A Sparse Transformer-Based Approach for Image Captioning

ZHOU LEI, CONGCONG ZHOU, SHENBO CHEN, YIYONG HUANG, AND XIANRUI LIU

School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
Corresponding author: Zhou Lei (leiz@shu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61572306, in part by the National Key Research and Development Program of China under Grant 2017YFB0701600, in part by the Science and Technology Committee of Shanghai Municipality under Grant 19511121002, and in part by the Shanghai Engineering Research Center of Intelligent Computing System under Grant 19DZ2252600.

ABSTRACT
Image Captioning is the task of providing a natural language description for an image. It has caught significant amounts of attention from both computer vision and natural language processing communities. Most image captioning models adopt deep encoder-decoder architectures to achieve state-of-the-art performances. However, it is difficult to model knowledge on relationships between input image region pairs in the encoder. Furthermore, the word in the decoder hardly knows the correlation to specific image regions. In this article, a novel deep encoder-decoder model is proposed for image captioning which is developed on sparse Transformer framework. The encoder adopts a multi-level representation of image features based on self-attention to exploit low-level and high-level features, naturally the correlations between image region pairs are adequately modeled as self-attention operation can be seen as a way of encoding pairwise relationships. The decoder improves the concentration of multi-head self-attention on the global context by explicitly selecting the most relevant segments at each row of the attention matrix. It can help the model focus on the more contributing image regions and generate more accurate words in the context. Experiments demonstrate that our model outperforms previous methods and achieves higher performance on MSCOCO and Flickr30k datasets. Our code is available at https://github.com/2014gaokao/ImageCaptioning.

INDEX TERMS
Image captioning, self-attention, explicit sparse, local adaptive threshold.

I. INTRODUCTION
Nowadays, we encounter a large number of images from the Internet while most of them do not have a description. However, machine needs to automatic interpret image captions in the era of Artificial Intelligence. Image captioning has been an important research direction for many years. For example, it could help one who is visually impaired to understand images on the Internet. Also it has found applications in many areas such as content-based image retrieval, social media platforms and human-machine interaction. Image captioning is one of the most challenging tasks in computer vision that intends to automatically generate natural descriptions for images. It does not only require detecting the objects and their relationships in an image but also needs to generate a syntactically and semantically correct sentence [1].

The encoder-decoder architecture is the mainstream approach of image captioning followed by [2], [3]. In such a framework for image captioning, an image is first encoded into a set of feature vectors via a CNN based network [4] and then decoded to words via an RNN based network [5]. In addition, attention mechanism [6] plays an important role in such an encoder-decoder framework to generate encoded vectors based on image features and hidden states of LSTM at each time step which aims to align words to specific image regions [7]. In recent years, the great success of Transformer network [8] has attracted many researchers to explore towards the cutting edge techniques of self-attention for image captioning. The outstanding performance of self-attention comes from the ability to explore the relationships between the detected entities in encoder and model the correlation between image regions and hidden states as attention mechanism [9], [10].

However, the self-attention operation in original Transformer framework has an obvious drawback, as the original
Transformer assigns credits to all components of the context. It is unreasonable because many credits may be assigned to irrelevant information and should be ignored. For instance, the mainstream self-attention based mechanism estimates attention weights via multiplying the given query(hidden states) with key(encoded image features) from different modalities. The attention matrix is then applied to the value(encoded image features) to derive a weighted sum. Nevertheless, encoded image features may have little correlation to some irrelevant words(hidden states), which will lead to a very small fraction after multiplying the given query by key. So explicit sparse Transformer framework is needed to tackle the problem. More recently, local attention constraints was introduced in [11] and Zhao et al. [12] proposed to sparse attention with top-k strategy, however constraint local attention will break long term dependency and top-k strategy can not filter irrelevant information efficiently when the attention scores are very close.

In order to resolve these problems mentioned above, we propose a novel model named Local Adaptive Threshold to explicit sparse attention matrix for image captioning, which mitigates the drawback of previous methods by enhancing attention selection ability. Technically, we group attention matrix into multiple chunks and the element will be filtered if its value is smaller than the average distribution of the data in the according chunk. Local Adaptive Threshold will not hurt long term dependency and is determined by the distribution of neighbor nodes for each element. For further investigation, we integrate Local Adaptive Threshold into self-attention and acts as attention mechanism in the decoder which triggers the interaction between visual features and natural description. In a nutshell, Local Adaptive Threshold pays little attention to the least contributive states and can perform more concentrated attention than original Transformer framework. On the encoder side, the application of self-attention encourages our model to explore the relationships between the detected entities. Furthermore, we extend the detected entities with extra vectors to encode a persistent memory, these vectors are shared across all attention heads instead of relying on certain head and can be designed as trainable weights.

