Motion Guided Spatial Attention for Video Captioning

Shaoxiang Chen, Yu-Gang Jiang*
Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University
Shanghai Institute of Intelligent Electronics & Systems
{sxchen13, ygj}@fudan.edu.cn

Abstract

Sequence-to-sequence models incorporated with attention mechanism have shown promising improvements on video captioning. While there is rich information both inside and between frames, spatial attention is rarely explored and motion information is usually handled by 3D-CNNs as just another modality for fusion. On the other hand, researches about human perception suggest that apparent motion can attract attention. Motivated by this, we aim to learn spatial attention on video frames under the guidance of motion information for caption generation. We present a novel video captioning framework by utilizing Motion Guided Spatial Attention (MGSA). The proposed MGSA exploits the motion between video frames by learning spatial attention from stacked optical flow images with a custom CNN. To further relate the spatial attention maps of video frames, we designed a Gated Attention Recurrent Unit (GARU) to adaptively incorporate previous attention maps. The whole framework can be trained in an end-to-end manner. We evaluate our approach on two benchmark datasets, MSVD and MSR-VTT. The experiments show that our designed model can generate better video representation and state of the art results are obtained under popular evaluation metrics such as BLEU@4, CIDEr, and METEOR.

1 Introduction

Automatically describing the content of a video using natural language, i.e., video captioning, is a challenging task in computer vision. Lots of practical applications such as auxiliary aid for visually impaired people, human computer interaction, and video retrieval can benefit from video captioning, thus it has drawn great research attention. In general, video captioning systems can be roughly divided into two components: video representation and sentence generation. Traditional approaches (Krishnamoorthy et al. 2013; Thomason et al. 2014; Guadarrama et al. 2013) used various visual classifiers/trackers to detect visual concepts and then generate sentences with predefined language templates. For video representation, these approaches rely on handcrafted features which do not generalize well and can not be trained in an end-to-end manner. With the rapid development of deep learning, two major changes have been made to video captioning systems: convolutional neural networks (CNNs) for video representation and recurrent neural networks (RNNs) for sequence modeling. Earlier researchers (Venugopalan et al. 2015a; 2015b; Donahue et al. 2015) directly extracted global feature (i.e., a single vector to represent one frame) of video frames from a pre-trained CNN and fed to RNNs (LSTM (Hochreiter and Schmidhuber 1997), GRU (Chung et al. 2014), etc.) for sentence generation. While these plain sequence-to-sequence approaches can achieve significant improvements over traditional methods, they still suffer from loss of both spatial and temporal information in videos. Later works (Yao et al. 2015; Baraldi, Grana, and Cucchiara 2017; Pan et al. 2016) tried to exploit the temporal structure of videos by adaptively assigning weights to video frames at every word generation step, which is known as temporal attention. But in these works, video frames are still represented by global feature vectors extracted from CNNs. Thus, the rich visual contents in video frames are not fully exploited. Spatial attention is widely adopted in image captioning. (Xu et al. 2015) proposed to learn a set of attention weights for image regions[1] to represent their relevance to the generated words, so that the regional features can be better combined. Motivated by

Figure 1: Best viewed in color. Example video frames shown with optical flow in the bottom row. The frame in red box is the sampled key frame. The flow images in green and yellow boxes are the horizontal and vertical component of the optical flow computed around the key frame, respectively.

*Corresponding author.

[1] For convenience, we refer the 2D attention weight matrix as "attention map" hereinafter.

the success of spatial attention in image captioning, recent works (Yang, Han, and Wang 2017; Li, Zhao, and Lu 2017) in video captioning have also adopted spatial attention, in which the attention weights are learned from scratch.

Videos by nature have clear indication of where the actions/events are happening, that is the motion between video frames. Prior researches of human perception (Itti and Baldi 2005; Howard and Holcombe 2010) have shown that human attentions are also more likely to be drawn to the apparently changing regions of a video. Motivated by this, we propose to guide the spatial attention by optical flow, which can capture the pattern of apparent motion between consecutive video frames. An example is shown in Figure 1: the motion captured by optical flow in both horizontal and vertical direction is a strong indication of the action and the related objects in the video. Besides, the actions in videos is related across time. So we also consider the temporal relation between attention maps, and propose a GRU-like Gated Attention Recurrent Unit (GARU) to model this relationship.

Our contributions of this work are as follows: (1) We present a novel video captioning framework named Motion Guided Spatial Attention (MGSA), which utilizes optical flow to guide spatial attention. To the best of our knowledge, this is the first work that incorporates optical flow for attention guidance in video captioning. (2) We show that introducing recurrent relations between consecutive spatial attention maps can give a boost to captioning performance and designed a recurrent unit called Gated Attention Recurrent Unit (GARU) for this purpose. (3) We achieve the current state of the art results on two large-scale datasets: MSVD and MSR-VTT. We also investigate spatial attention maps learned by our MGSA model and show that it can better locate regions of interest.

