RSTNet: Captioning with Adaptive Attention on Visual and Non-Visual Words

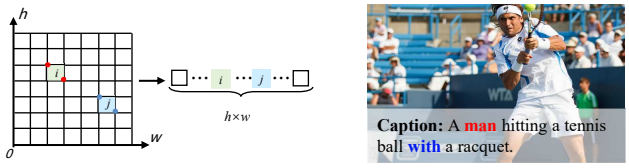
Xuying Zhang[1], Xiaoshuai Sun[12]*, Yunpeng Luo[1], Jiayi Ji[1], Yiyi Zhou[1],
Yongjian Wu[2], Feiyue Huang[2], Rongrong Ji[124]
[1]Media Analytics and Computing Lab, Department of Artificial Intelligence, School of Informatics,
Xiamen University, 361005, China. [2]Institute of Artificial Intelligence, Xiamen University.
[3]Youtu Lab, Tencent. [4]Peng Cheng Laboratory, Shenzhen, China.
zhangxuying@stu.xmu.edu.cn, xssun@xmu.edu.cn, {lyricpoem1997,jjyxmu}@gmail.com,
zhouyiyi@xmu.edu.cn, {littlekenwu,garyhuang}@tencent.com, rrji@xmu.edu.cn

Abstract

Recent progress on visual question answering has explored the merits of grid features for vision language tasks. Meanwhile, transformer-based models have shown remarkable performance in various sequence prediction problems. However, the spatial information loss of grid features caused by flattening operation, as well as the defect of the transformer model in distinguishing visual words and non visual words, are still left unexplored. In this paper, we first propose ***Grid-Augmented (GA)*** *module, in which relative geometry features between grids are incorporated to enhance visual representations. Then, we build a BERT-based language model to extract language context and propose* ***Adaptive-Attention (AA)*** *module on top of a transformer decoder to adaptively measure the contribution of visual and language cues before making decisions for word prediction. To prove the generality of our proposals, we apply the two modules to the vanilla transformer model to build our Relationship-Sensitive Transformer (RSTNet) for image captioning task. The proposed model is tested on the MSCOCO benchmark, where it achieves new state-of-art results on both the Karpathy test split and the online test server. Source code is available at GitHub* [1].

1. Introduction

Image captioning task aims to automatically generate a natural language sentence to describe the visual content of a given image. The encoder-decoder framework inspired by neural machine translation [34] has been widely adopted by captioning models[39, 40, 41, 5, 25], in which the CNN based encoder extracts visual features and the RNN based decoder generates the output sentence. Besides, the attention mechanism was introduced in order to help the model focus on the relevant positions when generating each word [41, 16]. Based on the encoder-decoder framework, most efforts to improve image captioning model focus on two main aspects: a) optimizing the visual features extracted from the input image [2, 42, 12], and b) improving the model structure for feature processing [41, 2, 23, 6, 28].



(a) (b)

Figure 1. This paper aims at reducing the spatial information loss of features and characterizing the visualizability of words in Captioning task. (a) shows the loss of spatial information when the grid features are flattened and fed to the transformer encoder. (b) illustrates the examples of visual (red) and non-visual (blue) word.

In terms of visual representation, region-based visual features [2] have become the dominant approach in major vision and language tasks like image captioning and visual question answering. However, the region extraction process is so time-consuming that currently most of the models with region features are directly trained and evaluated on cached visual features. Recently, Jiang *et al.* [15] revisited the grid features for VQA and demonstrated that grid features extracted from exactly the same layer of region feature detector [30] work quite well, both in speed and accuracy. In this paper, we also utilize grid features as the main visual representation for our captioning model. Nevertheless, grid features are flattened when fed to a transformer model, which inevitably leads to the loss of spatial information, as shown in Figure 1(a). Thus, we propose **Grid-Augmented (GA)** module which incorporates the spatial geometric relationships between relative locations into grids in order to facilitate a more comprehensive use of the grid features.

*corresponding author
[1]https://github.com/zhangxuying1004/RSTNet

In terms of model structure for feature processing, transformer [37] based captioning models [13, 6, 28] have been leading state-of-the-art performance on public benchmarks. The transformer architecture is able to better capture the relationship between visual features and process sequences in parallel during training. However, not all words in a caption are visual words and have corresponding visual signals due to the semantic gap between vision and language [23], as shown in Figure 1(b). For the attention module in transformer decoder layer, the intermediate representations used to predict each word are stacked together. As a result, all word predictions are treated equally, based on Scaled Dot-Product [37] operation, whether the word is a visual word or non-visual word. In other words, no effective measures have taken to process visual words and non-visual words differently for transformer based image captioning models. Thus, We build **Adaptive Attention (AA)** module based on language context and visual signals for transformer architecture to measure the contribution of visual signals and language context for a fine-grained caption generation.

We apply the **GA** module and **AA** module to our transformer based image captioning model, Relationship-Sensitive Transformer (RSTNet). For each attention module of transformer encoder, the relative geometry information of grid features is incorporated to calculate a more accurate attention distribution. For the decoder, there will be a trade-off between the contributions of visual and language cues rather than predicting words directly.

We extensively evaluate our RSTNet on the MSCOCO benchmark dataset [22], where quantitative and qualitative experiments prove the effectiveness of our model. In particular, our proposed RSTNet achieves state-of-the-art performance both offline and online. To gain more insights, we used the intermediate output of our RSTNet to measure the visualness of each word appeared in the Karpathy [17] test split of MSCOCO, which not only demonstrates the effectiveness of the proposed model but also reveals the impact of the semantic gap in a more intuitive way.