The contributions of this work are summarized as follows:

- Given that Transformer based attention mechanism may extract irrelevant information, we introduce a novel model named Local Adaptive Threshold to explicitly ignore the least contributive elements in attention matrix to reduce the extraction of irrelevant items and enhance the concentration of attention.
- We adopt a multi-level representations of image features to help encoder exploit low-level and high-level features, considering different feature levels may contribute different influences to the performance of model. Extra vectors are concatenated with feature vectors to encode a persistent memory.
- We conducted experiments on the MSCOCO [13] and Flickr30k [14] datasets. Our methods achieved a better performance of 129.2 CIDEr score on ”Karpathy” offline test split and 23.5 Meteor score on Flickr30k test split compared to previous methods.

The rest of the paper is organized as follows. In Section II, we review related works on modern image captioning methods including earlier popular approaches, Transformed based methods and LSTM improved methods. In Section III, we introduce our framework in detail. In particular, we discuss the modifications we make on the encoder and decoder. Section IV reports some experimental results that demonstrate the effectiveness and efficiency of our model. Section V concludes the paper.

II. RELATED WORK

A. IMAGE CAPTIONING

Popular approaches to image captioning adopt a deep encoder-decoder framework with attention mechanism. Typically, Anderson et al. [7] extract a set of salient image regions by a pooled convolutional feature vector using Faster R-CNN which enables the attention scores to be computed at the levels of salient image regions. Lu et al. [15] first generate a sentence template with blank slots which will be filled in by visual concepts using object detectors. Cornia et al. [16] propose a controllable approach to shift the rank of salient image regions by shift gate with adaptive attention. Li et al. [17] introduce a new architecture to facilitate vocabulary expansion and produce novel objects via pointing mechanism and object learners. Derived from human intuition, attention mechanism has achieved significant improvements for machine translation tasks. Attention mechanism generally calculates an importance score for each feature vector and normalizes to weights using a softmax function. Weights are applied to the feature vector to generate the attention result [6]. Chen et al. [18] propose a spatial and channel-wise attention mechanism that applies an attention mechanism in a channel-wise manner to be a process of selecting semantic attributes. Lu et al. [19] study an adaptive attention mechanism which is proposed to map feature vector to visual word or context word. Huang et al. [20] design an adaptive attention time mechanism to achieve arbitrary mapping between image region and word by adaptive steps across all time steps. More recently, more complex information such as attributes and relationships are integrated by GCN [21] to generate fine-grained level captions [22], [23] [24].

B. TRANSFORMER BASED METHODS

More researchers tend to utilize Transformer framework to improve captions due to the excellent performance of self-attention operation. Herdade et al. [25] design an architecture that incorporates information about the spatial relationships between detected objects through geometric self-attention. Li et al. [26] devise a unique attention mechanism to exploit the visual and semantic information simultaneously in the transformer framework. In addition, Huang et al. [9] introduce an extension of the attention operator in which the
final attended information is concatenated with the hidden state of LSTM and the attention context. For enhancing the connection between encoder and decoder, Cornia et al. [10] propose a meshed transformer to learn a hierarchical representation of the relationships between image regions and use a mesh-like connectivity at the decoding stage to exploit hierarchical image features. Multiple instance learning is also employed to build an attribute extractor in order to explore and distill cross-modal information where multi-head attention is adopted [27]. Pan et al. [28] introduce a X-Linear attention block which can measure both spatial and channel-wise attention distributions by bilinear pooling techniques. Those works both take image features and hidden states as input to generate attention scores in self-attention operation, however not all the image regions or hidden states play an equal importance on the model, filter irrelevant elements is able to make attention more concentrated.

C. LSTM IMPROVED METHODS

A Long Short-Term Memory performs as a sequence model to generate words according to input image features which is another important branch worth exploring. Ke et al. [29] introduce reflective position module and reflective attention module while the former selectively attends to the hidden states and the latter determines the attention distribution over the input image regions. Qin et al. [30] introduce a novel LSTM model consists of two parts where the one looks back attention value of previous time step and input into the current time step while the other predicts forward the next two words in one time step to improve captioning performance. Zheng et al. [31] start the description with a selected object and generate other parts of the sequence based on this object, the left-side sequence is generated according to visual features and the input object, while the right-side sequence is generated by left-side sequence. Deshpande et al. [32] propose to utilize part-of-speech tags to produce fast and accurate description, the captioning model take part-of-speech tags, image features and hidden states as input to generate a unified attention output as common methods do. The number of works aiming to improve LSTM is far less than other methods, however it is still worth exploring.

