Spatio-Temporal Ranked-Attention Networks for Video Captioning

Anoop Cherian[1] Jue Wang[2] Chiori Hori[1] Tim K. Marks[1]
[1]Mitsubishi Electric Research Labs, Cambridge, MA [2]Australian National University, Canberra
{cherian,chori,tmarks}@merl.com jue.wang@anu.edu.au

Abstract

Generating video descriptions automatically is a challenging task that involves a complex interplay between spatio-temporal visual features and language models. Given that videos consist of spatial (frame-level) features and their temporal evolutions, an effective captioning model should be able to attend to these different cues selectively. To this end, we propose a Spatio-Temporal and Temporo-Spatial (STaTS) attention model which, conditioned on the language state, hierarchically combines spatial and temporal attention to videos in two different orders: (i) a spatio-temporal (ST) sub-model, which first attends to regions that have temporal evolution, then temporally pools the features from these regions; and (ii) a temporo-spatial (TS) sub-model, which first decides a single frame to attend to, then applies spatial attention within that frame. We propose a novel LSTM-based temporal ranking function, which we call ranked attention, for the ST model to capture action dynamics. Our entire framework is trained end-to-end. We provide experiments on two benchmark datasets: MSVD and MSR-VTT. Our results demonstrate the synergy between the ST and TS modules, outperforming recent state-of-the-art methods.

1. Introduction

The recent advances enabled by deep neural networks in computer vision, audio, and natural language processing have stimulated researchers to look beyond these as isolated domains, instead tackling problems at their intersections [49, 15, 74, 10]. Automatic video captioning is one such multimodal inference problem that has gained attention in recent years [26, 57, 59], thanks to the availability of sophisticated CNN models [8, 17, 4, 48] and massive training datasets for video activity recognition [30, 22, 28], audio classification [20], and neural machine translation [41, 5]. However, learning to describe video data is still a challenging problem, as generating good captions requires inferring the intricate relationships and interactions between subjects and objects in a video. Despite recent progress [11, 26, 57, 59], this task remains difficult. This may be due to the high dimensionality of spatio-temporal data, which can generate large volumes of features of which only a few may be correlated to the way humans describe videos.

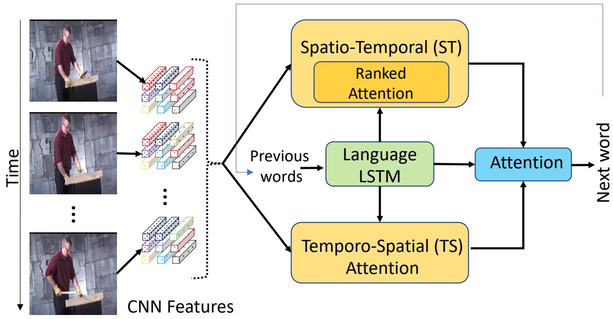


Figure 1. Our overall spatio-temporal and temporo-spatial (STaTS) attention architecture.

Taking inspiration from neural translation models, one promising way to approach the video captioning problem is to leverage *visual attention* [50, 71, 3, 62]. Such techniques use the compositional nature of language models to attend to specific visual cues in order to generate subsequent words in a caption. Attention has also been explored for multimodal fusion using image, audio, and motion cues [59, 26]. However, these works consider frame-level or clip-level representations of videos, which may not capture specific details of the scene or may represent too much information that is unrelated to the primary content.

There have been efforts to address such granularity issues by using spatial attention, as for example in image captioning [3, 64]. Such schemes usually use a pre-trained object detector, e.g., Fast RCNN [43], which may be useful for detecting specific objects in the scene but may miss out on the scene context or visual cues related to human actions or interactions. One could also use schemes such as action proposals [32, 66], but they can be computationally expensive. This paper is similar in vein to these works, in that we also explore video captioning using spatial and temporal attention. However, we apply and combine these attentions in a novel way.

Our main contribution is an attention model that we call STaTS (Spatio-Temporal and Temporo-Spatial). Our model, illustrated in Figure 1, hierarchically combines spatial and temporal attention in two different orders, which we call spatio-temporal (ST) attention and temporo-spatial (TS) attention. For ST attention, we first apply spatial attention and linear pooling on deep features derived from each video frame, then apply a temporal attention over these features. The ST model's composition of spatial and temporal attention modules helps reduce the size of the spatial/temporal attention space from multiplicative to additive.

Further, to ensure that temporal pooling captures the dynamic nature of actions in videos, we introduce a novel LSTM-based ranking formulation that attends to consecutive pairs of frames in a way that preserves their temporal order. We call this *ranked attention*. Our key idea is to use an LSTM to emulate a rank-SVM [19] such that the representation this module generates captures the temporal evolution of video features. Such a technique avoids the otherwise computationally challenging implicit differentiation that one needs to use for rank-pooling [18, 21].

One weakness of the ST model may be that not all words in a caption rely on such temporally varying holistic features. Words for the *subject* or *object*, for example, might be more directly obtained by considering more localized features from a single representative frame. To this end, we propose a novel temporo-spatial (TS) attention model that provides a shortcut for visual relationship inference, without going through the ST pipeline described above. Specifically, the TS pipeline first applies temporal attention to frame-level representations to (softly) select specific frames to attend to, then applies spatial attention to the spatial feature representations of these frames.

Our STaTS model generates two attention-weighted video representations (ST and TS), which we combine via a weighted average, conditioned on the state of the language model (sentence generator), where these weights are computed by passing the two representations through a further attention scheme across the ST and TS models.

