Normalized and Geometry-Aware Self-Attention Network for Image Captioning

Longteng Guo [1,2] Jing Liu [1] Xinxin Zhu [1] Peng Yao [3] Shichen Lu [4] Hanqing Lu [1]

[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]University of Science and Technology Beijing [4]Wuhan University
{longteng.guo,jliu,xinxin.zhu,luhq}@nlpr.ia.ac.cn,S20180598@xs.ustb.edu.cn,sclu@whu.edu.cn

Abstract

Self-attention (SA) network has shown profound value in image captioning. In this paper, we improve SA from two aspects to promote the performance of image captioning. First, we propose Normalized Self-Attention (NSA), a reparameterization of SA that brings the benefits of normalization inside SA. While normalization is previously only applied outside SA, we introduce a novel normalization method and demonstrate that it is both possible and beneficial to perform it on the hidden activations inside SA. Second, to compensate for the major limit of Transformer that it fails to model the geometry structure of the input objects, we propose a class of Geometry-aware Self-Attention (GSA) that extends SA to explicitly and efficiently consider the relative geometry relations between the objects in the image. To construct our image captioning model, we combine the two modules and apply it to the vanilla self-attention network. We extensively evaluate our proposals on MS-COCO image captioning dataset and superior results are achieved when comparing to state-of-the-art approaches. Further experiments on three challenging tasks, i.e. *video captioning, machine translation, and visual question answering, show the generality of our methods.*

1. Introduction

Automatically generating captions for images, namely image captioning [20, 40], has emerged as a prominent research problem at the intersection of computer vision (CV) and natural language processing (NLP). This task is challenging as it requires to first recognize the objects in the image, the relationships between them, and finally properly organize and describe them in natural language.

Inspired by the sequence-to-sequence model for machine translation, most image captioning approaches adopt an encoder-decoder paradigm, which uses a deep convolutional neural network (CNN) to encode the input image as a vectorial representation, and a recurrent neural network (RNN) based caption decoder to generate the output caption. Recently, *self-attention* (SA) networks, denoted as SANs, have been introduced by [46, 43] to replace conventional RNNs in image captioning. Since its first introduction in Transformer [32], SA and its variants have shown promising empirical results in a wide range of CV [45, 16, 37, 10, 24, 9] and NLP [30, 8, 41] tasks. Although SAN-based framework has achieved state-of-the-art performance in image captioning, it remains two problems to be solved.

Firstly, SA is susceptible to the internal covariate shift [18] problem. Typically, SA is regarded as a mapping of a set of query and key/value pairs. We observe, from another perspective, that computation of the attention weights in SA could be considered as feeding the queries into a fully-connected layer, whose parameters are dynamically computed according to the inputs. Problem could happen when the distribution of the queries shifts due to the change in network parameters during training. That is, the subsequent layers have to continuously adapt to the new input distribution, and consequently, SA may not be learned effectively. This problem is called "Internal Covariate Shift" in [18] — the tendency that the distribution of activations drifts during training in a feed-forward network.

To eliminate the internal covariate shift problem inside SA, in this paper, we introduce an effective reparameterization of SA, named Normalized Self-Attention (NSA). NSA performs a novel normalization method on the hidden activations of SA to fix their distributions. By doing so, we can effectively decouple the fully-connected layer's parameters from those of other layers, leading to a better-conditioned optimization of SA. While Layer Normalization (LN) [4]

is proven to be very critical for enabling the convergence of Transformer, however, LN is only applied *outside* SA blocks. To our knowledge, there has not been any deep exploration to find a suitable normalization method *inside* SA. We demonstrate that our NSA can collaborate with LN to bring improved generalization for SA-based networks.

Another critical issue in SA is its inability to model the geometric relationships among input elements. The vanilla self-attention treats its inputs as "*bag-of-features*", simply neglecting their structure and the relationships between them. However, the objects in the image, from which the region-based visual features are extracted for image captioning, inherently have geometric structure — 2D spatial layout and variations in scale/aspect ratio. Such inherent geometric relationships between objects play a very complex yet critical role in understanding the image content. One common solution to inject position information into SA is adding representations of absolute positions to each element of the inputs, as is often used in the case of 1D sentences. Nonetheless, this solution does not work well for image captioning because the 2D geometry relations between objects are harder to infer from their absolute positions.

We present a more efficient approach to the above problem: explicitly incorporating *relative* geometry relationships between objects into SA. The module is named Geometry-aware Self-Attention (GSA). GSA extends the original attention weight into two components: the original content-based weight, and a new geometric bias, which is efficiently calculated by the relative geometry relations and, importantly, the *content* of the associated elements, *i.e.* query or key.