Our work draws on those related works and introduces a novel model named Local Adaptive Threshold which can sparse attention matrix efficiently and perform more concentrated attention than original Transformer framework. In next section, we introduce the modifications on self-attention module and describe the architecture of our image captioning model in detail.

III. OUR METHOD

A. LOCAL ADAPTIVE THRESHOLD ON SELF-ATTENTION

In a transformer framework, an attention function consists of a query and a set of key-value pairs where the query, key and value are all vectors. As shown in Figure 1(a), it first measures a similarity score between query and key and scales down the score by $\sqrt{d_k}$, where $d_k$ is normally the dimension of query and key. Then it applies a softmax function to obtain the weights on the values. The final output of the matrix can be computed as:

$$\alpha(q, k, v) = \text{softmax}(\frac{qk^T}{\sqrt{d_k}})v$$

where $q$, $k$ and $v$ represent query, key and value, respectively. Dot-product attention and multi-head attention are both proven to be efficient in practice in Transformer, where multi-head attention is concatenation of dot-product attention.

![FIGURE 1.](image1.png)

In Figure 1(b), we integrate our sparse module between the scale function and the softmax function to converge attention by selecting the items with higher numerical value and ignoring the elements with smaller numerical values, since softmax function is dominated by the largest elements. As shown in Figure 2, based on the hypothesis that the elements in matrix $P$ with higher values represent a closer relevance, we select the most contributive elements at each row of matrix $P$ to aggregate focus, where $P$ is the matrix of multiplication between query and key.

Specifically, we split attention matrix into $n$ chunks and compute the mean value of each chunk. The next step is to set

![FIGURE 2.](image2.png)
the elements smaller than \( t \) percent of the average to negative infinity:

\[
P_t = \begin{cases} 
  P_{ij}, & \text{if } P_{ij} \geq \frac{\sum_{i,j \in w} P_{ij}}{w} \times t \\
  -\infty, & \text{if } P_{ij} < \frac{\sum_{i,j \in w} P_{ij}}{w} \times t 
\end{cases}
\]

where \( w \) is the window size and \( t \) represents the threshold value. The rationale is to maintain a local window for each element in the attention matrix and set irrelevant information to negative infinity according to the data distribution of neighbor nodes. After the operation of softmax function:

\[
\alpha_{i,j} = \frac{e^{\alpha_{i,j}}}{\sum_j e^{\alpha_{i,j}}}
\]

those negative infinity elements will become zero and thus avoid the effect of negative noise. It should be noted that our proposed Local Adaptive Threshold draws on previous methods and is able to filter noisy information effectively.

**B. MULTI-LEVEL REPRESENTATIONS IN ENCODER**

Given an image, we first extract a set of feature vectors \( V = \{v_1, v_2, \ldots, v_k\} \) using a R-CNN based network, where \( k \) is the number of feature vectors in \( V \) and \( v_i \) is a vector that represents a salient object region. Instead of directly feeding these vectors to the decoder, we build a multi-level image feature network consisting of low-level and high-level features to refine their representations as illustrated in Figure 3.

**C. DECODER WITH SPARSE MULTI-HEAD SELF-ATTENTION**

A captioning model normally uses two LSTM layers to produce words where the first LSTM layer is used as an attention model and the second LSTM layer as a language model shown as Figure 4(a). Followed by [9], we apply multi-head self-attention to decoder. The input vector to the first LSTM layer at each time step consists of mean pooled image feature, the previous output of the language LSTM and words:

\[
x_t^1 = [h_{t-1}^2, \bar{v}, W_v \Pi_t] 
\]

where \( \bar{v} \) is a word embedding matrix for the vocabulary and \( \Pi_t \) is one-hot encoding of the input word at time step \( t \).