Our contributions can be summarized as follows:

- We propose a **Grid-Augmented (GA)** module, an extension to the flattened grid features, to boost the captioning performance by integrating the spatial information of raw visual features extracted from an image.
- We propose an **Adaptive-Attention (AA)** module, dynamically measuring the contribution of visual signals and language signal for the prediction of each word, to facilitate a more fine-grained captioning generation.
- We apply GA module and AA module into our RSTNet to achieve new state-of-art performance on COCO benchmark dataset. To grain more insights, We define a cross-domain attribute termed *visualness*, which quantitatively measures the visualizability of each word in vocabulary.

2. Related Work

2.1. Image Captioning

The main development of image captioning [41, 40, 5, 14, 24] can be divided into two stages: traditional method stage and deep learning method stage. In traditional method stage, retrieval-based [8, 27, 10] and template-based [19, 26, 36] methods are two common types of implementation for image captioning. Given an image, retrieval-based methods retrieve one or a set of most similar sentence from a pre-specified sentence pool, while template-based methods generate slotted sentence templates and use detected visual concepts to fill in the slots. With great progress made in deep learning, the encoder-decoder paradigm derived from neural machine translation was exploited in captioning models [25, 39] where CNN was used as the encoder to extract visual features from an image and RNN as the decoder to generate the corresponding output sequence. After that, the main focus of image captioning is to model the interaction between visual and lingual cues via attention mechanism to get more faithful and rich captions. For example, Xu *et al.* [41] introduced soft and hard attention into LSTM-based decoder, Lu *et al.* [23] proposed an adaptive attention mechanism to dynamically decide whether to attend visual signals when generating each word, Anderson *et al.* [2] proposed bottom-up and top-down attention mechanism that makes the visual features in attention upgrade from grid-level to object and salient region level.

2.2. Region Features vs. Grid Features

The representation of visual features have gone through two main stages after the extensive application of deep learning. In the first stage, a series of convolution neural network [32, 11] were proposed to represent visual information with grid features, and these grid features have achieved excellent performance on visual tasks like image classification [18, 32, 35, 11] and multi-modal tasks like image captioning. In the second stage, the emerging of R-CNN based detection models demonstrate the effectiveness of region features for fine-grained tasks. Typically, Anderson *et al.* [2] applied the pre-trained region feature to multi-modal task and achieved excellent performance in both image captioning and visual question answering. After that, region features have been extensively studied and become the de-facto standard for most vision and language tasks.

Recently, Jiang *et al.* [15] revisited the grid features for VQA and discovered that grid features extracted from exactly the same layer of a pre-trained detector can perform competitively against their region-based counterparts and meanwhile solve several critical issues like time-consuming, end-to-end training *etc.* These problems also exists in the state-of-art image captioning models. In this paper, we utilize grid features as visual representation. In

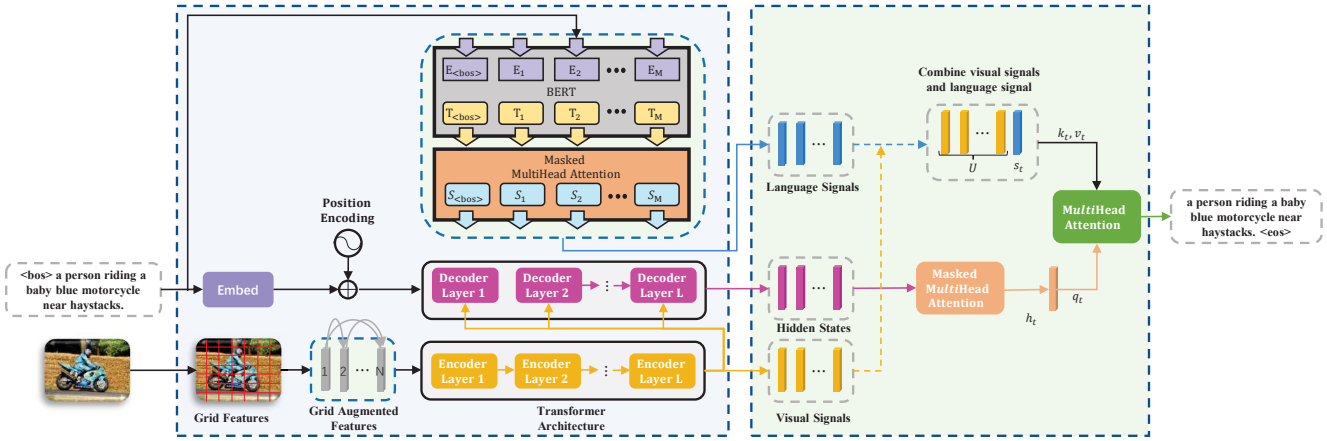


Figure 2. Overview of our RSTNet architecture for image captioning. Firstly, raw grid features are extracted according to [15], based on which we apply our grid-augmented module to enrich the grid features with spatial position and spatial relation. The language signal is encoded by a pre-trained BERT-based language model. Depending on the enhanced visual and language signals, we propose an adaptive attention module to perform multi-modal reasoning for word prediction. Our RSTNet is able to dynamically measure the contribution of visual and language signal to get more fine-grained image captions.

addition, we also explore an augmented form of grid features trying to solve the issue of spatial information loss caused by grid features flattening.