By combining both NSA and GSA, we obtain an enhanced SA module. We then construct our Normalized and Geometry-aware Self-Attention Network, namely NG-SAN, by replacing the vanilla SA modules in the encoder of the self-attention network with the proposed one. Extensive experiments on MS-COCO validates the effectiveness of our proposals. In particular, our NG-SAN establishes a new state-of-the-art on the MS-COCO evaluation sever, improving the best single-model result in terms of CIDEr from 125.5 to 128.6. To demonstrate the generality of NSA, we further present video captioning, machine translation, and visual question answering experiments on the VATEX, WMT 2014 English-to-German, and VQA-v2 datasets, respectively. On top of the strong Transformer-based baselines, our methods can consistently increase accuracies on all tasks at a negligible extra computational cost.

To summarize, the main contributions of this paper are three-fold:

- We presented Normalized Self-Attention, an effective reparameterization of self-attention, which brings the benefits of normalization technique inside SA.
- We introduce a class of Geometry-aware Self-Attention that explicitly makes use of the relative geometry relationships and the content of objects to aid image understanding.
- By combining the two modules and apply it on the self-attention network, we establish a new state-of-the-art on the MS-COCO image captioning benchmark. Further experiments on video captioning, machine translation, and visual question answering tasks demonstrate the generality of our methods.

2. Related Work

2.1. Image Captioning

Existing image captioning approaches typically follows the CNN-RNN architecture [36]. Recently, a variety of improving works have been proposed. [40] introduces soft and hard attention mechanisms to automatically focus on salient objects when generating each word. [13] mimics human polishing process with a ruminant decoder. [2] uses an object detector to propose salient image regions (objects) and extract for each object a feature vector, which are then used as inputs for attention mechanism. [28] introduces reinforcement-learning with a self-critical reward for model training. Recently, [46] and [43] propose to replace conventional RNN with the Transformer architecture, achieving state-of-the-art performance. However, more deep exploration of the self-attention module in Transformer is not conducted on the task of image captioning, which motivates our work in this paper.

2.2. Normalization

Normalization [18] has become a critical ingredient in constructing a deep neural network. It is proposed by Batch normalization (BN) [18] to control the distributions of the internal activations of feed-forward neural networks, thereby reducing internal covariate shift. Several variants of normalization method such as Layer Normalization (LN) [4], Instance Normalization (IN) [31], and Group Normalization [39] have been developed mainly to reduce the mini-batch dependencies inherent in BN. LN operates along the channel dimension for each individual element in an example. IN performs BN-like computation but only for each sample. Though BN and LN have been adopted in networks that contain the SA module, *e.g.* Transformer, they are typically used outside the SA module. For the first time, our normalized self-attention brings the benefit of normalization inside the SA module.

2.3. Position encoding in self-attention networks

To inject sequence ordering into SA module, in Transformer, absolute position encodings based on sinusoids are

added to the input elements both in the encoder and decoder. Recently, [29] modulates SA by incorporating the relative distances between sequence elements. [16] proposes an SA-like module for object detection, which multiplies a new relation weight on the original self-attention weight, and is used by [15] in Transformer. Its relation weight is computed solely with the relative coordinates and sizes between bounding boxes. Different from these works, our GSA module explores a broader range of geometric biases that involve not only the geometry information but also the content of the associated objects.

3. Preliminaries

3.1. Self-Attention (SA)

We first review a basic form of self-attention, called "Scaled Dot-Product Attention", which is first proposed as a core component in Transformer.

The self-attention layer first transforms a set of N d_k-dimensional vectors, packed into a matrix $X \in \mathbb{R}^{N\times d_k}$, into queries $Q \in \mathbb{R}^{N\times d}$, keys $K \in \mathbb{R}^{N\times d}$, and values $V \in \mathbb{R}^{N\times d}$ given by $Q = XW_Q,\ K = XW_K,\ V = XW_V$, where the projections W_Q, W_K, and W_V are all $d_k \times d$ parameter matrices. The energy scores E between any queries and keys are computed as [1]

$$E = QK^\top, \tag{1}$$

where E is an $N \times N$ weight matrix, on which a softmax function is applied to obtain the weights of the values. The output is computed as a weighted sum of the values as

$$Z = \text{Attention}\,(Q, K, V) = \text{Softmax}\,(E)\,V. \tag{2}$$

3.2. Self-attention network for image captioning

Figure 1 shows self-attention network (SAN), which is our baseline architecture for image captioning. Similar to Transformer, the model consists of an image encoder and a caption decoder, both of which are composed of a stack of L layers. Each layer consists of one (for the encoder layer) or two (for the decoder layer) multi-head attention (MHA) sub-layers followed by a feed-forward network (FFN). The MHA sub-layer contains h parallel "heads" with each head corresponding to an independent scaled dot-product attention function. Besides, a residual connection and layer normalization are used between all the sub-layers.

The inputs to the encoder are the region-based visual features extracted from Faster-RCNN [27] object detector. Each input element corresponds to an object in the image. Before feeding the input vectors into the encoder, they are first passed through a dense layer followed by a ReLU layer to adapt their dimension to be consistent with the encoder.

[1] $QK^T/\sqrt{d}$, the scaling factor $\sqrt{d}$ is omitted for simplicity.