With the LSTM adopted, the hidden state \( h_t^1 \) and cell state \( c_t^1 \) are modeled as:

\[
h_t^1, c_t^1 = \text{LSTMCell}(x_t^1, h_{t-1}^1, c_{t-1}^1)
\]

where \( x_t \) is the input of the first LSTM layer. The iteration is performed in a for loop with an LSTMCell rather than executing automatically without a loop with an LSTM. Because we need to obtain the attention scores at each decode step, an LSTMCell is able to generate outputs in a single time step operation, whereas an LSTM would iterate over all time steps continuously and provide all the outputs at once. Following the idea of self-attention, suppose that

\[
\begin{align*}
\text{query} &= h_t^1, \\
\text{key} &= v_i, \\
\text{value} &= v_i,
\end{align*}
\]
where $i \in \{1, 2, \ldots, 6\}$, $v_i$ is the $i$-layer feature vector from the encoder. For multi-head attention, we divide $q$, $k$, $v$ into $8$ slices and generate the attention matrix $P$:

$$ P = q_i k_i^T \sqrt{d} $$

The final output representation of self-attention can be computed as:

$$ head_i = Av_i $$

where $A$ is the result of select operation on matrix $P$ shown as Figure 2. The concatenate operation is shown as follows:

$$ att = \text{concat}(head_i) $$

In order to make full use of each encoding layer, the final $att$ operation is defined as:

$$ att = \sum_{i=1}^{N} \alpha_i \times att_i $$

where $\alpha_i$ is a sigmoid gate representing the weight of $att_i$ and $att_i$ is the sparse self-attention operation of the $i$-th encoder layer. We make full use of each encoding layer and fuse the layers together by weighted sum in case of attending a single visual input from the encoder. The intention is that we exploit multi-level representations of image features and each level may have different contribution to the context of the decoder. The input to the language model consists of the extension attention operation, concatenated with the output of the first LSTM layer, given by:

$$ x_t^2 = \left[ att, h_t^1 \right] $$

A gate linear unit is adopted afterwards instead of LSTM because a gate linear unit is more light-weight and can model context information as LSTM does:

$$ c_t = f(x_t^2) \times g(f(x_t^2)) $$

where $f$ is a linear function and $g$ represents a sigmoid function. The conditional probabilities of words in vocabulary at each time step can be computed as:

$$ p(y_t|y_{1:t-1}, V) = \text{softmax}(W_p c_t) $$

The distribution over complete output sequences is calculated as the product of conditional distributions:

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

### IV. EXPERIMENTS

#### A. DATASETS AND EVALUATION METRICS

We use MSCOCO Dataset [13] and Flickr30k [14] for our evaluation. MSCOCO Dataset is a very large image dataset for many computer vision tasks such as image recognition, object detection, instance segmentation and image captioning. There are more than $300,000$ images, $80$ object categories and $5$ captions per image in MSCOCO Dataset. The standard train split of MSCOCO contains $82783$ images, with $40504$ for validation and $40775$ for test. The “Karpathy” data split is used in our paper where $5000$ images are used for validation, $5000$ images for testing and the rest for training. The “Karpathy” data split does not adopt standard test split. Instead it puts train split and validation split together and separate $5000$ images each for validation and test. We use the “Karpathy” split for offline evaluation and standard split for online evaluation. Flickr30k is another dataset for automatic image captioning which contains $31014$ images and $158k$ captions provided by human annotators. It does not provide a fixed split of images for train, validation and test. Researchers are able to choose their own options for train, validation and test split.

The MSCOCO caption evaluation tool including BLEU [35], METEOR [36], CIDEr [37] and ROUGE-L [38] metrics are used to evaluate our method and compare with other methods. BLEU is used to measure the quality of machine-generated text while individual parts of text are compared with reference text. However, BLEU heavily depends on the length of generated text and achieves good scores only if the generated text is short. Nonetheless, BLEU is a popular metric because of its pioneer in automatic evaluation of machine translation. In addition to comparing word segments to reference text, METEOR takes synonyms of words and stems of a sentence into consideration to make a better calibration. CIDEr is another important evaluation metric that adopts term frequency-inverse document frequency. It does not only calculate the frequency and proportion of specific words in a document, but also the proportion of documents

| Dataset          | Train | Val  | Test |
|------------------|-------|------|------|
| Karpathy split [6] | 113287 | 5000 | 5000 |
| Standard split [34] | 82783 | 40504 | 40775 |
| Flickr30k [14]     | 29000 | 1014 | 1000 |

### TABLE 1. Different split of MSCOCO and Flickr30k dataset.
with specific words in all documents. ROUGE is a set of metrics that measure the quality of text summary and are designed to compare word sequences, word pairs and n-grams with a set of reference summaries.