In Section 4, we present experiments evaluating the benefits of each of the above modules. We base our experiments on two frequently used video captioning benchmarks: the MSVD (YouTube2Text) [23] and MSR-VTT [61] datasets. For the spatial features, we explore the advantages of using 3D CNN features from the recent Inflated 3D (I3D) activity recognition model [8], as well as features from a Fast RCNN object detection model [43]. Our experiments clearly demonstrate the advantages of our STaTS model, leading to state-of-the-art results on the MSVD dataset on all evaluation metrics. On MSR-VTT, we achieve the best performance on some metrics and are competitive with the recent state of the art on others.

We now summarize the main contributions of this work:

1. We present a novel spatio-temporal and temporo-spatial attention model, in which each of the two submodels selectively attends to complimentary visual cues required to generate sentences.
2. We propose a novel temporal attention scheme, *ranked attention*, by formulating an LSTM-based objective that emulates a rank-SVM algorithm for temporally ordered feature aggregation.
3. We present extensive experiments and analysis on two benchmark datasets, using varied 2D and 3D CNN-based feature representations, and demonstrate state-of-the-art performance.

2. Related Work

Video Captioning. Traditional methods for video captioning are usually based on predefined language templates [23, 31, 45, 33, 49, 67, 13, 27, 60], which reduce a free-form caption generation model into one of recognizing the categories to fill in for various attributes and keywords in the template (such as the subject, verb, and object). For example, in Rohrbach et al. [45], a conditional random field is proposed to model the correlation between activities and objects in the video. Markov models are also adopted to produce semantic features for sentence generation [67, 13, 27, 60]. Such models disentangle the need for the language model to learn grammar, thereby simplifying the problem. However, the captions generated are limited by the syntactical structure, which limits their diversity and the system's ability to generalize. In contrast to these prior works, there have been recent efforts at leveraging deep recurrent architectures such as long short-term memory (LSTM) for sequence learning tasks, starting with the seminal work of Karpathy et al. [29]. Venugopalan et al. [54] propose an LSTM-based model to generate captions from temporally average-pooled CNN visual features. Since the average pooling destroys the temporal dynamics of the sequence, Yao et al. [65] present a temporal attention mechanism to associate a weighting for the feature from each frame and fuse them using a weighted average. Along similar lines, Venugopalan et al. [53] introduce S2VT, which utilizes LSTMs in both encoder and decoder and includes optical flow to incorporate temporal dynamics. Zhang et al. [72] propose a two-stream feature encoder to aggregate both spatial and temporal cues jointly using 3D CNN features. Hori et al. [26] extend temporal attention by attending to different input modalities such as image, motion, and audio features. Our method differs from these in the way we disentangle the video features. Our approach allows simultaneous hierarchical and coupled extraction of spatio-temporal video cues in a simple framework.

Spatio-Temporal Attention. As mentioned above, temporal attention has been widely used in recent video captioning work to decide which frame(s) in the video are important for generating the next word in a caption. However, these systems usually map the raw video frames into high-level CNN features (via a suitable spatial pooling operator), which marginalizes away important spatial information (such as location and class of specific objects or actions) that are important for captioning.

Spatial-temporal video feature learning has been widely used in several video applications, such as video classification [40, 16, 69] and video super-resolution [63]. Related work in image captioning includes [3], which applies top-down and bottom-up attention to Fast R-CNN features, and [38], which applies an attention-based LSTM to generate a spatially weighted feature map. In video captioning, Yang et al. [64] localize regions of interest in every frame using attention; however, not every frame may have have such a region, and they need additional semantic supervision to attend to informative regions. Zanfir et al. [71] propose a spatial-temporal attention model that assigns a weighting to both spatial and temporal CNN visual features from optical flow, RGB frames, and detected objects in each frame. Tu et al. [50] and Yu et al. [68] propose hierarchical attention schemes that condition on the current caption word and visual features. They first generate spatial attention weights, conditioned on which a similar attention scheme is adopted temporally; the weighted features are used to generate the word. While this scheme shares a similar motivation to ours, their attention model must select from a much larger number of features—a harder attention problem that demands larger datasets for training. We avoid this difficulty by attending to spatial and temporal features in stages, each stage reducing the data complexity. More recently, Aafaq et al. [1] use spatio-temporal feature engineering to improve captioning performance. In [73], object saliency is combined with bidirectional temporal graph reasoning; this is related to our proposed ranked attention model, but our formulation is much simpler.

Reinforcement Learning (RL). There are two key ways a video captioning problem can be cast in an RL setting: (i) selecting informative features or frames, and (ii) optimizing the training on evaluation metrics that are usually not differentiable (such as BLEU, CIDER, METEOR, etc.). For the former setting, several recent works have achieved promising results [11, 58] by picking suitable frames to encode based on a predesigned reward function. Chen et al. [11] incorporate visual diversity and CIDER score into the reward function. Similarly, [58] models a manager and a worker within a hierarchical LSTM to achieve better feature encoding. When using RL to optimize non-differentiable losses, prior works typically use the policy-gradient algorithm [11]. While we believe our sophisticated attention scheme can pick visual features without needing an RL engine, we do use policy gradients to optimize our model for losses defined over METEOR and BLEU metrics (as in [44]).

3. Proposed Method

In this section, we introduce our Spatio-Temporal and Temporo-Spatial (STaTS) attention model for video captioning, illustrated in Figure 1. In Section 3.1, we describe our spatio-temporal (ST) attention model, which consists of a spatial attention model (Section 3.1.1) followed by our proposed ranked temporal attention model (Section 3.1.2). We explain our temporo-spatial (TS) model in Section 3.2. Finally, we describe how the ST and TS models are combined into our full STaTS attention model in Section 3.3.