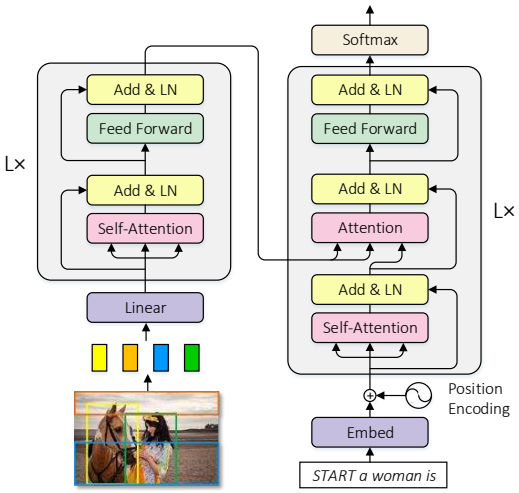

Figure 1. Architecture of the self-attention network (SAN) for image captioning.

The decoder takes the attended visual features and the embeddings of the previous words to predict the next word recursively. Following Transformer, we add sinusoidal "positional encodings" to the inputs at the bottoms of the decoder. Because the regions in the image don't have a natural order like sequences, no position information is added in the encoder side.

4. Approach

4.1. Normalized SA (NSA)

This section introduces a reparameterization of self-attention that takes advantage of normalization method for improved training.

We first review the formulation of Batch Normalization (BN). Consider feeding an input mini-batch x into a feed-forward layer $y = F(x, \Theta)$, where F is an arbitrary transformation, and Θ is the parameter to be learned. The internal covariate shift happens when the distribution of x shifts during training. To reduce internal covariate shift, BN normalizes each channel of x using the mean and variance accumulated over the same channel in the whole mini-batch.

We then take a closer look at the attention weight in Eqn. 2:

$$\begin{aligned} S &= \text{Softmax}(QK^\top) \\ &= \text{Softmax}((XW_Q)\cdot(W_K^\top X^\top)). \end{aligned} \tag{3}$$

It can be considered as an input instance $X \in \mathbb{R}^{N\times d_k}$ first goes through a $d_k \times d$ linear layer parameterized by W_Q to obtain $Q = XW_Q \in \mathbb{R}^{N\times d}$, which is then further fed into a $d \times N$ linear layer parameterized by $K^\top = W_K^\top X^\top$ followed by a Softmax activation to output N probabilities over the keys. Thus, we can re-formulate Eqn. 3 as a fully-

connected layer F followed by a Softmax activation:

$$\begin{aligned} S &= \mathrm{Softmax}(F(Q, \Theta)), \\ Q &= XW^Q, \quad \Theta = K^\top = W_K^\top X^\top. \end{aligned} \tag{4}$$

Note that the parameter Θ is *dynamically* calculated based on X. From this perspective, SA can be susceptible to the internal covariate shift problem just as in a standard feed-forward network. That is, when the distribution of input Q shifts due to the change in network parameters during training, the layer parameter Θ needs to continuously adapt to the new input distribution. Consequently, SA may not be learned effectively.

Therefore, to eliminate the internal covariate shift, it is advantageous for the distribution of Q to remain fixed over time. Then Θ does not have to readjust to compensate for the change in the distribution of Q. This can be accomplished by performing normalization on Q by

$$\hat{Q} = \mathrm{Norm}(Q). \tag{5}$$

We now consider the implementation of Norm. BN is not directly suitable for Norm because instead of using a shared layer parameters for all examples in the dataset, the layer parameter $\Theta = W_K^\top X^\top$ is dynamically computed with the *instance-specific* X. Therefore, it is more desirable to perform normalization, Norm, for every single instance independently.

Let $x \in \mathbb{R}^{B\times T\times C}$ and x_{btc} denote the $btc-$th element of x, where b is the sample index, c is the channel index, and t is the index of the additional spatial dimension. We implement Norm as normalizing each instance in the mini-batch independently using per-channel feature statistics:

$$\begin{aligned} \hat{x}_{btc} &= \frac{x_{btc} - \mu_{bc}}{\sqrt{\sigma_{bc}^2 + \epsilon}}, \\ \mu_{bc} = \frac{1}{T}\sum_{t=1}^{T} x_{btc}, \quad \sigma_{bc}^2 &= \frac{1}{T}\sum_{t=1}^{T} \left(x_{btc} - \mu_{bc}\right)^2. \end{aligned} \tag{6}$$

The above normalization method is exactly the Instance Normalization (IN) in the 1D case. Subtracting the mean from the queries could be considered as highlighting the differences among the queries and encourage them to query information from distinctive aspects.

We represent the normalization operation in Eqn. 6 as $\hat{x} = \mathrm{IN}(x)$. Finally, we derive our normalized self-attention that reparameterizes the self-attention as

$$\hat{Q} = \mathrm{IN}(Q), \quad Z = \mathrm{Softmax}(\hat{Q}K^\top)V. \tag{7}$$

Similar to BN and IN, it is optional to further apply the channel-wise affine transformation $\tilde{x}_{btc} = \hat{x}_{btc}\gamma_c + \beta_c$ in Norm, where $\gamma, \beta \in \mathbb{R}^C$ are learnable scale and shift parameters. But we empirically found it not necessary in our experiments. It is also optional to normalize K with $\hat{K} = \mathrm{IN}(K)$. This is equivalent to normalizing the dynamic parameters Θ, which, however, may limit the capacity of SA.