2 Related Works

Temporal Attention. One of the first works that have adopted CNN and RNN for video captioning is (Venugopalan et al. 2015b), in which video representation is obtained by mean-pooling CNN features extracted from a sequence of sampled video frames and then fed it to LSTM for caption generation. This approach actually treated video as an image and ignored the temporal structure of videos. Thus, following works tries to encode the videos while exploiting their structures. S2VT (Venugopalan et al. 2015a) first encodes the video feature sequence with two layers of LSTM and then the language generation (decoding) is conditioned on the final encoding state. The LSTMs in these two stages share the same parameters. This kind of encoding-decoding approach have been successfully applied to neural machine translation (Sutskever, Vinyals, and Le 2014). In (Yao et al. 2015), the authors exploit the temporal structure of a video by introducing soft-attention mechanism in the decoding stage, which assigns weights to video frames calculated from the decoder state and video features. (Baraldi, Grana, and Cucchiara 2017) further propose to model the hierarchical structure of videos by detecting the shot boundaries while generating captions. (Zhu, Xu, and Yang 2017) also propose Multirate Gated Recurrent Unit to encode frames of a video clip with different intervals, so that the model can be capable of dealing with motion speed variance.

Spatial Attention. Our focus in this paper is to exploit the spatial information of video frames. Due to the large amount of video data, spatial information has been overlooked in video captioning due to the high computational cost. But in image captioning, spatial information is widely utilized through attention. In (Xu et al. 2015), two forms of attention mechanism are proposed for image captioning. One is stochastic hard attention, which selects a single image region according to a multinoulli distribution and requires Monte Carlo sampling to train. The other is a differentiable approximation of the former, which computes weights for all the image regions and then a weighted sum over all the regional features. Although the hard attention was shown to give better performance, later researchers have preferred the soft approximation for its ease of training. (Liu et al. 2017) show that if supervision for attention is available during training image captioning models, the trained models can better locate regions that are relevant to the generated captions. However, due to the vast amount of video data, there are no such fine-grained spatial annotation in existing video captioning datasets. Recently, there are works that try to incorporate spatial attention in video captioning. (Li, Zhao, and Lu 2017) apply region-level (spatial) soft attention to every video frame and then frame-level (temporal) attention to all the frames to obtain a multi-level attention model for video captioning. (Yang, Han, and Wang 2017) propose to generate spatial attention under the guidance of global feature, which is the mean-pooled regional features. They also designed a Dual Memory Recurrent Model (DMRM) to incorporate the information of previously encoded global and regional features. The MAM-RNN (Li, Zhao, and Lu 2017) applies spatial attention in the encoding stage followed by temporal attention in the decoding stage. The spatial attention maps are directly propagated during encoding. (Li, Zhao, and Lu 2017) and (Yang, Han, and Wang 2017) are most related to our work, however, their spatial attentions are generated from the regional features and recurrent states of the RNNs, without direct guidance.

Optical Flow for Visual Recognition. We propose to generate spatial attention from a more explicit clue: optical flows. Our idea is motivated by the success of optical flow in video action recognition. Recent works (Simonyan and Zisserman 2014; Wang et al. 2016) have shown that CNNs trained on multi-frame dense optical flow is able to achieve good action recognition performance in spite of limited amount of training data. Although the C3D network (Tran et al. 2015), which operates on consecutive RGB frames has also been proven to be successful for recognizing action in videos, it requires training on large-scale datasets. As a result, video captioning approaches (Hori et al. 2017; Xu et al. 2017; Chen et al. 2017) have always used motion information from C3D as just another modality for fusion only. (Venugopalan et al. 2015a) tries to feed optical flow to a CNN pre-trained on UCF101 video dataset for feature extraction and then for multi-modal fusion. None of these works have used optical flow as guidance for visual attention.

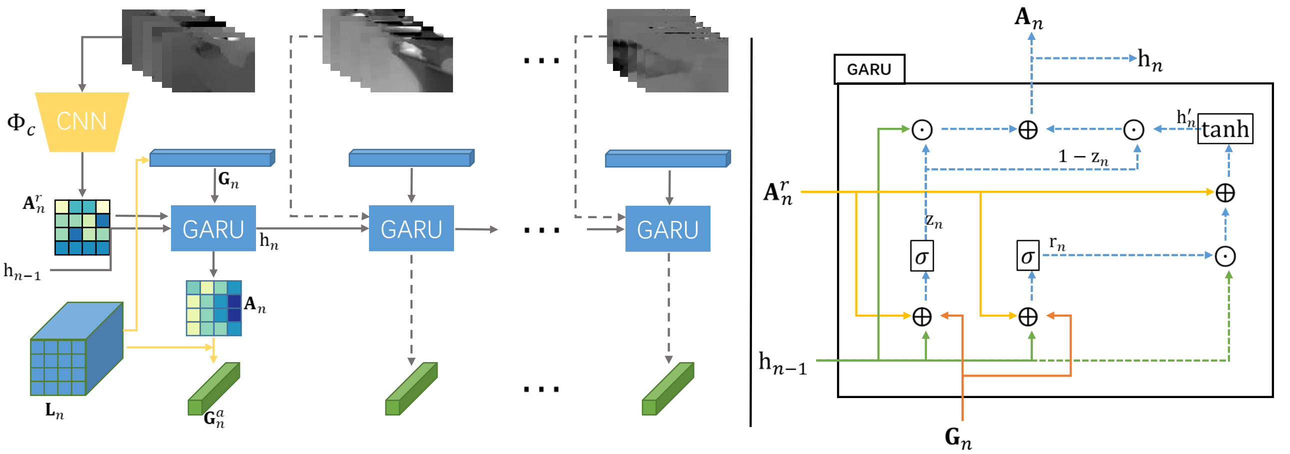

Figure 2: The architecture of our proposed MGSA for video feature encoding. Left: Overview of MGSA. Dashed connections have some components omitted for clarity. $\mathbf{A}_n^r$ is the rough attention map produced by the CNN that operates on stacked optical flow images. $\mathbf{G}_n$ is obtained by spatially mean-pooling $\mathbf{L}_n$. $\mathbf{G}_n^a$ is the weighted sum of $\mathbf{L}_n$ with $\mathbf{A}_n$ as the weights. Right: The details of GARU. Solid lines stands for weighted connections, i.e., the inputs are multiplied by a weight matrix, and dashed lines stands for direct connections. $\odot$, $\oplus$ and σ stands for element-wise multiplication, addition and sigmoid function, respectively.