2.3. Transformer Models

RNN-based models are limited by their sequence nature and suffer from dependencies between distant positions [37]. In order to address this problem, [37, 7, 33] proposed to replace recurrence and convolutions with attention mechanisms and excitedly refreshed almost all the metrics of neural language processing (NLP). Subsequently, great efforts have been made to transfer this idea into image captioning. [3] explored the convolutional language model in image captioning model. Herdade *et al.* [12] incorporated geometry relationships between region features into transformer architecture for captioning. [13] proposed a GLU like structure on attention mechanism to determine the relevance between attention results and queries. Li *et al.* [20] extended the attention module linking transformer encoder and decoder to exploit visual information and semantic knowledge extracted by a external attribute detector. [6] introduced a learnable priori information to augment the attention module in transformer encoder and a mesh structure to build full connections between each encoder layer and each decoder layer. Pan *et al.* [28] introduced Bi-linear Pooling into transformer model to exploit both spatial and channel-wise bi-linear attention distributions.

Although the aforementioned transformer-based captioning models have achieved quite promising results, a serious problem still exists: all word sequences are coupled into high dimensional tensor, where visual and non-visual words are treated equally. In this paper, we explore an adaptive attention based on a transformer backbone so that the model can adaptively measure the contributions of visual signals and the current language context when predicting the word sequence for captioning.

3. Method

Figure 2 shows the overall architecture of our proposed RSTNet. The visual signals are represented by our grid-augmented features, the language signal is extracted by a pre-trained BERT-based language model, and our adaptive attention module measures the contribution of visual signals and the language context for word prediction.

We first show grid feature representation in Sec 3.1. Then, we give the details of language feature representation in Sec 3.2. Next, we introduce the implementation of the proposed Relationship-Sensitive Transformer (RSTNet) for image captioning task in Sec 3.3. Besides, we define *visualness* to describe the visualizability of words in Sec 3.4 and give the training details of the RSTNet in Sec 3.5.

3.1. Grid Feature Representation

We get the raw grid features following the operation in [15]. Given a set of $h \times w$ grid features, previous approaches usually flatten them and directly send them into a transformer encoder. However, this flattening operation will inevitably cause the loss of spatial information of the input image, *e.g.*, the position and relationship of grid pairs.

In this paper, we build a Grid-Augmented (GA) module which incorporates relative geometry relationships between grid positions to solve the above issue.

We first calculate a pair of 2D relative positions of each grid: $\left\{\left(x_i^{min}, y_i^{min}\right), \left(x_i^{max}, y_i^{max}\right)\right\}$, where $\left(x_i^{min}, y_i^{min}\right)$

is the relative position coordinate of the upper left corner of the grid i, and (x_i^{max}, y_i^{max}) is the relative position coordinate of the lower right corner of the grid i, as shown in Figure 1(a). The specific calculation process is shown in the supplementary material. We then calculate the relative center coordinate (cx_i, cy_i), relative width w_i, and relative height h_i of grid i as follows:

$$(cx_i, cx_i) = (\frac{x_i^{min} + x_i^{max}}{2}, \frac{y_i^{min} + y_i^{max}}{2}), \quad (1)$$

$$w_i = (x_i^{max} - x_i^{min}) + 1, \quad (2)$$

$$h_i = (y_i^{max} - y_i^{min}) + 1. \quad (3)$$

Finally, we imitate the computation of region geometry features in [12, 9] to obtain the relative geometry features between two grids i and j:

$$r_{ij} = \begin{pmatrix} \log(\frac{|cx_i - cx_j|}{w_i}) \\ \log(\frac{|cy_i - cy_j|}{h_i}) \\ \log(\frac{w_i}{w_j}) \\ \log(\frac{h_i}{h_j}) \end{pmatrix}, \quad (4)$$

$$G_{ij} = FC(r_{ij}), \quad (5)$$

$$\lambda_{ij}^g = ReLU(w_g^T G_{ij}), \quad (6)$$

where $r \in \mathbb{R}^{N \times N \times 4}$ is the relative geometry relationship between grids, FC is a fully-connected layer with activation function, $G \in \mathbb{R}^{N \times N \times d_g}$ is a high-dimensional representation of r, w_g is a weight parameter to be learned, $\lambda^g \in \mathbb{R}^{N \times N}$ is the relative geometry feature, and $N = h \times w$. The ReLU function acts as a zero trimming operation, which makes sure that we only consider the relations between grids with geometric relationships.

3.2. Language Feature Representation

In order to get the language features of given sequence, we once tried to imitate the method in [23], which utilizes a gated word memory as the language feature of current word. However, we found through experiments that memory information and hidden information are highly coupled for transformer decoder, resulting in a serious language bias.