Before proceeding, let us review our notation. Suppose we are given a training set of N videos, $\mathfrak{S} = \{(\mathcal{S}_1, \mathcal{Y}_1), (\mathcal{S}_2, \mathcal{Y}_2), \cdots, (\mathcal{S}_N, \mathcal{Y}_N)\}$. Here, $\mathcal{S}_k$ is a temporally ordered sequence of frame-level features for video k, and each $\mathcal{Y}_k$ is a textual description of the video (caption), the words of which are encoded using their indices in a predefined dictionary. Let each video sequence $\mathcal{S}_k = \langle x_1, x_2, \cdots, x_T\rangle$ be a sequence of T temporally ordered video frames. For each video frame t, we have n features, denoted x_{tj} for $t = 1, 2, \ldots, T$ and $j = 1, 2, \ldots, n$, where each $x_{tj} \in \mathbb{R}^d$. For each j, x_{tj} encodes visual information from a different region (out of n regions of the image). Such spatial features could be produced, for example, from each cell of a non-overlapping grid as from the intermediate spatial pooling layers of a CNN, or regions obtained from an RCNN object detector. To encode captions, we assume each $\mathcal{Y}_k = \langle y_1, y_2, \cdots, y_m\rangle$ is an ordered sequence of word embeddings, where the ith word in the caption, $y_i \in \mathbb{B}^D$, is a one-hot vector encoded using a language dictionary of size D.

Given that the size of the language dictionary D is usually enormous, learning a neural network model to generate a caption with m words would demand exploring a space of D^m sentences, which may be computationally challenging. Fortunately, however, the language model is highly structured and compositional, so one can generate each word sequentially conditioned on the previously generated words. This idea is usually implemented via a long short-term memory (LSTM), which takes as input the current word y_i in a sentence $\mathcal{Y}_k$ and a state representation h_{i-1} of the previous words in the sentence, and produces a new state as output: $h_i = \mathrm{LSTM}(h_{i-1}, y_i)$. Apart from the language model, an integral part of the caption generation process is selecting informative visual features from the videos to be fed to the language model (which is also the main contribution of this paper). A standard approach to this problem is to use *visual attention*. Mathematically, let $e \in \Delta^T$ be a probability vector in the T-dimensional simplex; its tth dimension e_t captures the probability that visual feature x_t is useful for generating the ith word, typically given by:

$$e_t = \mathrm{softmax}\left(\mathrm{att}\left(h_{i-1}, x_t\right)\right), \tag{1}$$

where att is a suitable nonlinear attention function, usually

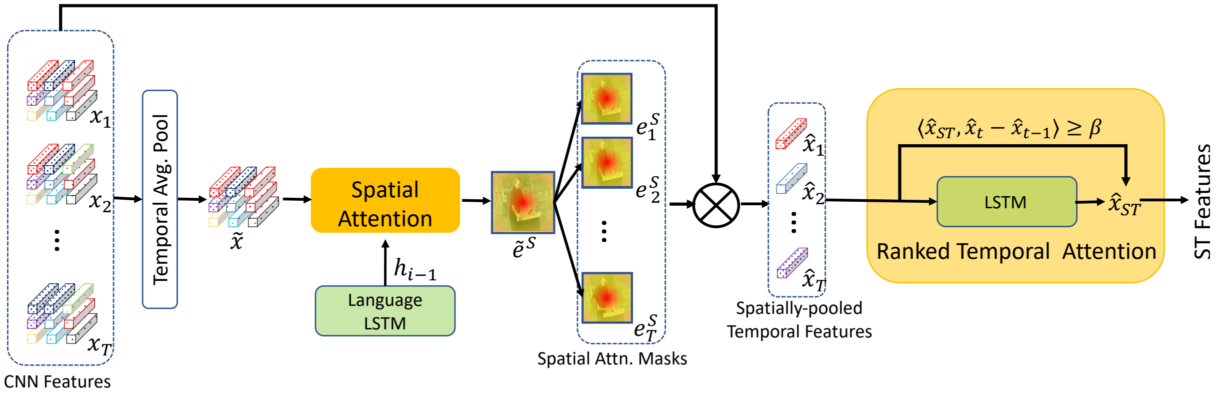


Figure 2. Our spatio-temporal (ST) network with the ranked temporal attention module.

chosen as

$$\mathrm{att}(h_{i-1}, x_t) = w^{\mathrm{T}} \tanh\left(W_h h_{i-1} + W_x x_t + b\right). \quad (2)$$

Here, b is a learned bias, while W_h and W_x are learned matrices transforming the respective features into an attention space, in which they are linearly combined using the w vector after passing through the nonlinear $\tanh$ function. The score e is projected onto the simplex via the $\mathrm{softmax}$ operator in (1), thereby generating a probability vector over the visual features. The visual features x_t are linearly combined using weights e_t to produce the attended visual feature.

3.1. Spatio-Temporal (ST) Attention

In this section, we present the Spatio-Temporal module of our attention framework. As may be noted, using multiple spatial (region-based) features (for every frame) introduces an additional degree of freedom in the visual domain (as against using only a single feature per frame), which needs to be attended to effectively. A straightforward way to extend the temporal attention in (1) to the spatio-temporal domain would be to ignore the spatial nature of these additional features and treat all nT features as if they were the temporal features of the standard temporal attention model. However, given that each spatial feature could be noisy (i.e., containing features irrelevant or redundant to the end task), increasing the number of features to be attended may amplify the noise, thus diluting the attention paid to useful features. Further, there is temporal continuity in these features that should be incorporated in the method, for example to attend to spatially localized actions that span across frames. To circumvent such issues, we propose to compose the spatial and temporal attention one after the other. We explain the spatial aggregation in Section 3.1.1, then explain the subsequent temporal attention in Section 3.1.2. Figure 2 illustrates our ST pipeline.