3 Approach

Different from many existing works, which focus on fully exploiting the multimodal information in videos, we aim at designing a video captioning model that effectively attends to spatial regions-of-interest under the guidance of motion information in videos. We use dense optical flow to explicitly capture the motion between consecutive frames. First, stacked dense optical flow extracted around sampled key frame is fed to a CNN to compute a rough spatial attention map. To utilize the relation between attention maps, we then designed a gated attention recurrent unit (GARU) to incorporate attention information from previous frames. The GARU outputs refined attention map, and the regional features are aggregated into a discriminative global representation with the help of the attention map.

Our approach is also in a sequence-to-sequence manner, and can be divided into encoding and decoding stages. We first give an overview of our approach in Section 3.1. We then introduce the encoding stage, including our proposed MGSA and GARU in Section 3.2. Finally we describe the decoding stage in Section 3.3.

3.1 Overview

Given a video, we uniformly sample N frames as the key frames. We compute horizontal and vertical components of optical flow for M frames centered at each key frame to capture the short-term motion information. These optical flows are stacked as a tensor[2] $\mathbf{F}$ with shape $(N, H_I, W_I, 2M)$, where H_I and W_I are the height and width of the frames, respectively. The video frames are fed to a pre-trained CNN to extract regional features. We generally take the activations of the last convolutional layer of the CNN. The resulting feature is also a tensor $\mathbf{L}$ with shape (N, H, W, D), where D is the number of output channels of the pre-trained CNN, and H and W are the spatial dimensions. As figure 2 shows, the stacked flows are fed to a custom CNN denoted by Φ_c and the output will be a tensor $\mathbf{A}^r$ with shape $(N, H, W, 1)$, or (N, H, W) if we squeeze the last dimension. Each sub-tensor of $\mathbf{A}^r$, which is denoted as $\mathbf{A}_n^r$ and has shape of (H, W), is the rough spatial attention map of each corresponding frame. The rough attention map itself can also be used to aggregate the regional features. But we wish to refined it by considering the interrelationship of the attention maps across time. Thus the rough attention maps are then sequentially processed by our designed Gated Attention Recurrent Unit (GARU) to incorporate previous attention maps. We also feed global feature vector $\mathbf{G}_n$, which is the spatially mean-pooled $\mathbf{L}_n$ with length D to the GARU to provide high-level information of the key frame. The refined attention maps $\mathbf{A}$ is applied to weigh $\mathbf{L}$, obtaining the attended global representation of the corresponding frames, denoted as $\mathbf{G}^a$. The above encoding stage can be formalized as below:

$$\mathbf{A}^r = \Phi_c(\mathbf{F}), \tag{1}$$

$$h_n, \mathbf{A}_n = \mathrm{GARU}(h_{n-1}, \mathbf{A}_n^r, \mathbf{G}_n), \tag{2}$$

$$\mathbf{G}_n^a = f_{att}(\mathbf{L}_n, \mathbf{A}_n), \tag{3}$$

where $n = 1, 2, ..., N$ and f_{att} is the attention operation along the spatial dimensions of $\mathbf{L}_n$ with $\mathbf{A}_n$, the attention map refined by GARU as the weights. $\mathbf{h}_n$ is the hidden state of GARU. $\mathbf{G}^a$ is the concatenation of $\mathbf{G}_1^a, \mathbf{G}_2^a, ..., \mathbf{G}_N^a$.

As Figure 3 shows, the decoding stage of our approach also use a stack of two LSTMs as in state of the art works (Xu et al. 2017; Song et al. 2017; Venugopalan et al. 2015a). The input word sequence are encoded as one-hot

[2]Following the programming convention in Tensorflow, we use tensor to denote multidimensional arrays in this paper. A tensor $\mathbf{T}$ with shape $(N_0, N_1, ..., N_{D-1})$ is an array of rank D, and its axis i has length N_i. We use subscript to index sub-tensors, e.g., $\mathbf{T}_i$ stands for $\mathbf{T}$'s i-th sub-tensor in the first axis, which has shape $(N_1, ..., N_{D-1})$.

vectors $(x_1, x_2, ..., x_T)$, where T is the length of the sentence. We then embed the one-hot vectors to W-dimensional vectors, and the embedding is trained jointly with the model. At each time step t, the decoder is trained to predict the t-th word conditioned on the previous $t-1$ words and $\mathbf{G}^a$. The output of the decoder is thus a conditional probability distribution:

$$p(x_t|x_1, x_2, ..., x_{t-1}, \mathbf{G}^a; \theta), \tag{4}$$

where θ stands for all the trainable model parameters. We define the objective function as negative log-likelihood:

$$Loss = -\sum_{t=1}^{T} \log p(x_t|x_1, x_2, ..., x_{t-1}, \mathbf{G}^a; \theta). \tag{5}$$

The model learns to minimize the negative log-likelihood by stochastic gradient descent during training.