Thus, we follow the recent trends in the community of Natural Language Processing and build a BERT-based [7] language model (BBLM) to extract language features. Considering only being able to access the partially generated sentence information at testing phase, we add a masked attention module similar to transformer decoder layer on top of the BERT model. Figure 3 depicts a schema of our language feature module. Given a word sequence $W = (< bos >, w_1, w_2, ..., w_M)$, this module will predict this sequence $\hat{W} = (\hat{w}_1, \hat{w}_2, ..., \hat{w}_M, < eos >)$ word by word with offset by one time step. This process flow can be expressed by the following formulas:

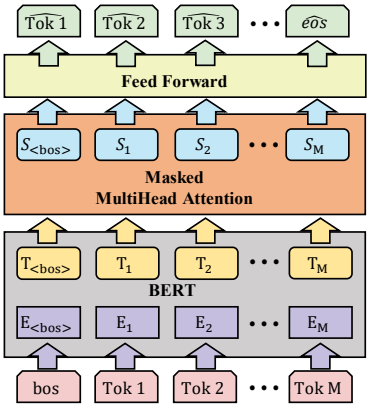


Figure 3. The architecture of our BERT-Based Language Model. The pre-trained BERT model is used to extract language features, and the Masked Multi-Head Attention prevents the word prediction of current step from the interference of the later step.

$$lf = BERT(W), \quad (7)$$

$$S = MaskedAttentionModule(FF1(lf) + pos), \quad (8)$$

$$\hat{W} = log\_softmax(FF2(S)), \quad (9)$$

where $lf \in \mathbb{R}^{M \times d_{bert}}$ is the output language feature of the BERT model, $pos \in \mathbb{R}^{M \times d_{model}}$ is the position encoding of word sequence, $S \in \mathbb{R}^{M \times d_{model}}$ is the output of the masked attention module, and $\hat{W} \in \mathbb{R}^{M \times d_{vocab}}$ is the log softmax distribution of predicted words.

We train this language model with cross-entropy loss. All parameters are frozen and the output of masked attention module S is used as the representation of language features in RSTNet. This operation is formulated as:

$$s_t \leftarrow BBLM(W_{<t}), s_t \in \mathbb{R}^{d_{model}}. \quad (10)$$

3.3. Relationship-Sensitive Transformer (RSTNet)

A typical transformer-based image captioning model follows the classic encoder-decoder framework, where the encoder takes the visual features extracted from the image as input and further processes them to strengthen their relationships. Given the encoded features from an encoder, the decoder then generates the output sequence word by word. The core component of transformer is Scaled Dot-Product Attention [37] whose input consists of matrix Q, K and V, where Q is the combination of n_q query vectors, K and V are the combining results of n_k key vectors and n_k value vectors, respectively.

Encoder The raw image feature is first flattened, and then embedded by a fully-connected layer followed by a ReLU and a dropout layer to project its dimension to $d_{model} =$

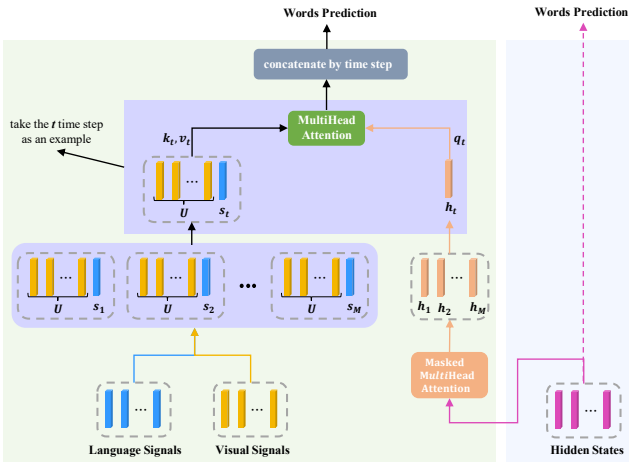


Figure 4. Illustration of our adaptive attention module. This module ensures that our model reconsiders the effect of language context before word prediction at each time step.

512. The embedded features are send to the first encoder layer of the transformer model. The Scaled Dot-Product Attention in the encoder layer is formulated as:

$$Q = UW_q, K = UW_k, V = UW_v, \tag{11}$$

$$Z = softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{12}$$

$$U \leftarrow U + Z. \tag{13}$$

where $U \in \mathbb{R}^{N \times d_{model}}$ is the packed visual feature vectors passing in transformer encoder layer, W_q, W_q, W_q are matrices of learnable weights, and d_k is a scaling factor.

Grid Augmented (AA) Module In order to compensate for the spatial information loss of the grid features caused by the flattening operation, we propose a grid-augmented Scaled Dot-Product Attention to enhance the encoder layer. In our proposal, incorporating the relative geometry feature introduced in Sec 3.1, we calculate a more accurate attention map. The grid-augmented Scaled Dot-Product Attention is formally define as follows:

$$Z_{aug} = softmax(\frac{QK^T}{\sqrt{d_k}} + \lambda^g))V, \tag{14}$$

$$U \leftarrow U + Z_{aug}, \tag{15}$$

where λ^g is the relative geometry feature of the grid features and Z_{aug} is the result of our augmented attention.

Decoder The word sequence features is first processed by word embedding and incorporated with word sequence position encoding before used as the input of the first decoder layer of the transformer model. The decoder of transformer can be formulated as:

$$h_t = Decoder(U, W_{<t}), \tag{16}$$

where $U \in \mathbb{R}^{N \times d_{model}}$ is the output of the last layer of transformer encoder, $W_{<t} = (w_0, w_2, ..., w_{t-1})^T, W_{<t} \in \mathbb{R}^{t \times d_{model}}$ is word sequence feature of the partially generated sentence, and h_t is the hidden state output by transformer decoder to predict the current word w_t.

The decoding process can be seen as a process continuously incorporating visual information under the guidance of word sequence features of the partially generated sentence to get the hidden state of the current word. However, the current word might be a non-visual word, as shown in Figure 1(b) in which case the language context should play a more important role than the visual signals for the word prediction. In the following, we proposed an Adaptive Attention which processes and attends to the visual and language signal simultaneously to solve the above issue.