Relation to prior works. Our normalization method differs from Layer Normalization (LN) in that LN normalizes along all channels of each individual element, while our method normalizes along each channel of all input elements in an instance. As for IN, it is typically used in 2D CNNs, *e.g.* on style transfer task. To our knowledge, IN has not been successfully used for language generation tasks, in particular for SAN.

4.2. Geometry-Aware SA (GSA)

The inherent geometric structure among the input objects is beneficial for reasoning about the visual information, which, however, is not modeled in the vanilla Transformer. Therefore, we propose GSA that improves the SA module by taking into account the pairwise geometry relationships and the content information of objects.

Denote the relative geometry features between two objects i and j as $\mathbf{f}_{ij}^g$, which is a 4-dimensional vector of the relative position and size of the bounding boxes:

$$\left(\log(\frac{|x_i - x_j|}{w_i}), \log(\frac{|y_i - y_j|}{h_i}), \log(\frac{w_i}{w_j}), \log(\frac{h_i}{h_j})\right)^T, \tag{8}$$

where $(x_i, y_i), w_i, h_i$ are the center coordinate, width, and height of box i, respectively.

We project $\mathbf{f}_{ij}^g$ to a high-dimensional representation G_{ij} with a fully-connected (FC) layer followed by a ReLU activation as

$$G_{ij} = \mathrm{ReLU}\left(\mathrm{FC}\left(\mathbf{f}_{ij}^g\right)\right), \tag{9}$$

where $G \in \mathbb{R}^{N\times N\times d_g}$.

We then modify the energy score in Eq. 1 to include the effect of G as

$$E = QK^\top + \phi(Q', K', G), \tag{10}$$

where ϕ is the geometric attention function, which outputs a score matrix of shape $N \times N$, and $Q', K' \in \mathbb{R}^{N\times d_g}$ are geometric queries and keys that are computed in the same way as Q, K, *i.e.* by projecting the input X. In the above equation, the first term is related to the queries and keys, namely *content-based weight*. The second term represents the *geometric bias*, which involves the geometry relations and the contents of Q' and K'.

We now discuss three choices of ϕ, which can be either used individually or combined.

Content-independent geometric bias. The geometry relation G_{ij} conveys useful information for understanding the

relationships between two objects, *e.g.* object i and j have "comparable sizes" and object i is "next to" object j. Thus, we directly project G_{ij} to a scalar score by

$$\phi^1_{ij} = \mathrm{ReLU}(w_g^\top G_{ij}), \tag{11}$$

where w_g is the parameter to be learned. The ReLU nonlinearity acts as a zero trimming operation so that only the relations between objects with certain geometric relationships are considered.

The relation network [16] presented recently for object detection is a special case of the content-independent geometric bias. Different from the above formulation, it fuses the content-independent geometric bias and the original attention weights by multiplication and use sinusoidal embedding of the geometry feature.

Query-dependent geometric bias. The above "content-independent" variant assumes a *static* geometric bias, *i.e.* the same geometric bias is applied to all the query-key pairs in an SA layer. However, the geometric biases are more often different, depending on what the associated query object is. For example, for the queries, "sea" and "ball", their scale difference are often huge in the image, and thus their sensitivities to the same change of a key's distance/position vary widely. Therefore, the geometric biases of the two queries should be adapted to match their content. To this end, we decide to *dynamically* compute the geometric bias for different queries:

$$\phi^2_{ij} = {Q'}_i^\top G_{ij}. \tag{12}$$

Here we use dot-product to match ${Q'}_i$ with G_{ij} since it is more computation and memory efficient than using the Concatenation-FC operation.

Key-dependent geometric bias. Similar to the query-dependent variant, geometric bias can also be associated with the content of the keys, computed as

$$\phi^3_{ij} = {K'}_j^\top G_{ij}. \tag{13}$$

4.3. Applying NSA and GSA modules to SAN

We first combine both NSA and GSA by replacing Q in Eqn. 10 with the normalized one, $\hat{Q}$. We then use this module to replace the vanilla SA modules in the encoder of SAN, which results in our full model, namely Normalized and Geometry-aware Self-Attention Network (**NG-SAN**). NSA is not applied in the decoder of SAN because the decoder is autoregressive and has variable-length inputs. This is undesirable for IN because the mean and variance statistics are meaningless when the sequence length is 1.

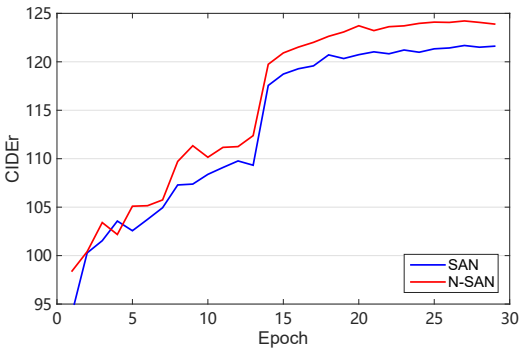

Figure 2. Changes of CIDEr scores during training.