3.2 Motion-Guided Spatial Attention

In video action recognition (Simonyan and Zisserman 2014; Wang et al. 2016), the CNNs that take optical flow as input are designed to have similar or identical architecture as the CNNs that process RGB frames, which are usually deep. Unlike recognizing actions and events, learning spatial attention map from optical flow images is an easier task. Since optical flow already explicitly captures the motion, which is a good hint for where the model show attend to. Thus for this task, we designed a 5-layer CNN, i.e., the Φ_c in Eq. (1). We interleave max-pooling with convolution to reduce the spatial resolution and increase receptive field.

We have also experimented with deeper CNN architectures and found that the increase of performance is not significant. Thus we chose a lightweight CNN for better training efficiency.

The rough spatial attention map $\mathbf{A}^r$ produced by Φ_c is solely generated from short-term motion information around a key frame. While this attention map is already applicable, we wish to take one step further. Since an action in video is continuous in time, the attention maps of nearby key frames should also be related. Based on this observation, we design a GRU-like gated recurrent unit named GARU to incorporate previous spatial attention map when generating the current one.

As shown in Figure 2, the proposed GARU take the rough attention map generated by the flow CNN, $\mathbf{A}_n^r$ and the global feature of the key frame, $\mathbf{G}_n$. At each time step, GARU produces the refined attention map $\mathbf{A}_n$, and propagates the state h_n to the next time step. The detailed computation of GARU is as follows:

$$\begin{aligned} r_n &= \sigma\big(W^{(r)}\mathbf{A}_n^r + U^{(r)}h_{n-1} + V^{(r)}\mathbf{G}_n\big), \\ z_n &= \sigma\big(W^{(z)}\mathbf{A}_n^r + U^{(z)}h_{n-1} + V^{(z)}\mathbf{G}_n\big), \\ h'_n &= \tanh\big(W^{(h)}\mathbf{A}_n^r + r_n \odot U^{(h)}h_{n-1}\big), \\ h_n &= h_{n-1} \odot z_n + h'_n \odot (1 - z_n), \end{aligned} \tag{6}$$

where σ stands for the sigmoid function and $\odot$ stands for element-wise multiplication. U, V and W with different superscripts are all trainable parameters. Note that h_{n-1} is equivalent to $\mathbf{A}_{n-1}$. r_n is the reset gate that controls how much information from previous hidden state we should keep. z_n is the element-wise update gate that directly controls how much of the previous attention map should be kept. The initial state of GARU is simply set to zero.

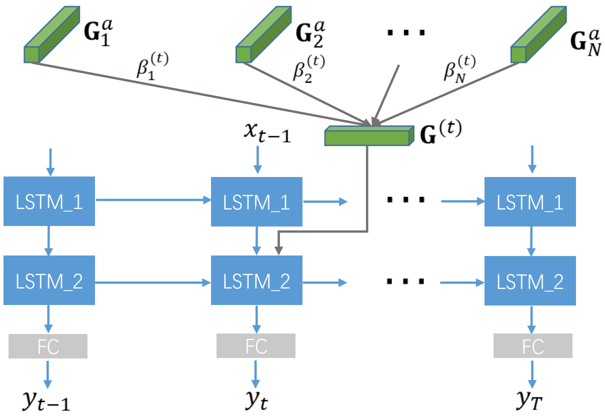

Figure 3: The decoder structure used in our approach.

The attention weights of every region is obtained by applying softmax function to $\mathbf{A}_n$ along the spatial dimensions:

$$\alpha_{nhw} = \frac{exp(\mathbf{A}_{nhw})}{\sum_{h=1}^{H}\sum_{w=1}^{W} exp(\mathbf{A}_{nhw})}. \tag{7}$$

Then the attended global feature by a weighted sum over the regional features:

$$\mathbf{G}_n^a = \sum_{h=1}^{H}\sum_{w=1}^{W} \alpha_{nhw}\mathbf{L}_{nhw}. \tag{8}$$

Eq. 7 and Eq. 8 also complement the details of the f_{att} operation in Eq. 3. The generated global feature sequence $(\mathbf{G}_1^a, \mathbf{G}_2^a, ..., \mathbf{G}_N^a)$ is the encoded feature representation for each frame and is fed to the decoder for sentence generation.

3.3 The Decoding Stage

In previous methods (Pan et al. 2016; Yang, Han, and Wang 2017), temporal attention in decoding stage is shown to improve the captioning performance. As shown in Figure 3, we further apply temporal attention to the spatially-attended feature sequence $(\mathbf{G}_1^a, \mathbf{G}_2^a, ..., \mathbf{G}_N^a)$ at each word generation step to further obtain a temporally attended feature:

$$\mathbf{G}^{(t)} = \sum_{n=1}^{N} \beta_n^{(t)}\mathbf{G}_n^a, \tag{9}$$

where $\beta_1^{(t)}, \beta_2^{(t)}, ..., \beta_N^{(t)}$ are the attention weights dynamically decided at t-th time step. The computation of $\beta_n^{(t)}$ relies on the recurrent states of decoder LSTMs, which stores the information of previously seen words and features, and is computed as