3.1.1 ST Model: Spatial Attention

A direct way to implement spatial attention is to use (1) on each frame. That is, let e_t^S denote the spatial attention for frame t:

$$e_{tj}^S = \mathrm{softmax}\left(\mathrm{att}(h_{i-1}, x_{tj})\right), \text{ where } \sum_{j=1}^{n} e_{tj}^S = 1. \quad (3)$$

However, such a formulation makes no assumptions about the temporal relationships of the attended features from frame to frame. For example, when one needs to reason about the temporal evolution of video regions, say for generating the *verb* part of a caption, a temporally-consistent spatial attention is preferred—we would like to attend to regions that contain the same entity over multiple frames. *But how can we generate such consistent attention in a computationally inexpensive way?* We propose a simple way to achieve this by making some practical assumptions about the way the spatial regions are organized in the videos. Specifically, we assume these regions form a fixed non-overlapping grid (see the input CNN Features in Figure 2), and each spatial feature summarizes the semantics in that grid location. Such an arrangement is a natural output of standard CNN pooling layers; e.g., the I3D model generates a 7×7 grid of spatio-temporal features. This grid is assumed to be consistent across all frames; as a result, when camera motion and scene changes are absent in the video, the features from the same grid cell across the frames are temporally consistent. However, when the camera moves or the scene changes, such an assumption no longer holds.

We circumvent this problem by *overestimating* the spatial attention region. Specifically, we propose a three-step process. First, we aggregate the spatial features at each grid cell across the temporal dimension, i.e., compute $\tilde{x}_j = \frac{1}{T}\sum_{t=1}^{T} x_{tj}$. Next, we use $\tilde{x}$ (which only contains n features) in (3) to compute spatial attention $\tilde{e}^S$. Finally, we replicate this attention to all frames: $e_t^S = \tilde{e}^S$ for all $t = 1, 2, \ldots, T$ (see Figure 2 middle block).

Given that our proposed spatial attention is an approximate union of the attentions for individual frames, feature noise due to short scene changes or camera motion may be diluted when averaging the spatial features over all the frames. When training the framework end-to-end alongside the temporal ranked attention scheme (discussed in the next section), our overestimated attention will be guided to be correlated with regions in the video that exhibit dynamics, thereby pruning away non-action-related cues. Further, our heuristic also reduces the inference time linearly as the number of attentions to compute in this module is now independent of the number of frames in the sequence. Once the spatial attention e^{S}_{tj} is computed, it is used to linearly average pool the spatial features for every frame (using (2)), thus producing T temporally-ordered features $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_T$ for the next module.

3.1.2 ST Model: Ranked Temporal Attention

In this section, we detail our temporal pooling scheme, *ranked temporal attention* (also in Figure 2). Using the spatially attended features $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_T$ produced by the spatial attention module described above, our goal is to capture the action dynamics in the input features. While there are several choices for modeling such dynamics popular in action recognition literature [8, 70, 17], we decided to use a model that is simple, effective, and lightweight. A standard approach is to use an LSTM for this task, but it is not guaranteed to capture the action dynamics unless it is trained with a suitable loss.

To this end, we take inspiration from recent work on ranking-based dynamic feature pooling [7, 19, 12]. For temporally ordered inputs $\langle \hat{x}_1, \hat{x}_2, \ldots, \hat{x}_T \rangle$, these methods propose to compute a feature w by solving the following rank-SVM formulation:

$$\arg\min_{w} \left[\frac{1}{2} \|w\|_2^2 + \lambda \sum_{t=1}^{T-1} \operatorname{softplus}(\zeta_t) \right], \tag{4}$$

$$\text{where } \zeta_t = \langle w, \hat{x}_t \rangle + \beta - \langle w, \hat{x}_{t+1} \rangle, \tag{5}$$

where $\lambda > 0$ is a regularizer, and $\operatorname{softplus}(z) = \log(1 + e^z)$ is a soft variant of the popular ReLU activation function. The rank pooling formulation seeks to find a direction $w \in \mathbb{R}^d$ (same dimension as the input features) such that projecting the inputs to this direction will preserve their temporal order (with a margin of $\beta > 0$), as enforced by the softplus function. Intuitively, the minimization encourages the projection of each frame's input feature, $\langle w, \hat{x}_{t+1} \rangle$, to be larger than the projection of the previous frame's input feature, $\langle w, \hat{x}_t \rangle$. Thus, the intuition is that this direction w, which lies within the input space, captures the temporal order (temporal dynamics), and can be used as an aggregated video feature for subsequent tasks. This has been found to be empirically useful in several recent works [7, 12].

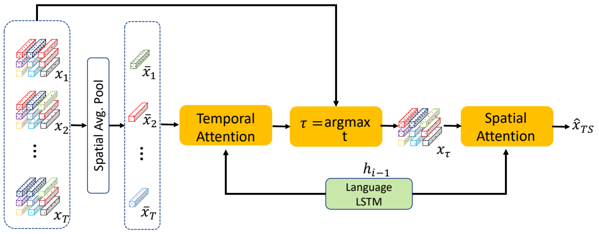


Figure 3. Our temporo-spatial (TS) attention module.

However, there is an important caveat for directly using rank pooling within a deep CNN framework: namely, (4) involves computing an $\arg\min$ function, which is not differentiable. While there are workarounds for computing the derivative of this function [21], they lead to second-order gradients, which can be computationally expensive and may even be infeasible when the feature dimensionality is high. To circumvent this problem, we propose a simple scheme in this paper that we call *ranked attention*.