5. Experiments on Image Captioning

5.1. Experimental setup

MS-COCO dataset [22]. It is the most popular benchmark for image captioning. We use the 'Karpathy' splits that have been used extensively for reporting results in prior works. This split contains 113,287 training images with 5 captions each, and 5k images for validation and test splits, respectively. We follow standard practice [35] to preprocess the text, resulting in a final vocabulary of 9,487 words. We use the region-based image features provided by Bottom-Up [2] for training.

Evaluation metrics. We use the standard automatic evaluation metrics to evaluate the quality of image captions, including BLEU-1/2/3/4 [26], METEOR [7], ROUGE-L [21], CIDEr [33], and SPICE [1], which are denoted as B@1/2/3/4, M, R, C and S, respectively.

Implementation details. We follow Transformer-*Base* model [32] and [43] to set the model hyper-parameters and train the model. Specifically, the dimensionality of input image features is 2048. The latent dimension in the MHA module is 512, and the number of heads is 8. Inner dimension in the FFN module is 2,048. We apply *dropout* with a probability of 0.1. We use the same number of layers L for the encoder and decoder. For training, we use the Adam optimizer [19] We use a step decay schedule with warm-up for varying the learning rate. The base learning rate is set to $min(t \times 10^{-4}; 3 \times 10^{-4})$, where t is the current epoch number that starts at 1. After 6 epochs, the learning rate is decayed by 1/2 every 3 epochs. All models are first trained for 15 epochs with the cross-entropy loss and then further optimized with CIDEr reward [28] for additional 15 epochs. If not specifically mentioned, by default we set $L = 4$, only normalize the query and do not apply γ, β in NSA, and use the query-dependent variant (ϕ^1) of GSA. Beam search with a beam width of 3 is used during testing stage.

Table 1. Comparisons between N-SAN and SAN using different numbers of self-attention layers L.

#Layers	Model	#params	B@4	M	R	C	S
1	SAN	18.1M	36.8	28.0	57.6	123.4	21.8
	N-SAN	18.1M	**38.2**	**28.6**	**58.2**	**127.2**	**22.2**
2	SAN	25.5M	38.2	28.5	58.3	127.1	22.3
	N-SAN	25.5M	**38.9**	**28.9**	**58.6**	**129.7**	**22.6**
4	SAN	40.2M	38.4	28.6	58.4	128.6	22.6
	N-SAN	40.2M	**39.3**	**29.1**	**58.9**	**130.8**	**23.0**
6	SAN	54.9M	38.6	28.6	58.5	128.8	22.5
	N-SAN	54.9M	**39.3**	**29.2**	**59.1**	**131.1**	**23.0**

Table 2. Comparison of using various normalization methods in NSA.

Approach	B@4	M	R	C	S
SAN	38.4	28.6	58.4	128.6	22.6
LN	38.5	28.6	58.3	128.2	22.5
BN	38.8	28.9	58.7	129.4	22.8
IN	**39.4**	**29.2**	**59.0**	130.7	**23.0**
IN w/o γ, β	39.3	29.1	58.9	**130.8**	**23.0**

5.2. Analysis on NSA

In this section, we examine the effectiveness of NSA module. We replace the SA modules in the encoder of SAN with NSA, resulting in a model named Normalized Self-Attention Network (**N-SAN**).

Number of attention layers. In Table 1 we compare the performance of N-SAN and SAN under the same number of SA layers $L \in \{1, 2, 4, 6\}$. We can see that the model size grows linearly as L increases. Regarding the performance, we have two observations as follows. 1) As L increases, the performance of both SAN and N-SAN gradually improves and reaches the optimal value when $L = 6$. However, the performance gain of increasing L from 4 to 6 is not very significant. Therefore, we use $L = 4$ for later experiments as a compromise between the model's performance and complexity. 2) N-SAN consistently outperforms SAN on all metrics under different L. In Figure 2, we further plot the CIDEr scores of the one-layer SAN and N-SAN models during training, evaluated on the validation split at each epoch. As we can see, the curve of N-SAN is above that of SAN for most of the time.

Different normalization methods. Since we introduced IN into the NSA module for normalization, an intuitive question to ask is whether we can replace IN with other normalization methods. In Table 2 we show the results of using different normalization methods including BN, LN, IN and IN without using the affine transformations (γ and β). We have the following observations. 1) Using LN slightly decreases the performance. We conjecture that is because LN normalizes activations of all channels with the same normalization terms (μ and σ), thus limiting the expression capacity of each channel when calculating attention weights. 2) IN and *IN w/o* γ, β significantly outperform SAN and all the other normalization methods. Meanwhile, the extra affine transformations (γ and β) are not necessary. 3) Applying BN outperforms SAN but is inferior to adopting IN. BN has a similar effect as IN to reduce the internal covariate shift by fixing the distribution of the queries. However, as is described in Sec. 4.1, since the layer parameter Θ in Eqn. 4 depends on instance-specific input, it is more desirable to perform input normalization also on each instance instead of on the whole mini-batch.