Adaptive Attention (AA) Module We build our adaptive attention module on top of the classic transformer decoder. Instead of predicting word using the hidden state h_t directly, language signals introduced in Sec 3.2, visual signals output by encoder and the hidden states are combined together to measure the contribution of visual signals and language signal for each word prediction. Figure 4 depicts the the architecture of our Adaptive Attention module. The Adaptive Attention is formally define as follows:

$$q_{i,t} = h_t W_i^Q, k_{i,t} = [U; s_t] W_i^K, v_{i,t} = [U; s_t] W_i^V, \tag{17}$$

$$head_{i,t} = softmax(q_{i,t} k_{i,t}^T) v_{i,t}, \tag{18}$$

$$head_i = Concate(head_{i,1}, ..., head_{i,M}), \tag{19}$$

$$att = Concate(head_1, ..., head_h) W^O, \tag{20}$$

where $q_{i,t}$ is the query vector for the t-th word word in head i of multi-head attention, $k_{i,t}, v_{i,t}$ are the key matrix and value matrix for the t time step word in head i of multi-head attention respectively, $head_{i,t}$ is the attention result for the t-th word word in head i, $head_i$ is the attention result for the word sequence in head i, att is the attention result of multi-head attention for sequence generation.

3.4. Visualness

To gain more insights of our model, we define a cross-domain attribute called *visualness*, denoted as γ, based on our adaptive attention. The *visualness* of the t time step word γ_t is define as follows:

$$\alpha_{i,t} = softmax(q_{i,t} k_{i,t}^T), \alpha_{i,t} \in \mathbb{R}^{n+1}, \tag{21}$$

$$\beta_{i,t} = \alpha_{i,t}[-1], \beta_{i,t} \in \mathbb{R}, \tag{22}$$

$$\beta_t = average(\beta_{1,t}, ..., \beta_{h,t}), \beta_t \in \mathbb{R}, \tag{23}$$

$$\gamma_t = 1 - \beta_t, \tag{24}$$

where $\alpha_{i,t}$ is the softmax distribution of attention for the t-th word in head i, $\beta_{i,t}$ is language signal weight for the

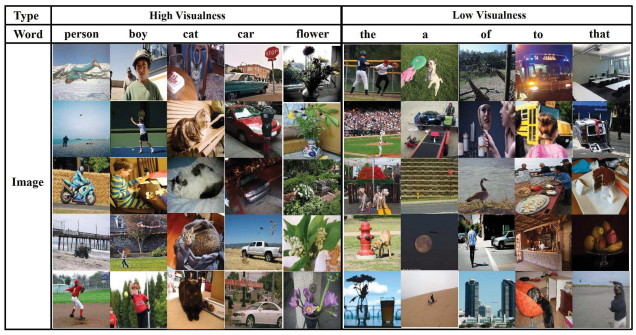


Figure 5. Visualization of the word visualness based on RSTNet. Words with high visualness can be clearly identified, while low visualness words show no direct correlation to the image contents.

t-th word in head i, β_t is average pooling of language signal weight over all head of multi-head attention for the t-th word. γ_t quantitatively measures the visualizability of the t-th word which can be aggregated across the entire dataset to generate the average γ score for each word in the vocabulary. We show and discuss the visualizations of typical words with high and low visualness, along with their related images, in Figure 5.

3.5. Training Details

Following a standard practice of image captioning [2, 31], we first optimize our model with the cross entropy loss:

$$L_{XE}(\theta) = -\sum_{t=1}^{T} \log(p_\theta(w_t^* | w_{1:t-1}^*)), \tag{25}$$

where θ is the parameters of our model, $w_{1:T}^*$ is the target ground truth sequence.

Then, we directly optimize the non-differentiable metric with Self-Critical Sequence Training [31]:

$$L_{RL}(\theta) = -E_{w_{1:T}\ p_\theta}[r(w_{1:T})], \tag{26}$$

where the reward $r(\cdot)$ is the CIDEr-D score.

Besides, we use the gradient expression in [6], where using the mean of rewards rather than greedy decoding to baseline the reward. The gradient expression for one sample is formulated as:

$$b = \frac{1}{k}(\sum_{i}^{k} r(w_i)), \tag{27}$$

$$\nabla_\theta L_{RL}(\theta) \approx -\frac{1}{k}\sum_{i=1}^{k}((r(w_{1:T}^i) - b)\nabla_\theta \log p_\theta(w_{1:T}^i)), \tag{28}$$

where k is the number of the sampled sequences, $w_{1:T}^i$ is the i-th sampled sequence, and b is the mean of the rewards obtained by the sampled sequences.

4. Experiments

4.1. Experimental setup

Dataset We evaluate our proposed model on the MS-COCO dataset [22], which is the most popular benchmark dataset for image captioning task. The MS-COCO dataset contains 123,287 images, which includes 82,783 training images, 40,504 validation images and 40,775 testing images, each of them annotated with 5 different captions. We adopt the splits provided by Karpathy et al.[17], where 5,000 images are used for validation, 5000 images for testing and the rest images for training. We remove punctuation from all sentences and convert them to lower case, and drop the words that occur less than 5 times, ending up with a vocabulary of 10201 words.

Evaluation Metrics Following the standard evaluation protocol, we use the full set of captioning metrics to evaluate the quality of image captioning, including BLEU [29], METEOR [4], ROUGR [21], CIDEr [38], and SPICE [1].