**B. TRAINING DETAILS**

Our code is implemented using PyTorch. For preprocessing coco captions, we map all words that occur less than 4 times to a special UNK token and create a vocabulary for all the remaining words. A pre-trained Faster-RCNN [39] model on ImageNet [40] and Visual Genome [41] is employed to extract bottom-up [7] feature vectors of images. For the MSCOCO dataset, the dimension of each feature vector is \( v \times d \) where \( k \) is a variable and \( d = 2048 \). Moreover we project it to a new dimension of \( d = 1024 \) which is also the hidden size of the LSTM in the decoder. As for the training process, the objective is to maximize the sum of the log likelihood of each generated word:

\[
\theta = \arg \max \sum_{t=1}^{T} \log p(w_t | V, w_{0}, \ldots w_{t-1}; \theta) \tag{18}
\]

where \( \theta \) represents the parameters to be learned and \( w_t \) represents generated words at time step \( t \). We train our model under cross entropy loss for 25 epochs with a mini batch size of 10 using an Adam [42] optimizer.

\[
L_{XE}(\theta) = - \sum_{i=1}^{T} \log p(y_{t} | y_{1:T-1}) \tag{19}
\]

We increase the scheduled sampling probability [43] by 0.05 every 5 epochs and optimize the CIDEr score with SCST [34] for another 15 epochs to directly optimize the non-differentiable metrics. The language model can be regarded as the agent to interact with the environment of images and words. The action is the process of predicting next word. The agent updates its state at the end of each action and gets the reward after meeting the end-of-sentence token where the state represents the hidden state and the reward indicates CIDEr score. The following formulation comes from SCST [34]:

\[
L_{RL}(\theta) = -E_{y_{1:T}} \log p_{\theta}(r(y_{1:T})) \tag{20}
\]

where the reward \( r(\cdot) \) uses the score of CIDEr. The gradients can be approximated:

\[
\nabla_{\theta} L_{RL}(\theta) \approx -(r(y_{1:T}^*) - r(\hat{y}_{1:T})) \nabla_{\theta} \log p_{\theta}(y_{1:T}^*) \tag{21}
\]

where \( y^* \) is a result sampled from a probability distribution, while \( \hat{y} \) indicates a result of greedy decoding.

**C. BASELINE METHODS**

To evaluate the effectiveness of our model in this study, we compare with several strong competitors in terms of four evaluation metrics on the stage of reinforcement training process as shown in Figure 5 and Figure 6. For fair comparison, all the competitive models are trained under cross entropy loss for 25 epochs and another 15 epochs for reinforcement training. A brief introduction of those baseline methods is as follows:

**FC model:** This model was proposed in SCST [34]. The FC model first encodes the input image using a deep CNN and embeds it by a linear projection. Words are fed back into the LSTM and are represented using one hot vector that is embedded with a linear embedding, which has the same output dimension as the linear projection in the feature extractor. This method is quite simple compared to other encoder-decoder frameworks.

**Show, Attend and Tell model:** Rather than using a fixed representation of the image, this attention model [6] dynamically focuses on a specific region of the image at each time step to boost the performance in the deep encoder-decoder framework.

**Bottom-Up and Top-Down model:** This up-down model consists of bottom-up and top-down attention mechanism, which enables the attention scores to be computed at the levels of salient image regions [7]. The bottom-up attention mechanism determines the image regions to be focused, while the top-down attention mechanism decides the weights of different image regions.

**Transformer model:** This model [8] consists of components including multi-head self-attention, positional encoding and feed forward. In view of the excellent performance on numbers of natural language processing tasks, we compare our modified model to this implementation.

**AAT model:** This model is proposed in AAT [20] and implements an arbitrary mapping between image regions and words across all attention steps. Traditional attention 

**TABLE 2.** Online evaluation performance on the MSCOCO test server where \( c5 \) and \( c40 \) represents that each image has 5 or 40 reference captions.

| Model          | B-1 | B-2 | B-3 | B-4 | M   | RL  | CIDEr |
|---------------|-----|-----|-----|-----|-----|-----|-------|
| NIC [2]       | 71.3| 89.5| 54.2| 80.2| 40.7| 69.4| 30.9  |
| SCST [34]     | 78.1| 93.7| 61.9| 86.0| 47.0| 75.9| 35.2  |
| LSTM-A [44]   | 78.7| 93.7| 62.7| 86.7| 47.6| 76.5| 35.6  |
| Up-Down [7]   | 80.2| 95.2| 64.1| 88.8| 49.1| 79.4| 36.9  |
| RNet [45]     | 80.4| 95.0| 64.9| 89.3| 50.1| 80.1| 38.0  |
| GCN-LSTM [23] | -   | -   | 65.5| 89.8| 50.8| 80.3| 38.7  |
| SGA [22]      | 81.0| 95.3| 65.6| 89.5| 50.7| 80.4| 38.5  |
| Ours          | 80.4| 94.7| 65.2| 88.7| 50.4| 80.2| 38.4  |
operations only compute weights in a single step while AAT model uses a threshold value to compute weights across a single step.