Table 3. Comparison of normalizing query and key in N-SAN.

Query	Key	B@4	M	R	C	S
✗	✗	38.4	28.6	58.4	128.6	22.6
✓	✗	39.3	**29.1**	**58.9**	**130.8**	23.0
✗	✓	39.2	29.0	58.8	130.1	22.8
✓	✓	**39.4**	**29.1**	58.8	130.7	**23.1**

Table 4. Comparison of various variants of GSA.

Approach	#params	B@4	M	R	C	S
SAN	40.2M	38.4	28.6	58.4	128.6	22.6
absolute	40.2M	38.3	28.5	58.4	128.4	22.6
content-independent	40.2M	39.2	29.1	58.9	131.0	22.9
key-dependent	41.5M	38.9	29.0	58.8	129.5	22.8
query-dependent	41.5M	**39.3**	**29.2**	**59.0**	**131.4**	**23.0**

What if we normalize the keys in addition to the queries? In Table 3, we compare the variants of Eqn. 7, including normalizing Q alone, K alone, and both Q and K. We have the following observations. 1) Normalizing either of Q and K could increase the performance. 2) The performances of normalizing both Q and K and normalizing Q alone are very similar, and are both significantly higher than that of SAN. 3) Normalizing K alone is inferior to normalizing Q alone. The reason is that normalizing K is equivalent to normalizing Θ in Eqn. 4, which may limit the model capacity of SA.

5.3. Analysis on GSA

In this section, we examine the effectiveness of GSA module. Similar to N-SAN, we replace the SA modules in the encoder of SAN with GSA to obtain a model named Geometry-aware Self-Attention Network (**G-SAN**).

Variants of GSA. In Table 4 we compare various variants of GSA module introduced in Sec. 4.2. "+absolute" denotes adding absolute geometry information of each individual object to their input representations at the bottoms of the encoder. It is obtained by embedding the geometry features, *i.e.* the center coordinates and the width/height of the box, normalized by the width/height of the image, to a

Table 5. Leaderboard of the published state-of-the-art, *single-model* methods on the online MS-COCO test server, where c5 and c40 denote using 5 and 40 references for testing, respectively. CIDEr (C40) is the default sorting metric on the leaderboard.

Model	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE-L		CIDEr-D	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
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
CAVP [23]	80.1	94.9	64.7	88.8	50.0	79.7	37.9	69.0	28.1	37.0	58.2	73.1	121.6	123.8
SGAE [42]	80.6	95.0	65.0	88.9	50.1	79.6	37.8	68.7	28.1	37.0	58.2	73.1	122.7	125.5
VSUA [14]	79.9	94.7	64.3	88.6	49.5	79.3	37.4	68.3	28.2	37.1	57.9	72.8	123.1	125.5
NG-SAN (Ours)	**80.8**	95.0	**65.4**	**89.3**	**50.8**	**80.6**	**38.8**	**70.2**	**29.0**	**38.4**	**58.7**	**74.0**	**126.3**	**128.6**

sinusoidal representation using the same method as the "positional encodings" in [32]. We have the following findings. 1) Adding the absolute geometry information ("absolute") is not beneficial to the performance. That is probably because it is too complex for SA to infer the 2D layout of objects from their absolute geometry information. 2) All the proposed variants of GSA can improve the performance of SAN, showing the advantages of using relative geometry information. 3) "query-dependent" brings the best performance and outperforms the content-independent variant, proving that incorporating the content information of the associated query can help infer a better geometric bias. 4) "key-dependent" is inferior to "query-dependent". That is because when using key-dependent geometric bias, the scores $\phi^3_{ij} = {K'_j}^\top G_{ij}$ condition on *different* keys K'_j, thus the differences in G_{ij} may be overwhelmed by the differences in K'_j when performing softmax on the keys' dimension. In comparision, when using query-dependent geometric bias, the effect of G_{ij} could be highlighted since the scores condition on a *common* query Q'_i when performing softmax. We did not observe further improvement when combing these variants into ϕ in Eq. 10.

5.4. Analysis on the full model (NG-SAN)

We now validate the effectiveness of NG-SAN that takes advantage of both NSA and GSA.

Comparisons with state-of-the-arts. We compare NG-SAN with the state-of-the-art methods, including Up-Down [2], CAVP [23], SGAE [42], VSUA [14], ORT [15], AoANet [17], and MT [43]. All the methods except ORT, AoANet, and MT are based on single- or multi-layer Long Short-Term Memory (LSTM) networks. MT adopts a Transformer-Base architecture, using 6 SA layers for both the encoder and the decoder, and inserts an additional LSTM layer in the decoder. ORT also adopts the Transformer-Base architecture and follows [16] to model the spatial relationship between inputs. AoANet uses SAN as the encoder and LSTM as the decoder.