Implementation Details We follow the implementation of a Transformer-base model proposed by [6] to set hyperparameters and facilitate the training of our RSTNet. Specifically, an input image I is represented as a grid feature following the operation in [15], where the grid size is 7×7 and the dimension of image features is 2048. If not specifically specified, the d_{model} of the transformer is 512, the number of heads is 8, and the inner dimension of FFN module is 2048. The dropout probability we use is 0.1.

We adopt Adam optimizer to train our model. For cross entropy training, We use a epoch decay schedule for varying the learning rate to replace the learning rate policy in [6], the initial learning rate is 1, and the lambda learning rate is define as follows:

$$lambda\_lr = \begin{cases} base\_lr * e/4, & e \leq 3, \\ base\_lr, & 3 < e \leq 10, \\ base\_lr * 0.2, & 10 < e \leq 12, \\ base\_lr * 0.2 * 0.2, & otherwise, \end{cases} \tag{29}$$

where $base\_lr$ is set to 0.0001 and e is the current epoch number that starts from 0. For self-critical sequence training, the learning rate is set to a fixed value of 5×10^{-6}.

Our training is started with cross entropy optimization. If the cider value drops for 5 consecutive epochs, it will turn to self-critical sequence training. The training process stops when the cider value drops for 5 consecutive epochs in self-critical sequence training.

4.2. Ablative Analysis

To fully examine the impact of our proposed Grid-Augmented (GA) module and Adaptive Attention (AA) module, we conduct ablative study to compare different variants of RSTNet. We start from a base model which uses

Table 1. Ablation study on ResNext101 grid features

GA module	**AA** module	B@1	B@4	M	R	C	S
✗	✗	80.9	38.9	29.0	58.5	131.2	22.7
✗	✓	80.9	39.0	29.2	58.6	132.6	22.8
✓	✗	80.9	39.0	29.2	58.7	132.1	22.8
✓	✓	**81.1**	**39.3**	**29.4**	**58.8**	**133.3**	**23.0**

Table 2. Ablation study on ResNext152 grid features

GA module	**AA** module	B@1	B@4	M	R	C	S
✗	✗	81.2	39.4	29.4	59.0	133.2	23.1
✗	✓	81.0	39.2	29.6	58.9	134.3	23.3
✓	✗	81.6	39.6	29.6	59.2	134.2	23.2
✓	✓	**81.8**	**40.1**	**29.8**	**59.5**	**135.6**	**23.3**

base transformer model without any modification. Then, we incorporate **GA** module, **AA** module into the base model respectively. Finally, we incorporate both **GA** and **AA** module into the base model to build our RSTNet. In order to examine the generality, the above experiments are conducted on ResNext101 grid features and ResNext152 grid features and shown in Table 1 and Table 2, respectively.

In ResNext101 experiments, **GA** and **AA** schema already respectively bring an improvement with respect to the base transformer (from 131.2 to 132.1 and 131.2 to 132.6 respectively). In ResNext152 experiments, the improvement still exists (from 133.2 to 134.3 and 133.2 to 134.2 respectively), thus confirming that the two modules we proposed are beneficial. When we combine the two modules together, a bigger improvement appears (from 131.2 to 133.3 and 133.2 to 135.6 in ResNext101 and ResNext152 experiment respectively).

Table 3. Performance comparision with the state of the art on the COCO "Karpathy" test split.

Model \ Metrics	B@1	B@4	M	R	C	S
SCST [31]	-	34.2	26.7	55.7	114.0	-
Up-Down [2]	79.8	36.3	27.7	56.9	120.1	21.4
RFNet [16]	79.1	36.5	27.7	57.3	121.9	21.2
GCN-LSTM [43]	80.5	38.2	28.5	58.3	127.6	22.0
SGAE [42]	80.8	38.4	28.4	58.6	127.8	22.1
ORT [12]	80.5	38.6	28.7	58.4	128.3	22.6
AoANet [13]	80.2	38.9	29.2	58.8	129.8	22.4
M^2 Transformer [6]	80.8	39.1	29.2	58.6	131.2	22.6
X-Transformer [28]	80.9	39.7	29.5	59.1	132.8	**23.4**
RSTNet(ResNext101)	**81.1**	39.3	29.4	58.8	**133.3**	23.0
RSTNet(ResNext152)	**81.8**	**40.1**	**29.8**	**59.5**	**135.6**	23.3

4.3. Quantitative Analysis

Offline Evaluation We report the performance comparisons between our RSTNet and the state-of-art models on the offline COCO Karpathy test split in Table 3. The models we compare to include SCST [31], Up-Down [2], RFNet [16], GCN-LSTM [43], SGAE [42], ORT [12], AoANet [13], M^2 Transformer [6], and X-Transformer [28]. SCST uses attention over features and proposes self-critical training policy. Up-Down and RFNet boost the performance by using attention over spatial regions. GCN-LSTM and SGAE use Graph CNN and auto-encoding scene graphs respectively to exploit pairwise relationships between image regions for a rich semantic representation of the image. ORT introduces transformer architecture into image captioning and models the spatial relationship between region features. AoANet enhances conventional visual attention by further measuring the relevance between the attention result and query. M^2 Transformer builds a full-connected architecture between each encoder layer and each decoder layer. X-Transformer introduces Bilinear Pooling into the attention module of a base transformer. As it can be observed, our RSTNet significantly outperforms all the other methods in terms of most evaluation metrics.

