This ACCV 2020 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ACCV 2020
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/accv

Image Captioning through Image Transformer

Sen He*[1], Wentong Liao*[2], Hamed R. Tavakoli[3], Michael Yang[4]
Bodo Rosenhahn[2], and Nicolas Pugeault[5]

[1] CVSSP, University of Surrey, UK
[2] Leibniz University of Hanover, Germany
[3] Nokia Technologies, Finland
[4] University of Twente, Netherlands
[5] School of Computing Science, University of Glasgow
senhe752@gmail.com

Abstract. Automatic captioning of images is a task that combines the challenges of image analysis and text generation. One important aspect of captioning is the notion of attention: how to decide what to describe and in which order. Inspired by the successes in text analysis and translation, previous works have proposed the *transformer* architecture for image captioning. However, the structure between the *semantic units* in images (usually the detected regions from object detection model) and sentences (each single word) is different. Limited work has been done to adapt the transformer's internal architecture to images. In this work, we introduce the ***image transformer***, which consists of a modified encoding transformer and an implicit decoding transformer, motivated by the relative spatial relationship between image regions. Our design widens the original transformer layer's inner architecture to adapt to the structure of images. With only regions feature as inputs, our model achieves new state-of-the-art performance on both MSCOCO offline and online testing benchmarks. The code is available at https://github.com/wtliao/ImageTransformer.

1 Introduction

Image captioning is the task of describing the content of an image in words. The problem of automatic image captioning by AI systems has received a lot of attention in the recent years, due to the success of deep learning models for both language and image processing. Most image captioning approaches in the literature are based on a *translational* approach, with a visual encoder and a linguistic decoder. One challenge in automatic translation is that it cannot be done word by word, but that other words influence then meaning, and therefore the translation, of a word; this is even more true when translating across modalities, from images to text, where the system must decide *what* must be described in the image. A common solution to this challenge relies on attention mechanisms. For example, previous image captioning models try to solve

* Equal contribution

where to look in the image [1–4] (now partly solved by the Faster-RCNN object detection model [5]) in the encoding stage and use a recurrent neural network with attention mechanism in the decoding stage to generate the caption. But more than just to decide what to describe in the image, recent image captioning models propose to use attention to learn how regions of the image relate to each other, effectively encoding their *context* in the image. Graph convolutional neural networks [6] were first introduced to relate regions in the image; however, those approaches [7–10] usually require auxiliary models (e.g. visual relationship detection and/or attribute detection models) to build the visual scene graph in the image in the first place. In contrast, in the natural language processing field, the transformer architecture [11] was developed to relate embedded words in sentences, and can be trained end to end without auxiliary models explicitly detecting such relations. Recent image captioning models [12–14] adopted the transformer architectures to implicitly relate informative regions in the image through dot-product attention achieving state-of-the-art performance.

However, the transformer architecture was designed for machine translation of text. In a text, a word is either to the left or to the right of another word, with different distances. In contrast, images are two-dimensional (indeed, represent three-dimensional scenes), so that a region may not only be on the left or right of another region, it may also contain or be contained in another region. The relative spatial relationship between the semantic units in images has a larger degree of freedom than that in sentences. Furthermore, in the decoding stage of machine translation, a word is usually translated into another word in other languages (one to one decoding), whereas for an image region, we may describe its context, its attribute and/or its relationship with other regions (one to more decoding). One limitation of previous transformer-based image captioning models [12–14] is that they adopt the transformer's internal architecture designed for the machine translation, where each transformer layer contains a single (multi-head) dot-product attention module. In this paper, we introduce the ***image transformer*** for image captioning, where each transformer layer implements multiple sub-transformers, to encode spatial relationships between image regions and decode the diverse information in image regions.

The difference between our method and previous transformer based models [12, 14, 13] is that our method focuses on the *inner architectures* of the transformer layer, in which we widen the transformer module. Yao *et al.* [10] used a hierarchical concept in the encoding part of their model, our model focuses on the local spatial relationships for each query region whereas their method is a global tree hierarchy. Furthermore, our model does not require auxiliary models (*ie*, for visual relation detection and instance segmentation) to build the visual scene graph. Our encoding method can be viewed as the combination of a visual semantic graph and a spatial graph which use a transformer layer to implicitly combine them without auxiliary relationship and attribute detectors.

The contributions of this paper can be summarised as follows:

Fig. 1: Image captioning vs machine translation.

- We propose a novel internal architecture for the transformer layer adapted to the image captioning task, with a modified attention module suited to the complex natural structure of image regions.
- We report thorough experiments and ablation study were done in the work to validate our proposed architecture, state-of-the-art performance was achieved on the MSCOCO image captioning offline and online testing dataset with only region features as input.

The rest of the paper is organized as follows: Sec. 2 reviews the related attention-based image captioning models; Sec. 3 introduces the standard transformer model and our proposed image transformer; followed by the experiment results and analysis in Sec. 4; finally, we will conclude this paper in Sec. 5.