Table 6 compares the results of each method. We can see that both G-SAN and N-SAN outperform the SAN baseline across all metrics. Moreover, NG-SAN further outperforms G-SAN and N-SAN, demonstrating that GSA and NSA are compatible with each other. NG-SAN significantly outperforms all the other methods, including both LSTM-based and SA-based ones, over all metrics. Particularly, we improve the best CIDEr score from 130.9 to 132.1. Table 5 further reports the performance of the top-performing *single-model* solutions on the official test server. Compared with the published methods, our single model significantly outperforms all the other methods in terms of all evaluation metrics except BLEU-1. In particular, we establish a new state-of-the-art score of 128.6 on CIDEr (C40).

Table 6. Comparisons with state-of-the-art single-model approaches on MS-COCO Karpathy test split.

Model	#params	B@4	M	R	C	S
Up-Down [2]	–	36.3	27.7	56.9	120.1	21.4
CAVP [23]	–	38.6	28.3	58.5	126.3	21.6
SGAE [42]	–	39.0	28.4	58.9	129.1	22.2
VSUA [14]	–	38.4	28.5	58.4	128.6	22.0
ORT [15]	–	38.6	28.7	58.4	128.3	22.6
AoANet [17]	–	38.9	29.2	58.8	129.8	22.4
MT [43]	57.0M	39.8	29.1	59.1	130.9	–
SAN	40.2M	38.4	28.6	58.4	128.6	22.6
N-SAN	40.2M	39.3	29.1	58.9	130.8	23.0
G-SAN	41.5M	39.3	29.2	59.0	131.4	23.0
NG-SAN	41.5M	**39.9**	**29.3**	**59.2**	**132.1**	**23.3**

Complexity. As can be seen in the "#params" column in Table 6, NG-SAN requires very few (about 2k) additional parameters compared with SAN. For NSA, it does not require any parameters, and the computation overhead of the additional normalization process is almost ignorable. While GSA indeed requires some additional parameters, the amount is ignorable. GSA can be efficiently implemented by matrix multiplication and the einstein summation (einsum) operations provided by mainstream deep learning frameworks.

6. Extension: Experiments on Other Tasks

We further investigate the effectiveness and generality of our methods on Video Captioning (VC) [34], Machine Translation (MT) [5], and Visual Question Answer-

Table 7. Video captioning results on VATEX dataset.

Model	B@4	M	R	C
VATEX [38]	28.2	21.7	46.9	45.7
Transformer (Ours)	30.6	22.3	48.4	53.4
+NSA	**31.0**	**22.7**	**49.0**	**57.1**

Table 8. Machine translation results on newstest2014 for WMT 2014 En-De dataset.

Model	BLEU
Transformer-Base [32]	27.30
Transformer-Big [32]	28.40
Transformer-Base (Ours)	27.56
+NSA	**27.92**

ing (VQA) [3] tasks. Since VC and MT are both sequence-to-sequence problems, we directly use Transformer as the baseline models, and we replace the SA modules in their encoder with the proposed NSA module to construct our methods. As for VQA, we use MCAN [44] as the baseline model, which uses a SAN-based network to simultaneously encode image and question information. To build our method for VQA, we replace all the SA modules in MCAN with our GSA modules.

6.1. Video Captioning

We use a recently released large-scale video captioning dataset, VATEX [38]. It contains over 41,250 videos and 412,500 English captions. For a fair comparison with VATEX, we directly use the pre-extracted video features provided by the paper. Specifically, each video is sampled at 25fps and 1,000-dimensional features are extracted from these sampled frames using a pretrained I3D [6] model. Because the dataset is relatively small, we found using one layer in both the encoder and decoder is satisfactory. We use a training configuration the same as that of our image captioning model.

In Table 7, we compare our method with the Transformer baseline and the VATEX model. We see that the performance of Transformer strongly exceeds that of VATEX, which adopts an LSTM-based architecture. Our Transformer+NSA method consistently improves over Transformer on all metrics. Particularly, our method improves the CIDEr score by 3.7 points when compared to Transformer, and significantly improves the CIDEr score by 11.4 points when compared to VATEX baseline.

6.2. Machine Translation

We also evaluate NSA on MT task, for which the Transformer was originally proposed. We trained on the widely-used WMT 2014 English to German (En-De) dataset, which consists of about 4.56 million sentence pairs. The models were validated on newstest-2013 and tested on newstest-2014 with BLEU. We use the well-known Transformer-Base [32] variant of Transformer as the baseline model, which has 6 layers in both the encoder and decoder. Specifically, we follow the implementation of the `fairseq-py` [25] toolkit.

As shown in Table 8, Compared to Transformer-Base model, NSA increases the BLEU score by 0.36 points without adding any parameters.

Table 9. Visual question answering accuracies on the VQA-v2 dataset to compare with the state-of-the-art single-model methods.