$$\begin{aligned} h_t^{(1)}, c_t^{(1)} &= \text{LSTM\_1}\big(x_{t-1}, (h_{t-1}^{(1)}, c_{t-1}^{(1)})\big), \\ h_t^{(2)}, c_t^{(2)} &= \text{LSTM\_2}\big([h_t^{(1)}, \mathbf{G}^{(t)}], (h_{t-1}^{(2)}, c_{t-1}^{(2)})\big), \end{aligned} \tag{10}$$

where $h_t^{(l)}, c_t^{(l)}$ are the hidden and cell state of the l-th LSTM at time step t, and $[\cdot]$ stands for tensor concatenation. $\beta_n^{(t)}$ is then computed as

$$\begin{aligned} e_n^{(t)} &= W_A^T \tanh(W[h_t^{(1)}, h_t^{(2)}] + U\mathbf{G}_n^a + b), \\ \beta_n^{(t)} &= \frac{exp(e_n^{(t)})}{\sum_{n=1}^{N} exp(e_n^{(t)})}, \end{aligned} \tag{11}$$

where W_A, W, U and b are trainable parameters that are shared across all time steps. We use a single output layer that maps the output of LSTM_2 into a distribution over the vocabulary:

$$\begin{aligned} y_t &= softmax(W^{(o)} h_t^{(2)}) \\ &= p(x_t | x_1, x_2, ..., x_{t-1}, \mathbf{G}^a; \theta). \end{aligned} \tag{12}$$

During testing time, the ground truth sentence $(x_1, x_2, ..., x_T)$ is not given. Thus the input to the LSTM_1 is the previous word prediction of the model:

$$x_t = \arg\max_{w \in \mathcal{V}} p(w | x_1, x_2, ..., x_{t-1}, \mathbf{G}^a; \theta), \tag{13}$$

where $\mathcal{V}$ is the vocabulary. The whole framework can be trained in an end-to-end manner.

4 Experiments

4.1 Datasets

MSVD. The MSVD dataset (Chen and Dolan 2011) is a widely used benchmark dataset for video captioning methods. It contains 1,970 videos clips collected from YouTube with an average duration of 9.6 seconds. Each video has around 40 human annotated sentences. In our experiments, we follow the split settings in prior works (Xu et al. 2017; Yao et al. 2015): 1,200 videos for training, 100 videos for validation and 670 videos for testing. The resulting training set has a vocabulary size of 9,657.

MSR-VTT. The MSR-VTT dataset (Xu et al. 2016) is a large scale open-domain video captioning dataset. It contains 10,000 video clips with an average duration of 14.9 seconds and 20 human annotated captions per clip. Besides, each clip has an expert-defined category label. We follow the standard dataset split in the dataset paper: 6,513 video for training, 497 videos for validation and 2,990 videos for testing. The resulting training set has a vocabulary size of 23,393.

4.2 Implementation Details

Feature Extraction. For every video, we uniformly sample 20 key frames. Optical flows in both horizontal and vertical directions are computed for 6 consecutive frames centered at each key frame. The flow magnitude is clipped to $[-20, 20]$ and then normalized to $[0, 255]$. The flow images are then cropped and resized so that the CNN outputs match the spatial size of image features. We extract static image features from models pre-trained on the ImageNet: GoogleNet (Szegedy et al. 2015) and Inception-Resnet-V2 (Szegedy et al. 2017). Futures from the C3D network are also included to model the short-term motion information. Note that C3D learns motion information from RGB images, while could be a complement to our model. For the MSR-VTT dataset, we also include audio feature (BoAW (Pancoast and Akbacak 2014)) and the coarse category information.

Training Settings. The LSTMs used in our model all have 1024 hidden units and the word embedding size is set to 512. The optical flow images are normalized so that their pixel values are in the range $[-1, 1]$ before being fed to CNN. We apply dropout with rate of 0.5 to all the vertical connections of LSTMs and L_2 regularization with a factor of 5×10^{-5} to all the trainable parameters to mitigate overfitting. We apply ADAM optimizer with a learning rate of 10^{-4} and batch size of 32 to minimize the negative log-likelihood loss. All the components of our model and training are implemented in Tensorflow[3]. On a commodity GTX 1080 Ti GPU, the times needed to extract frame features and optical flows for a typical 10-second video clip are 400ms and 800ms, respectively. After feature extraction, our model can generate caption for a video in 45ms.

Evaluation Settings. During evaluation/testing, we use beam search with size 5 for sentence generation. We employ three common metrics in video captioning task: BLEU@4, CIDEr, and METEOR. All the metrics are computed by the codes from the Microsoft COCO Evaluation Server[4].

4.3 Compared Methods

We choose to compare our proposed approach with the following state of the art methods. Their major approaches can be grouped to **three categories**: Temporal attention (1,2,3), spatial attention (4,5,6) and multi-modal fusion (7,8,9,10).