Table 4. Comparing with SOTAs on ResNext101 grid features

Model \ Metrics	B@1	B@4	M	R	C	S
UP-Down [2]	75.0	37.3	28.1	57.9	123.8	21.6
Transformer [6]	80.9	38.9	29.0	58.5	131.2	22.7
AoA Transformer [13]	80.8	39.1	29.1	59.1	130.3	22.7
M^2 Transformer [6]	80.8	38.9	29.1	58.5	131.8	22.7
X-Transformer [28]	81.0	**39.7**	29.1	**59.0**	130.2	22.8
RSTNet(Ours)	**81.1**	39.3	**29.4**	58.8	**133.3**	**23.0**

Comparison with strong baselines In order to eliminating the interference of grid features, we also conduct experiments to compare our proposed RSTNet with the SOTAs on the same ResNext101 grid features in Table 4. Note that, the d_{model} parameter of Transformer architecture in X-Transformer is set to 768, while the one in other models is set to 512, as a result, we adjust the d_{model} parameter of X-Transformer to 512 for a fair comparison, and choose the 50 epoch results of self-critical training for displaying. The eventual experimental results demonstrate that our proposed RSTNet can achieve a superior performance when comparing with other SOTA methods under the same visual features and architecture configuration.

Online Evaluation We also report the performance of our proposed RSTNet model on the online COCO test server. The model we use is an ensemble of 4 RSTNet models trained on the Karpathy training split. Table 5 shows the performance of our RSTNet model comparing with the top-performing models of the leaderboard over official testing images with 5 reference captions (c5) and 40 reference captions (c40) respectively.

4.4. Qualitative Analysis

Figure 6 shows some examples of the captions generated by our RSTNet and the original Transformer given the same

Table 5. Leaderboard of the published state-of-the-art image captioning models on the COCO online testing server, where B@N, M, R and C are short for BLEU@N, METEOR, ROUGE-L and CIDEr scores. All values are reported as percentage.

Model \ Metrics	B@1		B@2		B@3		B@4		METEOR		ROUGE-L		CIDEr-D	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
SCST [31]	78.1	93.7	61.9	86.0	47.9	75.9	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.7
Up-Down [2]	80.2	95.2	64.1	88.8	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
RFNet [16]	80.4	95.0	64.9	89.3	50.1	80.1	38.0	69.2	28.2	37.2	58.2	37.1	122.9	125.1
GCN-LSTM [43]	80.8	95.9	65.5	89.3	50.8	80.3	38.7	69.7	28.5	37.6	58.5	73.4	125.3	126.5
SGAE [42]	81.0	95.3	65.6	89.5	50.7	80.4	38.5	69.7	28.2	37.2	58.6	73.6	123.8	126.5
ETA [20]	81.2	95.0	65.5	89.0	50.9	80.4	38.9	70.2	28.6	38.0	58.6	73.9	122.1	124.4
AoANet [13]	81.0	95.0	65.8	89.6	51.4	81.3	39.4	71.2	29.1	38.5	58.9	74.5	126.9	129.6
M^2 Transformer [6]	81.6	96.0	66.4	90.8	51.8	82.7	39.7	72.8	29.4	39.0	59.2	74.8	129.3	132.1
X-Transformer(ResNet-101) [28]	81.3	95.4	66.3	90.0	51.9	81.7	39.9	71.8	29.5	39.0	59.3	74.9	129.3	131.4
X-Transformer(SENet-154) [28]	81.9	95.7	66.9	90.5	52.4	82.5	40.3	72.4	29.6	39.2	59.5	75.0	131.1	133.5
RSTNet(ResNext101)	81.7	**96.2**	66.5	**90.9**	51.8	**82.7**	39.7	72.5	29.3	38.7	59.2	74.2	**130.1**	**132.4**
RSTNet(ResNext152)	**82.1**	**96.4**	**67.0**	**91.3**	52.2	**83.0**	40.0	**73.1**	**29.6**	39.1	**59.5**	74.6	**131.9**	**134.0**

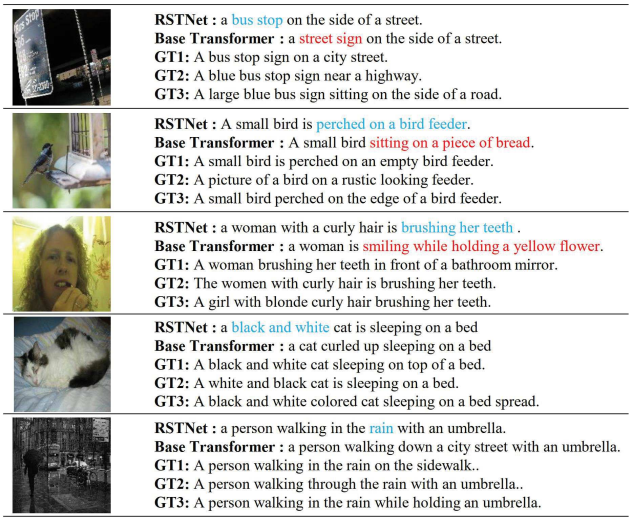


Figure 6. Examples of image captioning results by Base Transformer and our RSTNet, coupled with the corresponding ground-truth sentences. Note that, Base Transformer and our RSTNet are fed with the same visual features here.

images. Intuitively, the captions generated by our RSTNet are more accurate and distinguishable comparing to the original Transformer model.

