributes predicted at each time step. The previous attribute shown in the first row drives the SAP to select a concise subsequent attribute set at each time step. Intuitively, the subsequent attributes shown in Fig. 10 are generally appropriate for the linguistic context at the corresponding steps, leading to the generation of proper image captions. The weight of each attribute, assigned by the semantic attention module, is illustrated by the color saturation. The generated attribute words are usually assigned with high weights in the word generation process. A typical example in Fig. 10(a) shows the generation of the word *'station'* when the previous attribute is '*train*'. As an attribute word, '*station*' is semantically closely related to the image which is not only crowded with trains but also contains a station like building. Therefore, it is detected by the MAD and is assigned with a high prior probability. Then, since *'station'* usually follows the previous attribute *'train'* in the training set, it is selected by SAP with the highest probability. Finally, the semantic attention module also assigns it with a high weight, leading to its generation as the next word.

What is more, during caption generation, the SAP module is able to dynamically adjust the order of the selected subsequent attributes according to the generated partial caption. An example is shown in Fig. 10(b). Synonyms '*man*', '*person*' and '*people*' are all detected by MAD since they are closely related to the image content. At the very beginning, '*man*' and '*person*' are assigned with higher probabilities by SAP since there are a large number of training captions started with '*a man*' or '*a person*'. However, after the partial caption '*a group of*' is generated, the SAP automatically adjusts the order of the subsequent attributes and assigns '*people*' with the highest probability since the phrase '*a group of people*' is not only grammatically correct but also more commonly used. These two examples show that our model can predict reasonable subsequent attributes in proper order and assign them with appropriate weights to generate better captions. Also, the generated captions can be better explained according to the subsequent attributes generated at each step.

We show more image captioning results of different kind of models such as the **base** model, **base+MAD** model, and **MAD+SAP** model in Fig. 11. MAD provides accurate attribute prior to the captioning model to guide the network to generate correct attributes like '*blow drying*' and '*cart*' instead of wrong attributes like '*video game*' and '*cage*' as

is shown in the left two images in the top row of Fig. 11. SAP further strengthen the model with appropriate attributes at each time step in order to generate a more detailed caption. Our model correctly predicts the dogs' color by providing explicit semantic information at the corresponding time steps. Also, attending to different attribute set at different time steps leads our model to generate more correct attribute words like '*bathroom*' and '*holding*' and plausible attribute like '*computer*'. This demonstrate that our model generate precise and detailed captions by leveraging MAD and SAP.

VII. Conclusion

In this paper, we propose the use of subsequent attributes in image captioning. we demonstrate that selecting appropriate subsequent attributes to attend to is beneficial for image captioning models, especially in the CIDEr-D optimization process. We firstly propose a light weighted end-to-end trained multimodal attribute detector (MAD) to predict the probabilities of the image attributes. We then propose a subsequent attribute predictor (SAP) which leverages the probabilities as prior to select the appropriate subsequent attributes. Our proposed MAD+SAP model efficiently incorporates high-level semantic information to the image captioning model without additional pre-training of the attribute detector, leading to obvious boost in the performance of the image captioning model. We conduct extensive experiments to verify the benefits of MAD and SAP as well as the importance of end-to-end training of both modules. Experimental results show that our method achieves state-of-the-art performances in models trained by both cross-entropy loss and the RL-based loss. The visualization of subsequent attributes and attention weights further shows the explainability of our proposal. We believe that appropriate attribute information is beneficial for image captioning models. Besides, experiments on video captioning verify the applicability of our proposal in various tasks. For future work, we are going to model the relationship among image attributes in a more sophisticated way to further improve the visual captioning performance.

References

[1] 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.

[2] C. Gan, Z. Gan, X. He, J. Gao, and L. Deng, "StyleNet: Generating attractive visual captions with styles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3137–3146.

[3] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[4] P. Anderson *et al.*, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 3, no. 5, Jun. 2018, p. 6.

[5] 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.

[6] Z. Gan *et al.*, "Semantic compositional networks for visual captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jul. 2017, pp. 5630–5639.

[7] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1179–1195.

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

[9] 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. Linguist.*, 2002, pp. 311–318.

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

[11] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei, "Boosting image captioning with attributes," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 22–29.

[12] 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.

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

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

[15] 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.*, Jul. 2017, pp. 3242–3250.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[17] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.

[18] 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.

[19] C. Gan, T. Yang, and B. Gong, "Learning attributes equals multi-source domain generalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 87–97.