2 Related Work

We characterize current attention-based image captioning models into single-stage attention models, two-stages attention models, visual scene graph based models, and transformer-based models. We will review them one by one in this section.

2.1 Single-Stage Attention Based Image Captioning

Single-stage attention-based image captioning models are the models where attention is applied at the decoding stage, where the decoder attends to the most informative region [15] in the image when generating a corresponding word.

The availability of large-scale annotated datasets [16, 17] enabled the training of deep models for image captioning. Vinyals *et al.* [18] proposed the first deep model for image captioning. Their model uses a CNN pre-trained on ImageNet [16] to encode the image, then a LSTM [19] based language model is used to decode the image features into a sequence of words. Xu *et al.* [1] introduced an attention mechanism into image captioning during the generation of each word, based on the hidden state of their language model and the previous generated word. Their attention module generates a matrix to weight each receptive field in the encoded feature map, and then feed the weighted feature map and the previous generated word to the language model to generate the next word. Instead of only attending to the receptive field in the encoded feature map, Chen *et al.* [2] added a feature channel attention module, their channel attention module

re-weight each feature channel during the generation of each word. Not all the words in the sentence have a correspondence in the image, so Lu *et al.* [20] proposed an adaptive attention approach, where their model has a visual sentinel which adaptively decides when and where to rely on the visual information.

The single-stage attention model is computational efficient, but lacks accurate positioning of informative regions in the original image.

2.2 Two-Stages Attention Based Image Captioning

Two stage attention models consists of *bottom-up* attention and *top-down* attention, where bottom-up attention first uses object detection models to detect multiple informative regions in the image, then top-down attention attends to the most relevant detected regions when generating a word.

Instead of relying on the coarse receptive fields as informative regions in the image, as single-stage attention models do, Anderson *et al.* [3] train the detection models on the *Visual Genome* dataset [21]. The trained detection models can detect $10 - 100$ informative regions in the image. They then use a two-layers LSTM network as decoder, where the first layer generates a state vector based on the embedded word vector and the mean feature of the detected regions and the second layer uses the state vector from the previous layer to generate a weight for each detected region. The weighted sum of detected regions feature is used as a context vector for predicting the next word. Lu *et al.* [4] developed a similar network, but with a detection model trained on *MSCOCO* [22], which is a smaller dataset than *Visual Genome*, and therefore less informative regions are detected.

The performance of two-stage attention based image captioning models is improved a lot against single-stage attention based models. However, each detected region is isolated from others, lacking the relationship with other regions.

2.3 Visual Scene Graph Based Image Captioning

Visual scene graph based image captioning models extend two-stage attention models by injecting a graph convolutional neural network to relate detected informative regions, and therefore refine their features before feeding into the decoder.

Yao *et al.* [7] developed a model which consists of a semantic scene graph and a spatial scene graph. In the semantic scene graph, each region is connected with other semantically related regions, those relationships are usually determined by a visual relationship detector among a union box. In the spatial scene graph, the relationship between two regions is defined by their relative positions. Then the feature of each node in the scene graph is refined with their related nodes through graph neural networks [6]. Yang *et al.* [8] use an auto-encoder, where they first encode the graph structure in the sentence based on the SPICE [23] evaluation metric to learn a dictionary, then the semantic scene graph is encoded using the learnt dictionary. The previous two works treat the semantic relationships as edges in the scene graph, while Guo *et al.* [9] treat them as nodes in the scene

graph. Also, their decoder focuses on different aspects of a region. Yao *et al.*[10] further introduces the tree hierarchy and instance level feature into the scene graph.

Introducing the graph neural network to relate informative regions yields a sizeable performance improvement for image captioning models, compared to two-stage attention models. However, it requires auxiliary models to detect and build the scene graph at first. Also those models usually have two parallel streams, one responsible for the semantic scene graph and another for spatial scene graph, which is computationally inefficient.

2.4 Transformer Based Image Captioning

Transformer based image captioning models use the dot-product attention mechanism to relate informative regions implicitly.

Since the introduction of original transformer model [11], more advanced architectures were proposed for machine translation based on the structure or the natural characteristic of sentences [24–26]. In image captioning, AoANet [12] uses the original internal transformer layer architecture, with the addition of a *gated linear layer* [27] on top of the multi-head attention. The object relation network [14] injects the relative spatial attention into the dot-product attention. Another interesting result described by Herdade *et al.* [14] is that the simple position encoding (as proposed in the original transformer) did not improve image captioning performance. The entangled transformer model [13] features a dual parallel transformer to encode and refine visual and semantic information in the image, which is fused through gated bilateral controller.

Compared to scene graph based image captioning models, transformer based models do not require auxiliary models to detect and build the scene graph at first, which is more computational efficient. However current transformer based models still use the inner architecture of the original transformer, designed for text, where each transformer layer has a single multi-head dot-product attention refining module. This structure does not allow to model the full complexity of relations between image regions, therefore we propose to change the inner architecture of the transformer layer to adapt it to image data. We widen the transformer layer, such that each transformer layer has multiple refining modules for different aspects of regions both in the encoding and decoding stages.