AoANet model: This model is proposed in AoANet [9] where the results of self-attention and initial query are concatenated together and fed into two linear layers to obtain information vectors by multiplication with a sigmoid gate. The final result is used as an alternative to the original self-attention operation.

The comparison suggests that our method performs better than the baseline methods on the reinforcement training process. In particular, our method outperforms AoANet, which is considered to be one of the best models in recent years.

D. QUANTITATIVE RESULTS

In Table 3, we compare our methods with several state-of-the-art methods on MSCOCO dataset. The NIC [2] method was the first work proposed to apply deep encoder-decoder framework in image captioning. The SCST [34] method introduced the notion of reinforcement learning based training which is the key idea of improving CIDEr score. The Up-Down [7] method proposed a combined bottom-up and top-down attention mechanism that enables attention to be calculated at the level of salient image regions. The AoANet [9] method introduced an extension of the attention operator in which the final attended information is weighted by a gate. Our work is developed on top of these methods. The results in Table 3 show that our method is 15.2% higher than SCST [34] in terms of CIDEr score. Compared with Up-Down [7], our model is 9.1% higher. AoANet has been considered to be one of the most competitive models in recent years, but its Bleu score is still lower than ours. These results indicate that our model is better than these methods in some of the evaluation metrics.

We further evaluate our method on the Flickr30k dataset which is another popular dataset for early image captioning tasks. Recent research rarely adopts it because the scale of Flickr30k dataset is much smaller than the MSCOCO dataset. As shown in table 4, our model slightly outperforms other methods in terms of Meteor score, but most metrics are lower than that of VS-LSTM [51], the reason may be that our model focuses less on semantic information than VS-LSTM.

Moreover we evaluate our model on the official website of the MSCOCO team as shown in table 2, the evaluation is performed on the 40775 test split and 40504 validation
FIGURE 7. Examples of captions generated by our model on MSCOCO dataset, the caption is able to contain more nouns and verbs than AoANet.

FIGURE 8. Examples of captions generated by our model on Flickr30k dataset, the caption is able to contain more objects and adjective words than AoANet.

TABLE 4. The performances of various methods on Flickr30k test split.

| Model              | Bleu1 | Bleu2 | Bleu3 | Bleu4 | Meteor |
|--------------------|-------|-------|-------|-------|--------|
| NeuralTalk [3]     | 57.3  | 36.9  | 24.0  | 15.7  | -      |
| MindsEye [48]      | -     | 12.0  | 15.2  | -     | -      |
| NIC [2]            | 66.3  | 42.3  | 27.7  | 18.3  | -      |
| LRCN [49]          | 58.8  | 39.1  | 25.1  | 16.5  | -      |
| m-RNN-AlexNet [50] | 54.0  | 36.0  | 23.0  | 15.0  | -      |
| m-RNN-VggNet [50]  | 60.0  | 41.0  | 28.0  | 19.0  | -      |
| Hard-Attention [6] | 66.9  | 43.9  | 29.6  | 19.9  | 18.46  |
| SCA-CNN [18]       | 66.2  | 46.8  | 32.5  | 22.3  | 19.5   |
| NBT [15]           | 69.0  | -     | -     | 27.1  | 21.7   |
| VS-LSTM [51]       | 75.5  | 57.1  | 42.9  | 31.7  | 22.9   |
| **Ours**           | 74.0  | 56.3  | 41.8  | 31.2  | 23.5   |

split. There are two evaluation settings where c5 represents that each image has five reference captions while c40 represents that each image has forty reference captions. Compared with the top-performing methods over the past few years, our model outperforms SGAE [22] slightly in some of the evaluation metrics such as Bleu-4 [35], Meteor [36], Rouge [38] and achieves a promising performance.

E. ABLATION STUDY

To quantify the impact of each design of our proposed module, we compare our method against a set of models without some settings. As shown in Table 5 and 6, we compare the effect of memory vectors in the encoder and the effect of different attention modules in the decoder. The “base” model does not have a memory vectors module in its encoder and adopts original Transformer as attention mechanism in the decoder. We can infer that sparse attention is able to improve scores and our proposed Local Adaptive Threshold performs better than top-k strategy. On the encoder side, the results also verify the efficiency of memory vectors. Obviously, the experiments conducted on the two datasets demonstrate the superiority of our proposed Local Adaptive Threshold in the decoder. Furthermore we discover that for MSCOCO dataset, the optimal value of \(w\) and \(t\) is 4 and 0.8, for Flickr30k dataset, the optimal value of \(w\) and \(t\) is 4 and 0.16 after extensive comparative experiments.