[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[21] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 5177–5186.

[22] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1–6.

[23] M. J. 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.

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

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

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980* . [Online]. Available: https://arxiv.org/abs/1412.6980

[27] 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.

[28] J. Gu, J. Cai, G. Wang, and T. Chen, "Stack-captioning: Coarse-to-fine learning for image captioning," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 6837–6844.

[29] Z.-J. Zha, D. Liu, H. Zhang, Y. Zhang, and F. Wu, "Context-aware visual policy network for fine-grained image captioning," *IEEE Trans. Pattern. Anal. Mach. Intell.*, to be published.

[30] X. Yang, K. Tang, H. Zhang, and J. Cai, "Auto-encoding scene graphs for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 10685–10694.

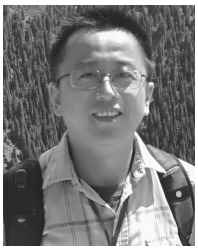
[31] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.

[32] T. Yao, Y. Pan, Y. Li, and T. Mei, "Exploring visual relationship for image captioning," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 684–699.

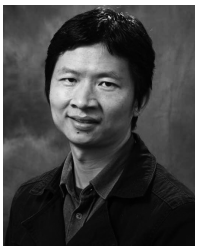
[33] Y. Pan, T. Yao, H. Li, and T. Mei, "Video captioning with transferred semantic attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6504–6512.

[34] B. Wang, L. Ma, W. Zhang, and W. Liu, "Reconstruction network for video captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7622–7631.
[35] X. Long, C. Gan, and G. De Melo, "Video captioning with multi-faceted attention," *Trans. Assoc. Comput. Linguistics*, vol. 6, pp. 173–184, Dec. 2018.
[36] C. Yan *et al.*, "STAT: Spatial-temporal attention mechanism for video captioning," *IEEE Trans. Multimedia*, vol. 22, no. 1, pp. 229–241, Jan. 2020.
[37] N. Aafaq, N. Akhtar, W. Liu, S. Z. Gilani, and A. Mian, "Spatio-temporal dynamics and semantic attribute enriched visual encoding for video captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 12487–12496.
[38] B. Wang, L. Ma, W. Zhang, W. Jiang, J. Wang, and W. Liu, "Controllable video captioning with pos sequence guidance based on gated fusion network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 2641–2650.
[39] D. L. Chen and W. B. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguist.*, 2011, pp. 190–200.
[40] J. Xu, T. Mei, T. Yao, and Y. Rui, "MSR-VTT: A large video description dataset for bridging video and language," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5288–5296.
[41] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6546–6555.
[42] W. Kay *et al.*, "The kinetics human action video dataset," 2017, *arXiv:1705.06950*. [Online]. Available: https://arxiv.org/abs/1705.06950
[43] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4534–4542.
[44] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.

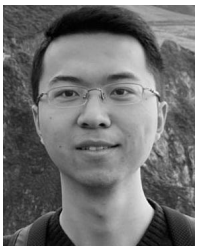
Yiqing Huang (Student Member, IEEE) received the bachelor's degree from Tsinghua University, Beijing, China, in 2017, where he is currently pursuing the Ph.D. degree. His research interests involve machine learning and image captioning.



Jiansheng Chen (Senior Member, IEEE) received the B.E. and M.E. degrees in computer science and technology from Tsinghua University, Beijing, China, in 2000 and 2002, respectively, and the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong in 2007. He is currently an Associate Professor with the Department of Electronic Engineering, Tsinghua University. He is also a member of the Beijing National Research Center for Information Science and Technology. His research interests include image processing, pattern recognition, and machine learning.



Wanli Ouyang (Senior Member, IEEE) received the Ph.D. degree from the Department of Electronic Engineering, The Chinese University of Hong Kong. Since 2017, he has been a Senior Lecturer with The University of Sydney. His research interests include image processing, computer vision, and pattern recognition.



Weitao Wan (Student Member, IEEE) received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 2016, where he is currently pursuing the Ph.D. degree with the Department of Electronic Engineering. His research interests include computer vision, adversarial learning, and weakly-supervised learning.

Youze Xue (Student Member, IEEE) received the bachelor's degree in electronic engineering from Tsinghua University, Beijing, China, in 2019, where he is currently pursuing the Ph.D. degree with the Department of Electronic Engineering. His current research interests include computer vision, 3D vision, and deep learning.