3 Image Transformer

In this section, we first review the original transformer layer [11], we then elaborate the encoding and decoding part for the proposed ***image transformer*** architecture.

3.1 Transformer Layer

A transformer consists of a stack of multi-head dot-product attention based transformer refining layer.

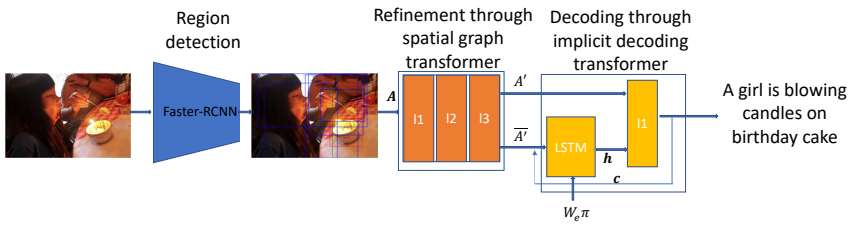


Fig. 2: The overall architecture of our model, the refinement part consists of 3 stacks of spatial graph transformer layer, and the decoding part has a LSTM layer with a implicit decoding transformer layer.

In each layer, for a given input $A \in \mathbb{R}^{N\times D}$, consisting of N entries of D dimensions. In natural language processing, the input entry can be the embedded feature of a word in a sentence, and in computer vision or image captioning, the input entry can be the feature describing a region in an image. The key function of transformer is to refine each entry with other entries through multi-head dot-product attention. Each layer of a transformer first transforms the input into queries ($Q = AW_Q$, $W_Q \in \mathbb{R}^{D\times D_k}$), keys ($K = AW_K$, $W_K \in \mathbb{R}^{D\times D_k}$) and values ($V = AW_V$, $W_A \in \mathbb{R}^{D\times D_v}$) though linear transformations, then the scaled dot-product attention is defined by:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{D_k}}\right)V, \tag{1}$$

where D_k is the dimension of the key vector and D_v the dimension of the value vector ($D = D_k = D_v$ in the implementation). To improve the performance of the attention layer, multi-head attention is applied:

$$\begin{aligned}&\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W_O,\\ &\text{head}_i = \text{Attention}(AW_{Q_i}, AW_{K_i}, AW_{V_i}).\end{aligned} \tag{2}$$

The output from the multi-head attention is then added with the input and normalised:

$$A_m = \text{Norm}(A + \text{MultiHead}(Q, K, V)), \tag{3}$$

where $\text{Norm}(\cdot)$ denote layer normalisation.

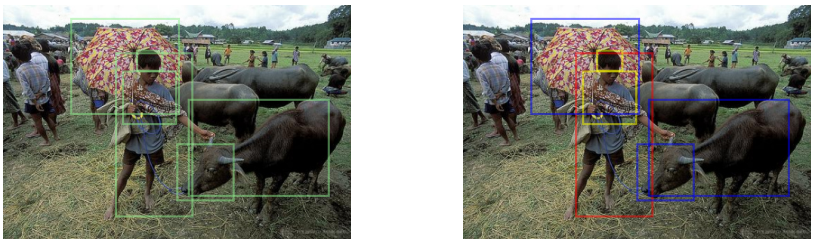
The transformer implements residual connections in each module, such that the final output of a transformer layer is:

$$A^{'} = \text{Norm}(A_m + \phi(A_m W_f)), \tag{4}$$

where ϕ is a feed-forward network with non-linearity.

Each refining layer takes the output of its previous layer as input (the first layer takes the original input). The decoding part is also a stack of transformer refining layers, which take the output of encoding part as well as the embedded features of previous predicted word.

3.2 Spatial Graph Encoding Transformer Layer



(a) (b)

Fig. 3: (a) Image with detected regions; (b) An example of query region in the image (man in the red bounding box), and its neighbor regions (regions in blue bounding boxes, bull, umbrella, etc), child regions (regions in the yellow bounding boxes, hair,cloth).

In contrast to the original transformer, which only considers spatial relationships between query and key pairs as ***neighborhood***, we propose to use a spatial graph transformer in the encoding part, where we consider three common categories of spatial relationship for each query region in a graph structure: *parent*, *neighbor*, and *child* (an example shown in Fig. 3). Thus we widen each transformer layer by adding three sub-transformer layers in parallel in each layer, each sub-transformer responsible for a category of spatial relationship, all sharing the same query. In the encoding stage, we define the relative spatial relationship between two regions based on their overlap. We first compute the graph adjacent matrices $\Omega_p \in \mathbb{R}^{N\times N}$ (parent node adjacent matrix), $\Omega_n \in \mathbb{R}^{\in N\times N}$ (neighbor node adjacent matrix), and $\Omega_c \in \mathbb{R}^{\in N\times N}$ (child node adjacent matrix) for all regions in the image:

$$\begin{aligned} \Omega_p[l,m] &= \begin{cases} 1, \text{ if } \dfrac{\text{Area}(l\cap m)}{\text{Area}(l)} \geqslant \epsilon \text{ and } \dfrac{\text{Area}(l\cap m)}{\text{Area}(l)} > \dfrac{\text{Area}(l\cap m)}{\text{Area}(m)} \\ 0, \text{ otherwise.} \end{cases} \\ \Omega_c[l,m] &= \Omega_p[m,l] \\ \text{with} &\sum_{i\in\{p,n,c\}} \Omega_i[l,m] = 1 \end{aligned} \tag{5}$$

where $\epsilon = 0.9$ in our experiment. The spatial graph adjacent matrices are used as the spatial hard attention embedded into each sub-transformer to combine the output of each sub-transformer in the encoder. More specifically, the original

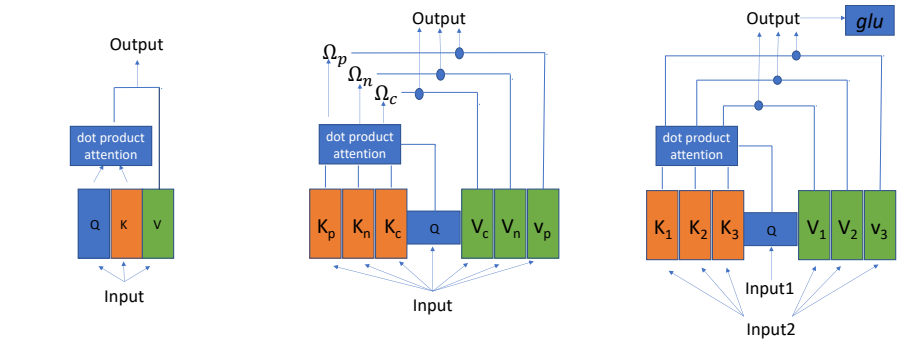


Fig. 4: The difference between the original transformer layer and the proposed encoding and decoding transformer layers.

encoding transformer defined in Eqs. (1) and (2) are reformulated as:

$$\text{Attention}(Q, K_i, V_i) = \Omega_i \circ \text{Softmax}\left(\frac{QK_i^T}{\sqrt{d}}\right) V_i, \tag{6}$$

$\circ$ is the Hadamard product, and

$$A_m = \text{Norm}\left(A + \sum_{i \in \{p,n,c\}} \text{MultiHead}(Q, K_i, V_i)\right). \tag{7}$$

As we widen the transformer, we halve the number of stacks in the encoder to achieve similar complexity as the original one (3 stacks, while the original transformer features 6 stacks). With our formulation, we combined the spatial graph and semantic graph (the scene graph based methods [7, 9] require two branches to encode them) into a transformer layer. Note that the original transformer architecture is a special case of the proposed architecture, when no region in the image either contains or is contained by another.

3.3 Implicit Decoding Transformer Layer

Our decoder consists of a LSTM [28] layer and an implicit transformer decoding layer, which we proposed to decode the diverse information in a region in the image. The LSTM layer is a common memory module and the transformer layer infers the most relevant region in the image through dot product attention.

At first, the LSTM layer receives the mean of the output ($\overline{A} = \frac{1}{N}\sum_{i=1}^{N} A_i^{'}$) from the encoding transformer, a context vector (c_{t-1}) at last time step and the embedded feature vector of current word in the ground truth sentence:

$$\begin{aligned} x_t &= [W_e \pi_t, \overline{A} + c_{t-1}] \\ h_t, m_t &= \text{LSTM}(x_t, h_{t-1}, m_{t-1}) \end{aligned} \tag{8}$$

Where, W_e is the word embedding matrix, π_t is the t^{th} word in the ground truth. The output state h_t is then transformed linearly and treated as the query for the input of the implicit decoding transformer layer. The difference between the original transformer layer and our implicit decoding transformer layer is that we also widen the decoding transformer layer by adding several sub-transformers in parallel in one layer, such that each sub-transformer can implicitly decode different aspects of a region. It is formalised as follows:

$$A^D_{t,i} = \text{MultiHead}(W_{DQ}h_t, W_{DKi}A^{'}, W_{DVi}A^{'}) \tag{9}$$

Then, the mean of the sub-transformers' output is passed through a gated linear layer (GLU) [27] to extract the new context vector (c_t) at the current step by channel:

$$c_t = \text{GLU}\left(h_t, \frac{1}{M}\sum_{i=1}^{M} A^D_{t,i}\right) \tag{10}$$

The context vector is then used to predict the probability of word at time step t:

$$p(y_t|y_{1:t-1}) = \text{Softmax}(w_p c_t + b_p) \tag{11}$$

The overall architecture of our model is illustrated in Fig. 2, and the difference between the original transformer layer and our proposed encoding and decoding transformer layer is showed in Fig. 4.