Our key idea is backed by the well known theoretical result that a recurrent neural network can approximate any algorithm (Turing machine) [47]. Motivated by this result, we propose to emulate the ranking SVM solution described above within an LSTM setting, such that it takes as input the sequence of features and produces a feature w as output while also minimizng the softplus loss specified by (4). Specifically, suppose the LSTM is an abstract function [25] parametrized by weights θ. Then, using the above notation, we define our temporal pooling module (during training) as one that generates a representation $\hat{x}_{ST}$ by learning θ that optimizes the following loss:

$$\min_{\theta} \sum_{t} \operatorname{softplus}(\zeta_t), \tag{6}$$

$$\text{where} \quad \zeta_t = \langle \hat{x}_{ST}, \hat{x}_{t-1} \rangle + \beta - \langle \hat{x}_{ST}, \hat{x}_t \rangle, \tag{7}$$

$$\text{and} \quad \hat{x}_{ST} = \bigoplus_{t=1}^{T} \operatorname*{LSTM}_{\theta}(\hat{x}_t). \tag{8}$$

Here, $\hat{x}_{ST}$ denotes the final output of the LSTM after it has seen all T features. (The notation $\oplus$ denotes the sequential nature of inputting the features $\hat{x}_1, \ldots, \hat{x}_T$ to the LSTM, one frame at a time, while updating its internal state.) Intuitively, the formulation (8) learns to produce a feature representation that preserves the temporal order of the input features; these features were output by our spatial attention model. Since the entire system is trained end-to-end, minimizing the softplus loss in turn trains the spatial attention to attend to temporally varying features, i.e., action dynamics. In (8), we avoid optimizing through $\arg\min$ as in (5), instead optimizing the LSTM parameters θ alongside other STaTS parameters while respecting the order constraints.

3.2. Temporo-Spatial (TS) Attention Model

The ST attention model may help the system generate caption words for dynamic visual features (e.g., verbs), but attention to such temporal cues may not be necessary when generating words for the *subject* or *object* in a caption. For example, consider the sentence *a boy is playing with a ball*. Here, the verb *playing* may benefit from ST attention. However, using the ST attention framework for generating words such as *boy* or *ball* may be overkill and inefficient, so we need a more direct way to infer them.

To this end, we propose a separate attention-over-attention model, which we call *temporo-spatial attention.* In this model, we first use the standard temporal attention scheme described in (1), then greedily select a single frame (or a few frames) to attend to (see Figure 3). Next, we apply spatial attention only to the features within these frames. Mathematically, suppose $\bar{x}_t$ represents a spatially agglomerated feature representation for frame t (here $\bar{x}_t$ could be the average of all the spatial features for this frame, or a Max-Pooled vector). Our temporo-spatial (TS) attention is thus:

$$\tau = \arg\max_t \operatorname{att}(h_{i-1}, \bar{x}_t), \tag{9}$$

$$e_j^{TS} = \operatorname{att}(h_{i-1}, x_{\tau j}), \quad \text{where } \sum_j e_j^{TS} = 1. \tag{10}$$

We define the *temporo-spatial attention* feature as:

$$\hat{x}_{TS} = \sum_j e_j^{TS} x_{\tau j}\,. \tag{11}$$

Note that while we write the frame selection via an $\arg\max$ function, we implement it via a $\operatorname{softmax}$ with a low temperature, as otherwise the model is non-differentiable.

3.3. Spatio-Temporal and Temporo-Spatial Model

For our full STaTS model, we combine the ST and TS models defined above via a further language attention-based weighting (see Figure 1). Let β_1 and β_2 be weight scalars: $\beta_1 = w_{ST}\tanh(W_{ST}\hat{x}_{ST} + W_h h_{i-1})$ and $\beta_2 = w_{TS}\tanh(W_{TS}\hat{x}_{TS} + W_h h_{i-1})$, where $W_{TS}, W_{ST}, w_{ST}, w_{TS}$ are learned parameters. Our STaTS model produces a combined feature representation:

$$\hat{x} = \tanh\left(\frac{\exp(\beta_1)\hat{x}_{ST} + \exp(\beta_2)\hat{x}_{TS}}{\exp(\beta_1) + \exp(\beta_2)}\right). \tag{12}$$

This is another level of attention conditioned on the language state, which determines how much to attend to each attention branch (ST or TS) when generating the next caption word.

3.4. Model Training

Our STaTS model is trained end-to-end using the ground truth video captions. A natural question in this regard is: what loss should we use? While softmax cross-entropy loss is the standard loss to consider, it is often argued that the cross-entropy may be only weakly correlated with the evaluation metrics we typically use on captions (such as METEOR or BLEU score). However, these metrics are non-differentiable and thus cannot be directly used. To this end, we follow [42, 36] to consider these metrics as reward functions in a reinforcement learning setup, and use policy gradients via the REINFORCE algorithm for optimizing against them. Specifically, following [42], we first optimize our STaTS model to minimize the cross-entropy loss (for about 10 epochs), then subsequent iterations are optimized using a combination of cross-entropy loss and METEOR+BLEU rewards. We also use teacher forcing via scheduled sampling [6] to reduce exposure bias when training the model.

4. Experiments

To validate the effectiveness of our STaTS architecture, we present experiments on the MSVD [9] and the MSR-VTT datasets [61], two popular benchmarks for video captioning. The MSVD dataset includes 1970 videos, split into 1200 videos for training, 100 for validation, and 670 for test, which is the recommended evaluation. Each video has about 40 ground truth (human-generated) captions, and 13,010 distinct words. MSR-VTT is has 10K training and 2990 test sequences and nearly 200,000 captions.