For more intuitive display, we show some examples of image captions generated by our model in Figure 7 and Figure 8. As shown in the first image and forth image in Figure 7, the captions include “a book shelf”, “tray” that AoANet model does not cover, indicating that our method can generate captions that contain more objects in an image. Moreover, we infer that our model can generate more verb than AoANet model, e.g. “blowing out” in the second image.
For the Flickr30k dataset, it is more likely for our method to generate adjective words like “pink”, “straw” and more concrete nouns, e.g. “children” instead of “people” compared to AoANet model. In a word, our method can generate more content related and grammatically correct sentences.

It is clear that the sentence generated by our method is able to visually describe the basic content of a picture. Compared to human-annotated ground truths, their sentences contain more detailed information including rich adjectives describing the objects and objects that does not appear at generated captions. Moreover, our generated sentences typically contain expressions like “a man” or “a woman”, while human-annotated ground truths typically contain “a young girl” which is more vivid. We can observe that pre-trained image features are underutilized when generating sentences and captions can not produce rich information as human-annotated ground truths do. Future research could overcome those drawbacks.

V. CONCLUSION
In this article, we introduce a novel model called Local Adaptive Threshold which can explicit sparse Transformer effectively. Local Adaptive Threshold will not hurt long term dependency and is determined by the distribution of neighbor nodes. It is able to make the attention in original Transformer more concentrated on the most contributive components. We integrate Local Adaptive Threshold into self-attention and acts as attention mechanism in the decoder which help the model generate more accurate word. Moreover, comprehensive comparisons with state-of-the-art methods and adequate ablation studies demonstrate the effectiveness of our strategies. In the future, we will develop novel frameworks on image captioning to further advance our research such as unsupervised learning. The development of advanced image captioning method is vital for deep image understanding.

ACKNOWLEDGMENT
The authors would like to thank the High Performance Computing Center of Shanghai University for providing the computing resources and technical support.

REFERENCES
[1] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, “A comprehensive survey of deep learning for image captioning," ACM Comput. Surv., vol. 51, no. 6, pp. 1–36, Feb. 2019.
[2] 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.
[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] 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.
[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.
[6] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in Proc. Int. Conf. Mach. Learn., 2015, pp. 2048–2057.
[7] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, “Bottom-up and top-down attention for image captioning and visual question answering," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 6077–6086.
[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, arXiv:1706.03762. [Online]. Available: http://arxiv.org/abs/1706.03762.
[9] L. Huang, W. Wang, J. Chen, and X.-Y. Wei, “Attention on attention for image captioning," in Proc. IEEE Int. Conf. Comput. Vis., Oct. 2019, pp. 4634–4643.
[10] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara, “Meshed-memory transformer for image captioning," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2020, pp. 10578–10587.
[11] R. Child, S. Gray, A. Radford, and I. Sutskever, “Generating long sequences with sparse transformers," 2019, arXiv:1904.10509. [Online]. Available: http://arxiv.org/abs/1904.10509.
[12] G. Zhao, J. Lin, Z. Zhang, X. Ren, Q. Su, and X. Sun, “Explicit sparse transformer: Concentrated attention through explicit selection," 2019, arXiv:1912.11637. [Online]. Available: http://arxiv.org/abs/1912.11637.
[13] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. Lawrence Zitnick, and P. Dollár, “Microsoft COCO: Common objects in context," 2014, arXiv:1405.0312. [Online]. Available: http://arxiv.org/abs/1405.0312.
[14] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, “Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2015, pp. 2641–2649.
[15] J. Lu, J. Yang, D. Batra, and D. Parikh, “Neural baby talk," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7219–7228.
[16] M. Cornia, L. Baraldi, and R. Cucchiara, "Show, control and tell: A framework for generating controllable and grounded captions," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2019, pp. 8307–8316.
[17] Y. Li, T. Yao, Y. Pan, H. Chao, and T. Mei, “Pointing novel objects in image captioning," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2019, pp. 12497–12506.
[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] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in Proc. Eur. Conf. Comput. Vis. Pattern Recognit., Nov. 2018, pp. 684–699.