3.4 Training Objectives

Given a target ground truth as a sequence of words $y^*_{1:T}$, for training the model parameters θ, we follow the previous method, such that we first train the model with cross-entropy loss:

$$L_{XE}(\theta) = -\sum_{t=1}^{T} \log(p_\theta(y^*_t|y^*_{1:t-1})) \tag{12}$$

then followed by self-critical reinforced training [29] optimizing the CIDEr score [30]:

$$L_R(\theta) = -E_{(y_{1:T}\sim p_\theta)}[r(y_{1:T})] \tag{13}$$

where r is the score function and the gradient is approximated by:

$$\bigtriangledown_\theta \approx -(r(y^s_{1:T}) - (\hat{y}_{1:T}))\bigtriangledown_\theta \log p_\theta(y^s_{1:T}) \tag{14}$$

4 Experiment

4.1 Datasets and Evaluation Metrics

Our model is trained on the MSCOCO image captioning dataset [17]. We follow Karpathy's splits [32], with 11,3287 images in the training set, 5,000 images in

model	Bleu1	Bleu4	METEOR	ROUGE-L	CIDEr	SPICE
single-stage model						
Att2all[29]	-	34.2	26.7	55.7	114.0	-
two-stages model						
n-babytalk[4]	75.5	34.7	27.1	-	107.2	20.1
up-down[3]	79.8	36.3	27.7	56.9	120.1	21.4
scene graph based model						
GCN-LSTM*[7]	80.9	38.3	28.6	58.5	128.7	22.1
AUTO-ENC[8]	80.8	38.4	28.4	58.6	127.8	22.1
ALV*[9]	-	38.4	28.5	58.4	128.6	22.0
GCN-LSTM-HIP*†[10]	-	39.1	28.9	**59.2**	130.6	22.3
transformer based model						
Entangle-T*[13]	**81.5**	**39.9**	28.9	59.0	127.6	22.6
AoA[12]	80.2	38.9	**29.2**	58.8	129.8	22.4
VORN[14]	80.5	38.6	28.7	58.4	128.3	22.6
Ours	80.8	39.5	29.1	59.0	**130.8**	**22.8**

Table 1: Comparison on MSCOCO Karpathy offline test split. * means fusion of two models. † means SENet [31] as feature extraction backbone

the validation set and 5,000 images in the test set. Each image has 5 captions as ground truth. We discard the words which occur less than 4 times, and the final vocabulary size is 10,369. We test our model on both Karpathy's offline test set (5,000 images) and MSCOCO online testing datasets (40,775 images). We use Bleu [33], METEOR [34], ROUGE-L [35], CIDEr [30], and SPICE [23] as evaluation metrics.

4.2 Implementation Details

Following previous work, we first train Faster R-CNN on Visual Genome [21], use resnet-101 [36] as backbone, pretrained on ImageNet [16]. For each image, we can detect $10-100$ informative regions, the boundaries of each are first normalised and then used to compute the spatial graph matrices. We then train our proposed model for image captioning using the computed spatial graph matrices and extracted features for each image region. We first train our model with *cross-entropy* loss for 25 epochs, the initial learning rate is set to 2×10^{-3}, and we decay the learning rate by 0.8 every 3 epochs. Our model is optimized through Adam [37] with a batch size of 10. We then further optimize our model by reinforced learning for another 35 epochs. The size of the decoder's LSTM layer is set to 1024, and beam search of size 3 is used in the inference stage.

4.3 Experiment Results

We compare our model's performance with published image captioning models. The compared models include the top performing single-stage attention model, Att2all [29]; two-stages attention based models, n-babytalk [4] and up-down [3]; visual scene graph based models, GCN-LSTM [7], AUTO-ENC [8], ALV [9], GCN-LSTM-HIP [10]; and transformer based models Entangle-T [13], AoA

model	B1		B4		M		R		C	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
scene graph based model										
GCN-LSTM*[7]	80.8	95.9	38.7	69.7	28.5	37.6	58.5	73.4	125.3	126.5
AUTO-ENC*[8]	-	-	38.5	69.7	28.2	37.2	58.6	73.6	123.8	126.5
ALV*[9]	79.9	94.7	37.4	68.3	28.2	37.1	57.9	72.8	123.1	125.5
GCN-LSTM-HIP*†[10]	81.6	95.9	39.3	71.0	28.8	38.1	59.0	74.1	127.9	130.2
transformer based model										
Entangle-T*[13]	81.2	95.0	38.9	70.2	28.6	38.0	58.6	73.9	122.1	124.4
AoA[12]	81.0	95.0	39.4	71.2	29.1	38.5	58.9	74.5	126.9	129.6
Ours	81.2	95.4	39.6	71.5	29.1	38.4	59.2	74.5	127.4	129.6

Table 2: Leaderboard of recent published models on the MSCOCO online testing server. * means fusion of two models. † means SENet [31] as feature extraction backbone

[12], VORN [14]. The comparison on the MSCOCO Karpathy offline test set is illustrated in Table 1. Our model achieves new state-of-the-art on the CIDEr and SPICE score, while other evaluation scores are comparable to the previous top performing models. Note that because most visual scene graph based models fused semantic and spatial scene graph, and require the auxiliary models to build the scene graph at first, our model is more computationally efficient. VORN [14] also integrated spatial attention in their model, and our model performs better than them among all kinds of evaluation metrics, which shows the superiority of our spatial graph transformer layer. The MSCOCO online testing results are listed in Tab. 2, our model outperforms previous transformer based model on several evaluation metrics.