Model	test-dev	test-std
MLIN [12]	70.18	70.28
DFAF [11]	70.22	70.34
MCAN [44]	70.63	70.90
MCAN (Ours)	70.54	70.83
+GSA	**70.76**	**71.28**

6.3. Visual Question Answering

We conduct experiments on the most commonly used VQA benchmark, VQA-v2 [3]. It contains human-annotated question-answer pairs relating to the images from the MS-COCO dataset, with 3 questions per image and 10 answers per question. We strictly follow MCAN [44] to implement our models. Specifically, images are represented with region features extracted from Faster R-CNN object detector and the input questions are transformed with GloVe word embeddings and an LSTM network.

Table 9 shows the overall accuracies of our methods and the current state-of-the-art models on the online test-dev and test-std splits. GSA boosts the test-std accuracy of MCAN from 70.83 to 71.28.

7. Conclusion

We proposed two improvements to the self-attention (SA) mechanism, *i.e.* a Normalized Self-Attention (NSA) to reduce the internal covariate shift problem inside SA, and a class of Geometry-aware Self-Attention (GSA) that explicitly and dynamically computes the geometric bias between objects to benefit image understanding. We have conducted extensive experiments on MS-COCO image captioning dataset to validate the effectiveness of NSA, GSA, and their combination. We further show the significance and generality of our methods on video captioning, machine translation, and visual question answering tasks. On all tasks, simply replacing the vanilla SA module with our proposed methods provides solid improvements over strong baselines.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No.61922086 and No.61872366) and Beijing Natural Science Foundation (No.4192059).

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. pages 382–398, 2016. 5

[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 vqa. *arXiv preprint arXiv:1707.07998*, 2017. 2, 5, 7

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 8

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 1, 2

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 7

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 8

[7] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *The Workshop on Statistical Machine Translation*, pages 376–380, 2014. 5

[8] 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. 1

[9] Jun Fu, Jing Liu, Yong Li, Yongjun Bao, Weipeng Yan, Zhiwei Fang, and Hanqing Lu. Contextual deconvolution network for semantic segmentation. *Pattern Recognition*, page 107152, 2020. 1

[10] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019. 1

[11] Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven C. H. Hoi, Xiaogang Wang, and Hongsheng Li. Dynamic fusion with intra- and inter-modality attention flow for visual question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 8

[12] Peng Gao, Haoxuan You, Zhanpeng Zhang, Xiaogang Wang, and Hongsheng Li. Multi-modality latent interaction network for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5825–5835, 2019. 8

[13] Longteng Guo, Jing Liu, Shichen Lu, and Hanqing Lu. Show, tell and polish: Ruminant decoding for image captioning. *IEEE Transactions on Multimedia*, 2019. 2

[14] Longteng Guo, Jing Liu, Jinhui Tang, Jiangwei Li, Wei Luo, and Lu Hanqing. Aligning linguistic words and visual semantic units for image captioning. In *ACM MM*, 2019. 7

[15] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: Transforming objects into words. *arXiv preprint arXiv:1906.05963*, 2019. 3, 7

[16] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018. 1, 3, 5, 7

[17] 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. 7

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Computer Science*, 2015. 1, 2

[19] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[20] Ryan Kiros, Ruslan Salakhutdinov, and Rich Zemel. Multimodal neural language models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 595–603, 2014. 1

[21] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004. 5

[22] Tsungyi Lin, Michael Maire, Serge J Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C Lawrence Zitnick. Microsoft coco: Common objects in context. *european conference on computer vision*, pages 740–755, 2014. 5

[23] Daqing Liu, Zheng-Jun Zha, Hanwang Zhang, Yongdong Zhang, and Feng Wu. Context-aware visual policy network for sequence-level image captioning. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 1416–1424. ACM, 2018. 7

[24] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019. 1

[25] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019. 8

[26] Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. Bleu: a method for automatic evaluation of machine translation. *Meeting on Association for Computational Linguistics*, pages 311–318, 2002. 5

[27] 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. 3

[28] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. *computer vision and pattern recognition*, 2017. 2, 5

[29] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018. 3

[30] Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. Deep semantic role labeling with self-attention. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 1

[31] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 2

[32] 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 6000–6010, 2017. 1, 5, 7, 8

[33] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. *Computer Science*, pages 4566–4575, 2015. 5

[34] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015. 7

[35] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3156–3164. IEEE, 2015. 5

[36] 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, 2017. 2

[37] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 1

[38] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. *arXiv preprint arXiv:1904.03493*, 2019. 8

[39] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 2

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

[41] Baosong Yang, Jian Li, Derek F Wong, Lidia S Chao, Xing Wang, and Zhaopeng Tu. Context-aware self-attention networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 387–394, 2019. 1

[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. 7

[43] Jun Yu, Jing Li, Zhou Yu, and Qingming Huang. Multimodal transformer with multi-view visual representation for image captioning. *arXiv preprint arXiv:1905.07841*, 2019. 1, 2, 5, 7

[44] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 8

[45] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018. 1

[46] Xinxin Zhu, Lixiang Li, Jing Liu, Haipeng Peng, and Xinxin Niu. Captioning transformer with stacked attention modules. *Applied Sciences*, 8(5):739, 2018. 1, 2