In order to better demonstrate the usefulness of our RSTNet, we investigate the Karpathy COCO test dataset, and calculate all γ values for each image based on our RSTNet. Note that if a word appears more than once in the caption generation process for an image, we only extract one single average value as the *visualness* of the word in that image. We get a matrix $image2word \in \mathbb{R}^{5000\times10201}$, which can be used to query word *visualness* for each image. We then average the $image2word$ matrix over the image dimension, and get a vector $word2visualness \in \mathbb{R}^{10201}$ to represent the *visualness* of each word. Finally, in Fiugre 5, we visualize the representative top and bottom words and their corresponding images according to $word2visualness$ and $image2word$. High *visualness* words can be clearly identified in the image, while the images of low *visualness* words tend to contain random objects and scenes. These results are in accordance with people's intuition, which shows that our RSTNet can well distinguish visual and non-visual words, providing an intuitive explanation to its superior performance against the previous models.

5. Conclusion

In this paper, we present RSTNet, a novel Relationship-Sensitive Transformer-based model for image captioning, which encodes images with grid-augmented visual features and models visual and non-visual words with adaptive attention. On the one hand, our RSTNet incorporates relative spatial geometry features to compensate for the loss of spatial information when grid features are flattened. On the other hand, our RSTNet utilizes an adaptive attention module to dynamically measure the contribution of visual signals and language signal for word prediction at each time step. Extensive experiments on MS-COCO dataset and the visualization of the proposed *visualness* attribute demonstrate the effectiveness of our model.

Acknowledgement

This work is supported by the National Science Fund for Distinguished Young Scholars (No.62025603), the National Natural Science Foundation of China (No.U1705262, No. 62072386, No. 62072387, No. 62072389, No. 62002305, No.61772443, No.61802324 and No.61702136) and Guangdong Basic and Applied Basic Research FoundationNo.2019B1515120049). and CCF-Baidu Open Fund (OF2020008).

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016. 6

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018. 1, 2, 6, 7, 8

[3] Jyoti Aneja, Aditya Deshpande, and Alexander G Schwing. Convolutional image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5561–5570, 2018. 3

[4] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. 6

[5] Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Show, control and tell: A framework for generating controllable and grounded captions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8307–8316, 2019. 1, 2

[6] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10578–10587, 2020. 1, 2, 3, 6, 7, 8

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3, 4

[8] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *European conference on computer vision*, pages 15–29. Springer, 2010. 2

[9] Longteng Guo, Jing Liu, Xinxin Zhu, Peng Yao, Shichen Lu, and Hanqing Lu. Normalized and geometry-aware self-attention network for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10327–10336, 2020. 4

[10] Ankush Gupta, Yashaswi Verma, and CV Jawahar. Choosing linguistics over vision to describe images. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012. 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[12] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: Transforming objects into words. In *Advances in Neural Information Processing Systems*, pages 11137–11147, 2019. 1, 3, 4, 7

[13] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4634–4643, 2019. 2, 3, 7, 8

[14] Jiayi Ji, Yunpeng Luo, Xiaoshuai Sun, Fuhai Chen, Gen Luo, Yongjian Wu, Yue Gao, and Rongrong Ji. Improving image captioning by leveraging intra-and inter-layer global representation in transformer network. *arXiv preprint arXiv:2012.07061*, 2020. 2

[15] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10267–10276, 2020. 1, 2, 3, 6

[16] Wenhao Jiang, Lin Ma, Yu-Gang Jiang, Wei Liu, and Tong Zhang. Recurrent fusion network for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 499–515, 2018. 1, 7, 8

[17] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 2, 6

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 2

[19] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2891–2903, 2013. 2

[20] Guang Li, Linchao Zhu, Ping Liu, and Yi Yang. Entangled transformer for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8928–8937, 2019. 3, 8

[21] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 6

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 6

[23] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 375–383, 2017. 1, 2, 4

[24] Yunpeng Luo, Jiayi Ji, Xiaoshuai Sun, Liujuan Cao, Yongjian Wu, Feiyue Huang, Chia-Wen Lin, and Rongrong Ji. Dual-level collaborative transformer for image captioning. *arXiv preprint arXiv:2101.06462*, 2021. 2

[25] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L Yuille. Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090*, 2014. 1, 2

[26] Margaret Mitchell, Jesse Dodge, Amit Goyal, Kota Yamaguchi, Karl Stratos, Xufeng Han, Alyssa Mensch, Alexander Berg, Tamara Berg, and Hal Daumé III. Midge: Generating image descriptions from computer vision detections. In

Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pages 747–756, 2012. 2

[27] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24:1143–1151, 2011. 2

[28] Yingwei Pan, Ting Yao, Yehao Li, and Tao Mei. X-linear attention networks for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10971–10980, 2020. 1, 2, 3, 7, 8

[29] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002. 6

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1

[31] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024, 2017. 6, 7, 8

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[33] Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Herve Jegou, and Armand Joulin. Augmenting self-attention with persistent memory. *arXiv preprint arXiv:1907.01470*, 2019. 3

[34] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014. 1

[35] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 2

[36] Yoshitaka Ushiku, Masataka Yamaguchi, Yusuke Mukuta, and Tatsuya Harada. Common subspace for model and similarity: Phrase learning for caption generation from images. In *Proceedings of the IEEE international conference on computer vision*, pages 2668–2676, 2015. 2

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2, 3, 4

[38] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. 6

[39] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015. 1, 2

[40] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663, 2016. 1, 2

[41] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015. 1, 2

[42] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10685–10694, 2019. 1, 7, 8

[43] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 684–699, 2018. 7, 8