4.4 Ablation Study and Analysis

In the ablation study, we use AoA [12] as a strong baseline [6] (with a single multi-head dot-product attention module per layer), which add the gated linear layer [27] on top of the multi-head attention. In the encoder part, we study the spatial relationship's effect in the encoder, where we ablate the spatial relationship by simply taking the mean output of three sub-transformers in each layer by reformulating Eqs. 6 and 7 as: $\text{Attention}(Q, K_i, V_i) = \text{Softmax}\left(\frac{QK_i^T}{\sqrt{d}}\right)V_i, A_m = \text{Norm}\left(A + \frac{1}{3}\sum_{i\in\{p,n,c\}}\text{MultiHead}(Q, K_i, V_i)\right)$. We also study where to use our proposed spatial graph encoding transformer layer in the encoding part: in the first layer, second layer, third layer or three of them? In the decoding part, we study the effect of the number of sub-transformers (M in Eq. 10) in the implicit decoding transformer layer.

As we can see from Tab. 3, by widening the encoding transformer layer, there is a significant improvement on the model's performance. While not every layers in the encoding transformer are equal, when we use our proposed transformer layer at the top layer of the encoding part, the improvement was reduced. This

[6] Our experiments are based on the code released at: https://github.com/husthuaan/AoANet

model	Bleu1	Bleu4	METEOR	ROUGE-L	CIDEr	SPICE
baseline(AoA)	77.0	36.5	28.1	57.1	116.6	21.3
positions to embed our spatial graph encoding transformer layer						
baseline+layer1	77.8	36.8	28.3	57.3	118.1	21.3
baseline+layer2	77.2	36.8	28.3	57.3	118.2	21.3
baseline+layer3	77.0	37.0	28.2	57.1	117.3	21.2
baseline+layer1,2,3	77.5	37.0	28.3	57.2	118.2	21.4
effect of spatial relationships in the encoder						
baseline+layer1,2,3 w/o spatial rela	77.5	36.8	28.2	57.1	117.8	21.4
number of sub-transformers in the implicit decoding transformer layer						
baseline+layer1,2,3 (M=2)	77.5	37.6	28.4	57.4	118.8	21.3
baseline+layer1,2,3 (M=3)	78.0	37.4	28.4	57.6	119.1	21.6
baseline+layer1,2,3 (M=4)	77.5	37.8	28.4	57.5	118.6	21.4

Table 3: Ablation study, results reported without RL training. baseline+layer1 means only the first layer of encoding transformer uses our proposed spatial transformer layer, other layers use the original one. M is the number of sub-transformers in the decoding transformer layer.

may be because spatial relationships at the top layer of the transformer are not as informative, we use our spatial transformer layer at all layers in the encoding part. When we reduce the spatial relationship in our proposed wider transformer layer, there is also some performance reduction, which shows the importance of the spatial relationship in our design. After widening the decoding transformer, the improvement was further increased (the CIDEr score increased from 118.2 to 119.1 after widening the decoding transformer layer with 3 sub-transformers), while not more wider gives better result, with 4 sub-transformers in the decoding transformer layer, there is some performance decrease, therefore the final design of our decoding transformer layer has 3 sub-transformers in parallel. The qualitative example of our models results is illustrated in Fig. 5. As we can see, the baseline model without spatial relationships wrongly described the police officers on a red bus (top right), and people on a train (bottom left).

Encoding implicit graph visualisation: the transformer layer can be seen as an implicit graph, which relates the informative regions through dot-product attention. Here we visualise how our proposed spatial graph transformer layer learn to connect the informative regions through attention in Fig. 6. In the top example, the original transformer layer strongly relates the train with the people on the mountain, yields wrong description, while our proposed transformer layer relates the train with the tracks and mountain; in the bottom example, the original transformer relates the bear with its reflection in water and treats them as 'two bears', while our transformer can distinguish the bear from its reflection and relate it to the snow area.