1. HRNE with Attention (Pan et al. 2016). HRNE considers the hierarchical structure of the video when encoding, and decodes the sentence with temporal attention.
2. Soft Attention (SA) (Yao et al. 2015). As introduced in Section 2.
3. hLSTMat (Song et al. 2017). In decoding stage, hLSTMat adaptively selects how much of the temporally attended features should be used for generating a specific word.
4. DMRM (Yang, Han, and Wang 2017). As introduced in Section 2.
5. MAM-RNN (Li, Zhao, and Lu 2017). As introduced in Section 2.
6. Dense Caption (Shen et al. 2017). This approach aims to select multiple spatial region sequences via a mapping between frame regions and lexical labels for dense video captioning.
7. MA-LSTM (Xu et al. 2017). MA-LSTM is conceptually similar to Attention Fusion, except that modality-wise fusion is done by the proposed Child-Sum fusion unit.
8. Attention Fusion (Hori et al. 2017). In decoding stage, temporal attentions are computed for multiple modalities and then fused by a modality-wise attention.
9. M&M-TGM (Chen et al. 2017). M&M-TGM uses a multi-modal multi-task training scheme which learns to jointly predict the captions and topic of the videos.

[3]https://github.com/tensorflow/tensorflow

[4]https://github.com/tylin/coco-caption

Dataset	MSVD				MSR-VTT			
Model	Features	B@4	C	M	Features	B@4	C	M
HRNE w/ Attention (Pan et al. 2016)	G	43.8	-	33.1	-	-	-	-
SA (Yao et al. 2015)	G	41.9	51.7	29.6	V+C	36.6	-	25.9
hLSTMat (Song et al. 2017)	G	48.5	-	31.9	R-152	38.3	-	26.3
DMRM w/o SS (Yang, Han, and Wang 2017)	G	50.0	73.2	33.2	-	-	-	-
MAM-RNN (Li, Zhao, and Lu 2017)	G	41.3	53.9	32.2	-	-	-	-
Dense Caption (Shen et al. 2017)	-	-	-	-	R-50+C+A	41.4	48.9	28.3
MA-LSTM (Xu et al. 2017)	G+C	52.3	70.4	33.6	G+C+A	36.5	41.0	26.5
Attention Fusion (Hori et al. 2017)	V+C	52.4	68.8	32.0	V+C+A	39.7	40.0	25.5
M&M-TGM (Chen et al. 2017)	I+C	48.76	80.45	34.36	I+C+A	44.33	49.26	**29.37**
v2t_navigator (Jin et al. 2016)	-	-	-	-	C+A	40.8	44.8	28.2
Aalto (Shetty and Laaksonen 2016)	-	-	-	-	G+C	39.8	45.7	26.9
VideoLab (Ramanishka et al. 2016)	-	-	-	-	R+C+A	39.1	44.1	27.7
MGSA(G)	G	49.5	74.2	32.2	G	39.9	45.0	26.3
MGSA(I)	I	53.0	86.4	34.7	I	41.7	48.1	27.5
MGSA(I+C)	I+C	**53.4**	**86.7**	**35.0**	I+C	42.4	47.5	27.6
MGSA(I+A+C)	-	-	-	-	I+A+C	**45.4**	**50.1**	28.6

Table 1: Captioning performance comparison on MSVD and MSR-VTT. We note the features used for fare comparison, where G, V, C, R-N, I and A denote GoogleNet, VGGNet, C3D, N-layer ResNet, Inception-ResNet-V2, and audio features, respectively. Note that audio is not available on MSVD. "-" means that the authors did not report the corresponding results.

10. MM2016 VTT Challenge winners (Jin et al. 2016; Ramanishka et al. 2016; Shetty and Laaksonen 2016). These approaches mainly use multimodal fusion encoders to fully exploit the visual, motion and audio information in videos.

4.4 Experimental Results of Model Variants

Model	B@4	C	M
Spatial Attention	49.8	72.2	32.9
Spatial Attention w/ GARU	51.0	81.8	34.0
MGSA w/o GARU	51.0	83.3	33.1
MGSA w/ GARU	**53.0**	**86.4**	**34.7**

Table 2: Comparison of model variants on MSVD.

First, we perform experiments on the MSVD dataset to test the effectiveness of individual components of our model. As shown in Table 2, the Spatial Attention is a simplified model of (Li, Zhao, and Lu 2017) with the propagation of spatial attention map removed. We make this modification in order to show the effectiveness of our proposed GARU, which also has the ability of relating attention maps. It can be observed that for both Spatial Attention and MGSA, adding GARU to incorporate the relations of attention maps across time can significantly improve the performance regarding the CIDEr measure: the relative improvement for Spatial Attention and MGSA is 13.3% and 5.6%, respectively. By comparing MGSA to Spatial Attention, we show that even without considering the interrelationship of spatial attention maps, motion-guided attention outperforms spatial attention computed from regional features. Regarding the CIDEr measure, the relative improvement of our MGSA over Spatial Attention is 15.4%. Overall, these comparisons of model variants prove both of our proposed MGSA and GARU to be effective.

4.5 Experimental Results on MSVD

We compare our complete model, i.e., the MGSA w/ GARU in Table 2 to the state of the art models on MSVD. The results are summarized in Table 1. Our approach outperforms approaches which only exploit the temporal structure of videos (HRNE, hLSTMat and SA). This apparently shows that exploiting spatial information in video frames can boost video captioning performance. For models based on spatial attention, MAM-RNN is the most related approach to ours. When both using GoogleNet features, ours significantly outperforms MAM-RNN. This can be attributed to the usage of motion-guided attention and GARU. As for another related approach that utilize spatial attention, DMRM, our approach achieves on-par performance with it. For the multi-modal fusion methods, ours can already outperform them even without fusing multiple features (MGSA(I)). Our full model (MGSA(I+C)) significantly outperforms the best competitor M&M TGM with relative improvements of 9.5%, 7.8% and 1.9% for BLEU@4, CIDEr and METEOR, respectively.