4.1. Implementation and Evaluation

As the primary contribution of this work is our spatio-temporal attention model, we mainly use two state-of-the-art CNN architectures for generating such features: (i) the Inflated 3D architecture (I3D) proposed in [8], which has shown state-of-the-art performance on activty recognition benchmarks; and (ii) Faster R-CNN algorithm [43] using a ResNet-101 architecture (FRCNN). The I3D features are generated for two modalities: (i) temporal chunks of 16 RGB frames at a temporal stride of 16, and (ii) temporal chunks of 16 optical flow frames at stride of 16. The I3D model implicitly uses the Inception-V3 architecture; we extract the spatial features from the "Mixed_5c" layer of this network, which are $2 \times 7 \times 7 \times 1024$ dimensional, which we reshape to $7 \times 7 \times 2048$, where the first two dimensions capture a 7×7 spatial grid. We use the same for the flow features. For the FRCNN features, we pass each frame (at a stride of 16) through a region-pooled ResNet-101 network [24]. We detect a fixed 10 bounding boxes per frame and extract features from the last fully-connected layer of the network, resulting in 10×2048 spatial features. However, unlike the grid-structured I3D features, the RCNN features are region-pooled without any fixed grid. On the MSR-VTT dataset, we provide results using ResNet-152 features as well, to understand the differences in our perfor-

Dataset	Scheme	Feature	CIDEr	BLEU4	ROGUE	METEOR
MSVD	ST	I3D	0.742	0.502	0.68	0.325
	TS	I3D	0.521	0.391	0.646	0.289
	STaTS	I3D	**0.802**	**0.526**	**0.695**	**0.335**
	ST	FRCNN	0.686	0.477	0.69	0.33
	TS	FRCNN	0.439	0.376	0.633	0.274
	STaTS	FRCNN	**0.709**	**0.492**	**0.68**	**0.319**
MSR-VTT	ST	I3D	0.429	0.397	0.600	0.271
	TS	I3D	0.427	0.380	0.595	0.273
	STaTS	I3D	**0.434**	**0.401**	**0.604**	**0.275**

Table 1. Combinations our method on the MSVD and MSR-VTT datasets using the I3D (RGB) and Faster R-CNN features.

Scheme	CIDEr	BLEU4	ROGUE	METEOR
Mean Pool	0.389	0.362	0.580	0.263
LSTM	0.385	0.347	0.578	0.261
Mean + LSTM	0.388	0.364	0.575	0.259
Temp Att	0.382	0.368	0.580	0.258
Mean + Temp Att	0.385	0.368	0.58	0.26
Ranked Att (ours)	0.387	0.376	0.589	0.264
Mean + Ranked Att (ours)	**0.404**	**0.376**	**0.592**	**0.268**

Table 2. Study on the benefits in using Ranked Attention. The results are on the MSR-VTT dataset using the I3D (RGB) features.

Scheme	CIDEr	BLEU4	ROGUE	METEOR
PickNet [11]	0.765	0.523	0.696	0.333
M^3 [56]	N/A	0.520	N/A	0.321
LSTM-LS [37]	N/A	0.511	N/A	0.326
MA-LSTM [62]	0.704	0.523	N/A	0.336
MAM-RNN [34]	0.539	0.413	0.688	0.322
RecNet [55]	0.803	0.523	0.698	0.341
GRU-EVE [2]	0.781	0.479	**0.715**	**0.350**
STaTS(FR+FL)	0.747	0.495	0.694	0.334
STaTS (I3D+FL)	**0.835**	**0.548**	0.711	**0.350**

Table 3. Comparisons to the state of the art on MSVD dataset. FR standds for FRCNN models, I3D and FL stands for the I3D RGB and optical flow models respectively.

mance due to feature type. Note that for either dataset, there is no standard feature type for comparing to prior methods; e.g., PickNet [11] uses ResNet-152, while DenseCap [46] uses C3D.

We use PyTorch software to implement our models. The CNN features are pre-computed and are embedded into 512-dimensions, while the words are embedded in 256 dimensions. We use single-head self-attention on the previously generated words (recall that the caption is generated sequentially, word by word) before combining them with the LSTM state for visual attention. We use an additive attention scheme with the query and key combined in an attention space of 128 dimensions [51]. The models are trained using RMSprop algorithm with a learning rate of 0.0001. The training usually converges in about 20 epochs. We use a batch size of 32 for I3D or FRCNN features. The scheduled sampling uses a teacher forcing ratio of the form $\eta/\big(\eta + \exp(p/\eta)\big)$, where $\eta = 24$ and p is the epoch. To evaluate the performance of our models, we use BLEU4 [39], METEOR [14], ROUGE-L [35] and CIDEr [52]. For fair comparisons with previous work, we compute scores using the code released on the Microsoft COCO evaluation server [10].

Scheme	CIDEr	BLEU4	ROGUE	METEOR
Dense-Cap [46]	**0.489**	0.414	0.611	0.283
PickNet [11]	0.441	0.413	0.598	0.277
OA-BTG (R200) [73]	0.469	0.414	–	0.282
M^3-VC [56]	–	0.381	–	0.266
GRU-EVE (C3D+IVR2) [2]	0.481	0.383	0.607	**0.284**
RecNet [55]	0.427	0.391	0.593	0.266
STaTS (R152)	0.445	0.392	0.597	0.279
STaTS (R152+C3D)	0.465	0.416	**0.615**	**0.284**
STaTS (I3D)	0.434	0.401	0.604	0.275
STaTS (I3D+FL)	0.438	0.410	0.611	0.276
STaTS (I3D+FL+C)	0.451	**0.417**	0.612	0.280

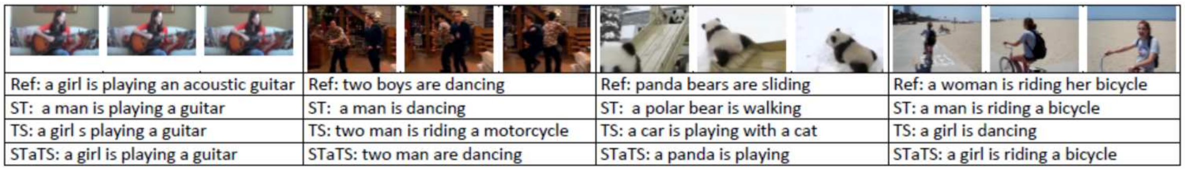
Table 4. Comparisons to the state of the art on MSR-VTT dataset. I3D and FL stand for the I3D RGB and optical flow models, respectively, while C stands for using the class annotations supplied with the dataset during training (as is also used by other methods).