Decoding feature space visualisation: We also visualised the output of our decoding transformer layer (Fig. 7). Compared to the original decoding transformer layer, which only has one sub-transformer inside it. The output of our proposed implicit decoding transformer layer covers a larger area in the reduced feature

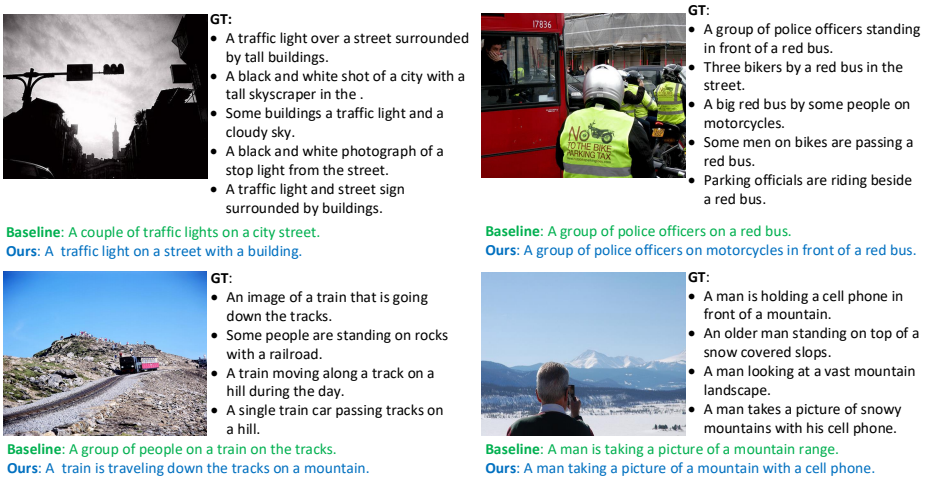


Fig. 5: Qualitative examples from our method on the MSCOCO image captioning dataset [17], compared against the ground truth annotation and a strong baseline method (AoA [12]).

space than the original one, which means that our decoding transformer layer decoding more information in the image regions. In the original feature space (1,024 dimensions) from the output of decoding transformer layer, we compute the trace of the feature maps' co-variance matrix from 1,000 examples, the trace for original transformer layer is 30.40 compared to 454.57 for our wider decoding transformer layer, which indicates that our design enables the decoder's output to cover a larger area in the feature space. However, it looks like individual sub-transformers in the decoding transformer layer still do not learn to disentangle different factors in the feature space (as there is no distinct cluster from the output of each sub-transformer), we speculate this is because we have no direct supervision to their output, which may not able to learn the disentangled feature automatically [38].

5 Discussion and Conclusion

In this work, we introduced the ***image transformer*** architecture. The core idea behind the proposed architecture is to widen the original transformer layer, designed for machine translation, to adapt it to the structure of images. In the encoder, we widen the transformer layer by exploiting the spatial relationships between image regions, and in the decoder, the wider transformer layer can decode more information in the image regions. Extensive experiments were done to show the superiority of the proposed model, the qualitative and quantitative analyses were illustrated in the experiments to validate the proposed encoding and decoding transformer layer. Compared to the previous top models in image captioning, our model achieves a new state-of-the-art SPICE score, while in the

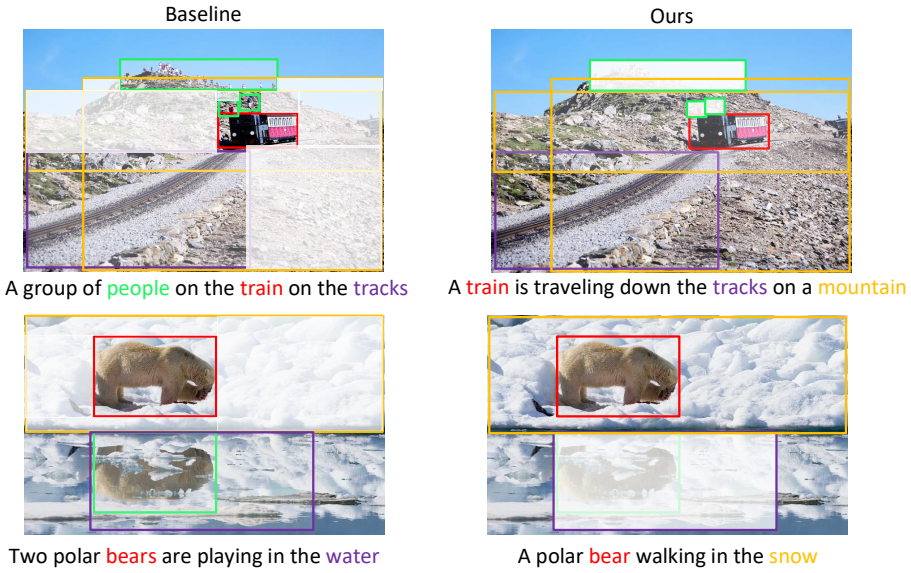
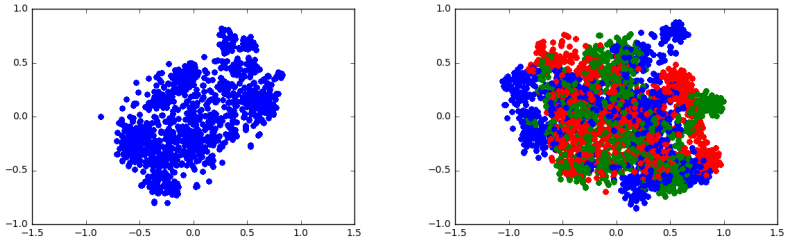


Fig. 6: A visualization of how the query region relates to its other key regions through attention, the region in the red bounding box is the query region and other regions are key regions. The transparency of each key region shows its dot-product attention weight with the query region. Higher transparency means larger dot-product attention weight, vice versa.