4.6 Experimental Results on MSR-VTT

The performance comparison on MSR-VTT is summarized in Table 1. MGSA again outperforms spatial and temporal attention methods. Notably, most approaches on this dataset are based on multi-modal fusion. Since the videos of MSR-VTT have audio channel and coarse category label. When not multi-modal features is used, MGSA(I) can surpass most of these methods. While our focus in this work is learning spatial attention, our method is compatible with multi-modal information and is expected to gain a performance boost by adding more features to the attended visual feature, i.e., the

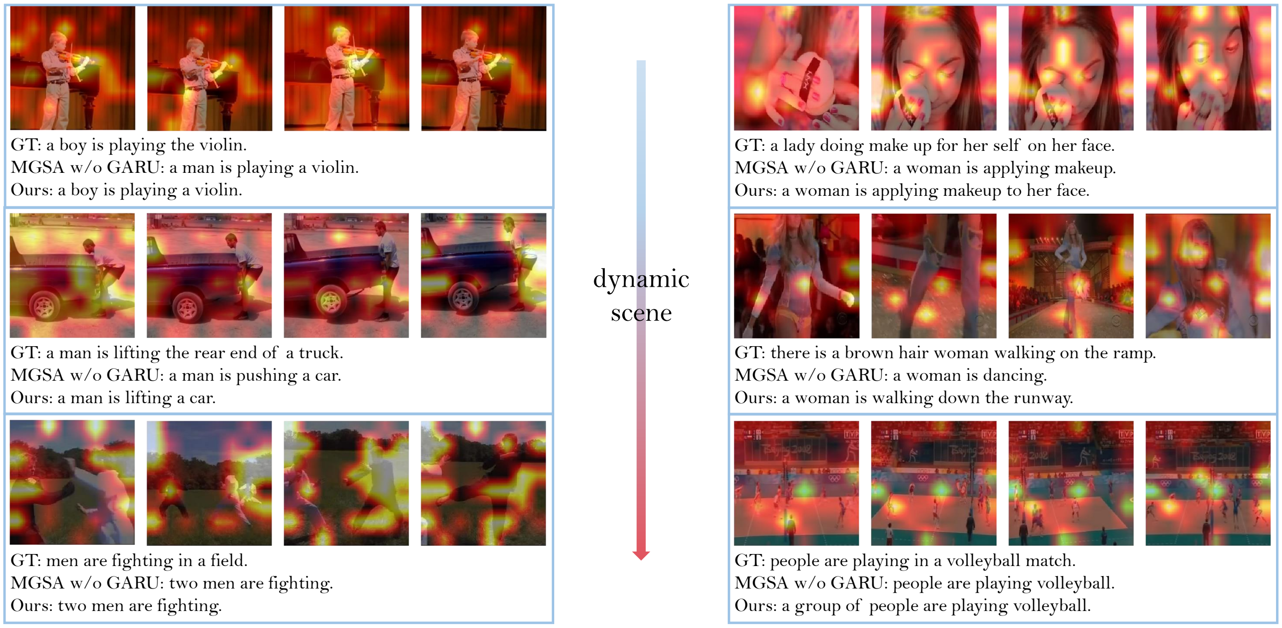

Figure 4: Sample captions generated by our model with and without GARU. The corresponding attention maps are generated by our model with GARU and visualized as heatmaps. The three rows to the left and right are from the MSVD and MSR-VTT, respectively. From top to bottom, the scenes are more and more dynamic.

$\mathbf{G}^{(t)}$ in the decoding stage. When multi-modal features including audio (A) and short-term motion (C) are incorporated, the full model, MGSA(I+A+C) outperforms all other methods. To summarize, the results achieved by our methods are the current state of the art on both datasets.

4.7 Qualitative Analysis

To gain an intuition of the spatial attention learned by our model, we present some example sentences generated by different models along with the attention maps generated by our model (MGSA w/ GARU). In order to demonstrate the effectiveness of motion-guided attention, we select scenes with different degrees of dynamics. In Figure 4, we can see that our model can generate relevant sentences while attending to the important regions of the frames. For example, in the "man lifting car" video, the important region is the man and the car. They can both be captured by optical flow, and our model can then generate accurate attention maps. It is also shown that without GARU, the model can make mistakes in distinguishing the actions such as "pushing/lifting" and "walking/dancing". This indicates that considering the interrelationship between attention maps is essential. For relatively static scenes like in the "playing violin" video, our model can capture the slight action of the person and attend to the important regions. For more dynamic scenes, such as the ones in the third row, there will be dramatic changes caused by camera motion. Our model can still robustly capture the correct attention region. The reason behind this could be that inputing stacked optical flow from multiple frames can mitigate the affection of sudden changes. In the "men fighting" video, our MGSA consistently focus on the fighting men, the changing background does not disturb the attention. Interestingly, in the "volleyball match" video the camera focus rapidly switches between two sides and our MGSA always attends to the focus of the match: it tracks the volleyball.

5 Conclusions

We propose a novel video captioning framework by learning spatial attention under the guidance of motion information. The proposed MGSA utilize motion information between consecutive frames by applying CNN to stacked optical flows. In addition, a gated recurrent unit named GARU is designed to adaptively relate spatial attention maps across time. With all the designs, we achieve the current state of the art results on both MSVD and MSR-VTT. Our future work will consider incorporating semantic information for spatial attention, which may complement the motion-guided attention in recognizing visual concepts.