4.2. Results

In the following, we first conduct an ablation study of the various components in our framework.

ST Spatial Attention: Table 1 shows the performance on MSVD and MSR-VTT datasets using I3D and FRCNN features with various attention schemes. We show the performance when using only our spatio-temporal (ST) model, only temporo-spatial (TS), and our combined STaTS model. TS is usually the weakest model, likely due to its greedy attention scheme. The table shows that there is significant synergy between the ST and TS models as substantiated on both the datasets. The table also compares our approximate ST attention and grid-based feature organization (using I3D features) against the alternative of attending to different image regions per frame (using FRCNN features). This comparison (ST and TS in the first two meta-rows of Table 1) shows that our heuristic performs significantly better than FRCNN on CIDER and BLEU4, which are measures capturing the exact match of parts of the generated caption with the ground truth. Also, comparing the full STaTS model using I3D and FRCNN shows that our model is substantially better (0.802 vs. 0.709 on CIDER).

Ranked Attention: In Table 2, we demonstrate the benefits of our ranked temporal attention scheme versus several other plausible choices on the MSR-VTT dataset using the I3D RGB features. We compare to: (i) using mean pooling of the spatially-pooled temporal features, (ii) using an LSTM, (iii) combining LSTM with average pooling, (iv) temporally attending over all spatio-temporal features (no ST-attention nor ranked attention), and (v) using average pooling of spatial features and then temporal pooling of them. We see that while the ranked attention by itself is not significantly better than other choices, combining ranking with average pooling demonstrates the best performance. This is not surprising, given that the ranked attention considers only the ordering of the features, but discards features that are invariant to temporal permutation (features which are captured by mean pooling). We use the combination of mean-pooling + ranked attention in our subsequent model.



Ref: a girl is playing an acoustic guitar	Ref: two boys are dancing	Ref: panda bears are sliding	Ref: a woman is riding her bicycle
ST: a man is playing a guitar	ST: a man is dancing	ST: a polar bear is walking	ST: a man is riding a bicycle
TS: a girl s playing a guitar	TS: two man is riding a motorcycle	TS: a car is playing with a cat	TS: a girl is dancing
STaTS: a girl is playing a guitar	STaTS: two man are dancing	STaTS: a panda is playing	STaTS: a girl is riding a bicycle

Figure 4. Qualitative results using our attention model.

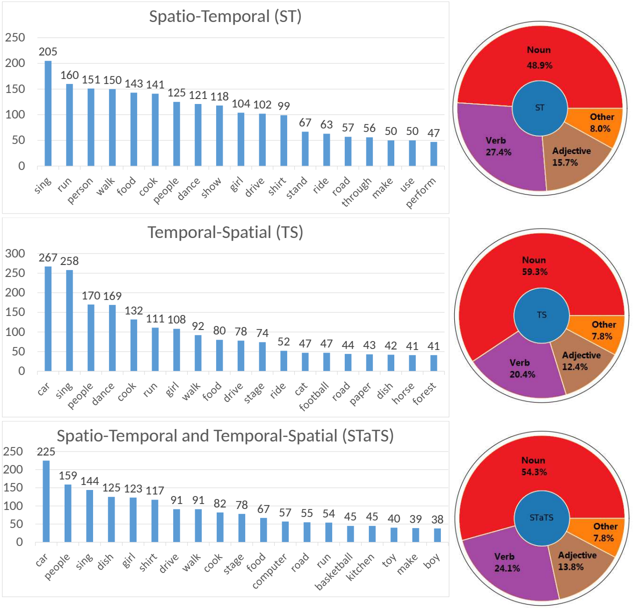


Figure 5. Words distribution analysis for generated captions in MSR-VTT testing set.

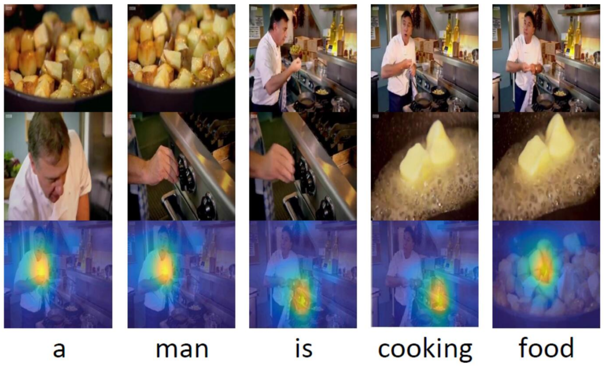


Figure 6. Attention Visualization. The 10 frames in the first two rows show the temporal sequence of the video. The 3rd row shows the frames selected by our TS model for each word in the generated caption, overlaid with its corresponding spatial attention map.

Qualitative Comparisons: Figure 4 shows improvements provided by each module. We find that the ST model captures more of action-related cues and provide caption verbs, while the TS model better captures the appearances predicting the right nouns. The STaTS module absorbs the benefits from both ST and TS, yielding the best video captioning. To back up these qualitative observations, Figure 5 provides more insight into how different attention modules affect the resulting caption. In the bar chart, we sort all the words from the generated captions for the testing set of MSR-VTT according to their frequency. First, we remove the top 5 most frequent words from each chart (such as "man" and "woman"). Each bar chart shows the top 20 verbs and nouns, from which it can be seen that the ST module generates more verbs (13 verbs out of 20) while the TS module generates more nouns (12 nouns out of 20). A similar phenomenon is shown in the adjacent pie charts, which indicate the total percentage of verbs, nouns, and adjectives in the generated captions. Notably, the ST model generates nearly 27% verbs (8% higher than the TS model), while the TS model generates 59% nouns (10% higher than the ST model), demonstrating their complementary nature. The combination of ST and TS, the STaTS module, provides a balance between the two. In Figure 6, we visualize an example of how STaTS attention is localized spatially and temporally in the sequence (more examples in the supplementary materials). The first two rows illustrate the sequence of events in the video. The third row visualizes the attention. For each word in the generated caption (fourth row), we chose the frame with highest temporal attention and overlaid the respective spatial attention.