(a) original (b) ours

Fig. 7: t-SNE [39] visualisation of the output from decoding transformer layer (1,000 examples), different color represent the output from different sub-transformers in the decoder in our model.

other evaluation metrics, our model is either comparable or outperforms the previous best models, with a better computational efficiency.

We hope our work can inspire the community to develop more advanced transformer based architectures that can not only benefit image captioning but also other computer vision tasks which need relational attention inside it. Our code will be shared with the community to support future research.

References

1. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning. (2015) 2048–2057
2. Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., Chua, T.S.: Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 5659–5667
3. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: 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. (2018) 6077–6086
4. Lu, J., Yang, J., Batra, D., Parikh, D.: Neural baby talk. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 7219–7228
5. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. (2015) 91–99
6. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
7. Yao, T., Pan, Y., Li, Y., Mei, T.: Exploring visual relationship for image captioning. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 684–699
8. Yang, X., Tang, K., Zhang, H., Cai, J.: Auto-encoding scene graphs for image captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 10685–10694
9. Guo, L., Liu, J., Tang, J., Li, J., Luo, W., Lu, H.: Aligning linguistic words and visual semantic units for image captioning. In: Proceedings of the 27th ACM International Conference on Multimedia. (2019) 765–773
10. Yao, T., Pan, Y., Li, Y., Mei, T.: Hierarchy parsing for image captioning. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 2621–2629
11. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. (2017) 5998–6008
12. Huang, L., Wang, W., Chen, J., Wei, X.Y.: Attention on attention for image captioning. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 4634–4643
13. Li, G., Zhu, L., Liu, P., Yang, Y.: Entangled transformer for image captioning. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 8928–8937
14. Herdade, S., Kappeler, A., Boakye, K., Soares, J.: Image captioning: Transforming objects into words. In: Advances in Neural Information Processing Systems. (2019) 11135–11145
15. Luo, W., Li, Y., Urtasun, R., Zemel, R.: Understanding the effective receptive field in deep convolutional neural networks. In: Advances in neural information processing systems. (2016) 4898–4906
16. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, Ieee (2009) 248–255

17. Chen, X., Fang, H., Lin, T.Y., Vedantam, R., Gupta, S., Dollár, P., Zitnick, C.L.: Microsoft coco captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325 (2015)
18. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 3156–3164
19. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: Continual prediction with lstm. Neural Computation **12** (2000) 2451–2471
20. Lu, J., Xiong, C., Parikh, D., Socher, R.: 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. (2017) 375–383
21. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International Journal of Computer Vision **123** (2017) 32–73
22. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision, Springer (2014) 740–755
23. Anderson, P., Fernando, B., Johnson, M., Gould, S.: Spice: Semantic propositional image caption evaluation. In: European Conference on Computer Vision, Springer (2016) 382–398
24. Hao, J., Wang, X., Shi, S., Zhang, J., Tu, Z.: Multi-granularity self-attention for neural machine translation. arXiv preprint arXiv:1909.02222 (2019)
25. Wang, X., Tu, Z., Wang, L., Shi, S.: Self-attention with structural position representations. arXiv preprint arXiv:1909.00383 (2019)
26. Wang, Y.S., Lee, H.Y., Chen, Y.N.: Tree transformer: Integrating tree structures into self-attention. arXiv preprint arXiv:1909.06639 (2019)
27. Dauphin, Y.N., Fan, A., Auli, M., Grangier, D.: Language modeling with gated convolutional networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR (2017) 933–941
28. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9** (1997) 1735–1780
29. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 7008–7024
30. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: Cider: Consensus-based image description evaluation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 4566–4575
31. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 7132–7141
32. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 3128–3137
33. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting on association for computational linguistics, Association for Computational Linguistics (2002) 311–318
34. Banerjee, S., Lavie, A.: 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. (2005) 65–72

35. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Proc. ACL workshop on Text Summarization Branches Out. (2004) 10
36. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
37. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
38. Locatello, F., Bauer, S., Lucic, M., Rätsch, G., Gelly, S., Schölkopf, B., Bachem, O.: Challenging common assumptions in the unsupervised learning of disentangled representations. In: Proceedings of the 36th International Conference on Machine Learning-Volume 97, JMLR (2019) 4114–4124
39. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9** (2008) 2579–2605