TABLE 5. Settings and results of ablation studies on MSCOCO dataset.

| Model     | Bleu1 | Bleu2 | Bleu3 | Bleu4 | Meteor | Rouge-L | CIDEr |
|-----------|-------|-------|-------|-------|--------|---------|-------|
| Base      | 80.2  | 38.3  | 28.5  | 38.5  | 127.7  |
| + Enc: Memory | 80.4  | 38.7  | 28.9  | 38.7  | 128.3  |
| + Dec: MH-Att | 80.2  | 38.4  | 28.8  | 38.7  | 128.2  |
| + Dec: Top-k | 80.4  | 38.7  | 28.8  | 38.7  | 128.4  |
| + Dec: LAT | 80.6  | 39.1  | 28.8  | 38.8  | 128.7  |
| Full      | 80.7  | 39.2  | 29.0  | 39.0  | 129.2  |

TABLE 6. Settings and results of ablation studies on Flickr30k dataset.

| Model     | Bleu1 | Bleu2 | Bleu3 | Bleu4 | Meteor |
|-----------|-------|-------|-------|-------|--------|
| Base      | 73.0  | 54.9  | 40.4  | 29.5  | 23.0   |
| + Enc: Memory | 73.2  | 55.4  | 40.8  | 30.0  | 23.2   |
| + Dec: MH-Att | 73.3  | 55.5  | 40.1  | 30.2  | 23.3   |
| + Dec: Top-k | 73.4  | 55.6  | 41.2  | 30.4  | 23.4   |
| + Dec: LAT | 73.6  | 56.3  | 41.8  | 31.1  | 23.4   |
| Full      | 74.0  | 56.3  | 41.8  | 31.2  | 23.5   |
[24] S. Chen, Q. Jin, P. Wang, and Q. Wu, “Say as you wish: Fine-grained control of image caption generation with abstract scene graphs,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2020, pp. 9962–9971.

[25] S. Herdade, A. Kappeler, K. Boakye, and J. Soares, “Image captioning: Transforming objects into words,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 11135–11145.

[26] G. Li, L. Zhu, P. Liu, and Y. Yang, “Entangled transformer for image captioning,” in Proc. IEEE Int. Conf. Comput. Vis., Oct. 2019, pp. 8928–8937.

[27] F. Liu, X. Ren, Y. Liu, K. Lei, and X. Sun, “Exploring and distilling cross-modal information for image captioning,” in Proc. 28th Int. Joint Conf. Artif. Intell., Aug. 2019, pp. 10–16.

[28] Y. Pan, T. Yao, Y. Li, and T. Mei, “X-linear attention networks for image captioning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2020, pp. 10971–10980.

[29] L. Ke, W. Pei, R. Li, X. Shen, and Y.-W. Tai, “Reflective decoding network for image captioning,” in Proc. IEEE Int. Conf. Comput. Vis., Oct. 2019, pp. 8888–8897.

[30] Y. Qin, J. Du, Y. Zhang, and H. Lu, “Look back and predict forward in image captioning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2019, pp. 8367–8375.

[31] Y. Zheng, Y. Li, and S. Wang, “Intention oriented image captions with guiding objects,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2019, pp. 8395–8404.

[32] A. Deshpande, J. Anjea, L. Wang, A. G. Schwing, and D. Forsyth, “Fast, diverse and accurate image captioning guided by part-of-speech,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2019, pp. 10695–10704.

[33] S. Sukhbaatar, E. Grave, G. Lample, H. Jegou, and A. Joulin, “Augmenting self-attention with persistent memory,” 2019, arXiv:1907.01470. [Online]. Available: http://arxiv.org/abs/1907.01470.

[34] 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. 7008–7024.

[35] T. W. K. Papineni, S. Roukos, 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.

[36] M. D. A. Lavie, “Meteor universal: Language specific translation evaluation for any target language,” in Proc. Assoc. Comput. Linguistics, 2014, pp. 376–380.

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

[38] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” in Proc. Workshop Text Summarization Branches Out, Barcelona, Spain, vol. 8, 2004, pp. 74–81.

[39] 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.

[40] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.

[41] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei, “Visual genome: Connecting language and vision using crowdsourced dense image annotations,” Int. J. Comput. Vis., vol. 123, no. 1, pp. 32–73, May 2017.

[42] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. Int. Conf. Learn. Represent., 2015, pp. 1–15.

[43] 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.

[44] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei, “Boosting image captioning with attributes,” 2016, arXiv:1611.01646. [Online]. Available: http://arxiv.org/abs/1611.01646.

[45] W. Jiang, L. Ma, Y.-G. Jiang, W. Liu, and T. Zhang, “Recurrent fusion network for image captioning,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 499–515.

[46] 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. 1–8.

[47] D. Liu, Z.-J. Zha, H. Zhang, Y. Zhang, and F. Wu, “Context-aware visual policy network for sequence-level image captioning,” in Proc. ACM Multimedia Conf. Multimedia Conf., 2018, pp. 1416–1424.