Acknowledgements. This work was supported in part by two projects from NSF China (61622204, 61572134) and one project from STCSM, China (16QA1400500).

References

Baraldi, L.; Grana, C.; and Cucchiara, R. 2017. Hierarchical boundary-aware neural encoder for video captioning. In *CVPR*, 3185–3194.

Chen, D., and Dolan, W. B. 2011. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 190–200.

Chen, S.; Chen, J.; Jin, Q.; and Hauptmann, A. G. 2017. Video captioning with guidance of multimodal latent topics. In *ACM MM*, 1838–1846.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Donahue, J.; Hendricks, L. A.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Darrell, T.; and Saenko, K. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2625–2634.

Guadarrama, S.; Krishnamoorthy, N.; Malkarnenkar, G.; Venugopalan, S.; Mooney, R. J.; Darrell, T.; and Saenko, K. 2013. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*, 2712–2719.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Hori, C.; Hori, T.; Lee, T.; Zhang, Z.; Harsham, B.; Hershey, J. R.; Marks, T. K.; and Sumi, K. 2017. Attention-based multimodal fusion for video description. In *ICCV*, 4203–4212.

Howard, C. J., and Holcombe, A. O. 2010. Unexpected changes in direction of motion attract attention. *Attention, Perception, & Psychophysics* 72(8):2087–2095.

Itti, L., and Baldi, P. 2005. Bayesian surprise attracts human attention. In *NIPS*, 547–554.

Jin, Q.; Chen, J.; Chen, S.; Xiong, Y.; and Hauptmann, A. 2016. Describing videos using multi-modal fusion. In *ACM MM*, 1087–1091. ACM.

Krishnamoorthy, N.; Malkarnenkar, G.; Mooney, R. J.; Saenko, K.; and Guadarrama, S. 2013. Generating natural-language video descriptions using text-mined knowledge. In *AAAI*.

Li, X.; Zhao, B.; and Lu, X. 2017. MAM-RNN: multi-level attention model based RNN for video captioning. In *IJCAI*, 2208–2214.

Liu, C.; Mao, J.; Sha, F.; and Yuille, A. L. 2017. Attention correctness in neural image captioning. In *AAAI*, 4176–4182.

Pan, P.; Xu, Z.; Yang, Y.; Wu, F.; and Zhuang, Y. 2016. Hierarchical recurrent neural encoder for video representation with application to captioning. In *CVPR*, 1029–1038.

Pancoast, S., and Akbacak, M. 2014. Softening quantization in bag-of-audio-words. In *ICASSP*, 1370–1374.

Ramanishka, V.; Das, A.; Park, D. H.; Venugopalan, S.; Hendricks, L. A.; Rohrbach, M.; and Saenko, K. 2016. Multimodal video description. In *ACM MM*, 1092–1096. ACM.

Shen, Z.; Li, J.; Su, Z.; Li, M.; Chen, Y.; Jiang, Y.; and Xue, X. 2017. Weakly supervised dense video captioning. In *CVPR*, 5159–5167.

Shetty, R., and Laaksonen, J. 2016. Frame-and segment-level features and candidate pool evaluation for video caption generation. In *ACM MM*, 1073–1076. ACM.

Simonyan, K., and Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 568–576.

Song, J.; Gao, L.; Guo, Z.; Liu, W.; Zhang, D.; and Shen, H. T. 2017. Hierarchical LSTM with adjusted temporal attention for video captioning. In *IJCAI*, 2737–2743.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NIPS*, 3104–3112.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S. E.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR*, 1–9.

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. A. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 4278–4284.

Thomason, J.; Venugopalan, S.; Guadarrama, S.; Saenko, K.; and Mooney, R. J. 2014. Integrating language and vision to generate natural language descriptions of videos in the wild. In *COLING*, 1218–1227.

Tran, D.; Bourdev, L. D.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 4489–4497.

Venugopalan, S.; Rohrbach, M.; Donahue, J.; Mooney, R. J.; Darrell, T.; and Saenko, K. 2015a. Sequence to sequence - video to text. In *ICCV*, 4534–4542.

Venugopalan, S.; Xu, H.; Donahue, J.; Rohrbach, M.; Mooney, R. J.; and Saenko, K. 2015b. Translating videos to natural language using deep recurrent neural networks. In *NAACL*, 1494–1504.

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Gool, L. V. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 20–36.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A. C.; Salakhutdinov, R.; Zemel, R. S.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2048–2057.

Xu, J.; Mei, T.; Yao, T.; and Rui, Y. 2016. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, 5288–5296.

Xu, J.; Yao, T.; Zhang, Y.; and Mei, T. 2017. Learning multimodal attention LSTM networks for video captioning. In *ACM MM*, 537–545.

Yang, Z.; Han, Y.; and Wang, Z. 2017. Catching the temporal regions-of-interest for video captioning. In *ACM MM*, 146–153.

Yao, L.; Torabi, A.; Cho, K.; Ballas, N.; Pal, C. J.; Larochelle, H.; and Courville, A. C. 2015. Describing videos by exploiting temporal structure. In *ICCV*, 4507–4515.

Zhu, L.; Xu, Z.; and Yang, Y. 2017. Bidirectional multirate reconstruction for temporal modeling in videos. In *CVPR*, 1339–1348.