Comparisons to the State of the Art: In Table 3, we show the results of our STaTS method with various feature combinations and compare it against state-of-the-art methods on the MSVD dataset. Our model fares better by more than 3.5% on the CIDEr and by 2% on BLEU4 than the next best method (RecNet [55]). In Table 4, we provide comparisons on the MSR-VTT dataset. We outperform several recently proposed methods. Specifically, we outperform RecNet on all four metrics, while outperforming more recent GRU-EVE [2] and OA-BTG [73] on most metrics. Note that these are powerful deep models that combine visual saliency with dynamics learning, and our results clearly demonstrate the superiority of our approach.

5. Conclusions

We proposed novel attention models for video caption generation combining spatio-temporal and temporo-spatial (STaTS) attention. We also presented ranked temporal pooling using an LSTM that emulates a rank-SVM. Our method can be seen as stage-wise attention, in which spatial and temporal cues are explored hierarchically. Our scheme yields state-of-the-art results on two benchmark datasets.

References

[1] N. Aafaq, N. Akhtar, W. Liu, S. Z. Gilani, and A. Mian. Spatio-temporal dynamics and semantic attribute enriched visual encoding for video captioning. In *CVPR*, 2019.

[2] N. Aafaq, N. Akhtar, W. Liu, S. Z. Gilani, and A. Mian. Spatio-temporal dynamics and semantic attribute enriched visual encoding for video captioning. In *CVPR*, 2019.

[3] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and VQA. In *CVPR*, 2018.

[4] Y. Aytar, C. Vondrick, and A. Torralba. Soundnet: Learning sound representations from unlabeled video. In *NIPS*, 2016.

[5] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[6] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *NIPS*, 2015.

[7] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould. Dynamic image networks for action recognition. In *CVPR*, 2016.

[8] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[9] D. L. Chen and W. B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *HLT-ACL*, 2011.

[10] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[11] Y. Chen, S. Wang, W. Zhang, and Q. Huang. Less is more: Picking informative frames for video captioning. In *ECCV*, 2018.

[12] A. Cherian, B. Fernando, M. Harandi, and S. Gould. Generalized rank pooling for activity recognition. In *CVPR*, 2017.

[13] P. Das, C. Xu, R. F. Doell, and J. J. Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *CVPR*, 2013.

[14] M. Denkowski and A. Lavie. METEOR universal: Language specific translation evaluation for any target language. In *Workshop on Statistical Machine Translation*, 2014.

[15] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *arXiv preprint arXiv:1804.03619*, 2018.

[16] C. Feichtenhofer, A. Pinz, and R. Wildes. Spatiotemporal residual networks for video action recognition. In *NIPS*, 2016.

[17] C. Feichtenhofer, A. Pinz, and R. P. Wildes. Spatiotemporal multiplier networks for video action recognition. In *CVPR*, 2017.

[18] B. Fernando, P. Anderson, M. Hutter, and S. Gould. Discriminative hierarchical rank pooling for activity recognition. In *CVPR*, 2016.

[19] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *ICCV*, 2015.

[20] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017.

[21] S. Gould, B. Fernando, A. Cherian, P. Anderson, R. S. Cruz, and E. Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, 2016.

[22] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, et al. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018.

[23] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko. Youtube2Text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*, 2013.

[24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

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

[26] C. Hori, T. Hori, T.-Y. Lee, Z. Zhang, B. Harsham, J. R. Hershey, T. K. Marks, and K. Sumi. Attention-based multimodal fusion for video description. In *ICCV*, 2017.

[27] T. Hospedales, S. Gong, and T. Xiang. A markov clustering topic model for mining behaviour in video. In *ICCV*, 2009.

[28] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *CVPR*, 2016.

[29] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[30] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[31] A. Kojima, T. Tamura, and K. Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *IJCV*, 50(2):171–184, 2002.

[32] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles. Dense-captioning events in videos. In *ICCV*, 2017.

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

[34] X. Li, B. Zhao, and X. Lu. MAM-RNN: multi-level attention model based rnn for video captioning. In *IJCAI*, 2017.

[35] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

[36] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy. Improved image captioning via policy gradient optimization of spider. In *ICCV*, 2017.

[37] Y. Liu, X. Li, and Z. Shi. Video captioning with listwise supervision. In *AAAI*, 2017.

[38] J. Lu, C. Xiong, D. Parikh, and R. Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*, 2017.

[39] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[40] Y. Peng, Y. Zhao, and J. Zhang. Two-stream collaborative learning with spatial-temporal attention for video classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[41] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[42] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

[43] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[44] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017.

[45] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating video content to natural language descriptions. In *ICCV*, 2013.

[46] Z. Shen, J. Li, Z. Su, M. Li, Y. Chen, Y.-G. Jiang, and X. Xue. Weakly supervised dense video captioning. In *CVPR*, 2017.

[47] H. T. Siegelmann and E. D. Sontag. On the computational power of neural nets. In *COLT*, 1992.

[48] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[49] J. Thomason, S. Venugopalan, S. Guadarrama, K. Saenko, and R. Mooney. Integrating language and vision to generate natural language descriptions of videos in the wild. In *ICCL*, 2014.

[50] Y. Tu, X. Zhang, B. Liu, and C. Yan. Video description with spatial-temporal attention. In *ACM on Multimedia Conference*, 2017.

[51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, 2017.

[52] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015.

[53] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence-video to text. In *ICCV*, 2015.

[54] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL-HLT*, 2015.

[55] B. Wang, L. Ma, W. Zhang, and W. Liu. Reconstruction network for vide ocaptioning. In *CVPR*, 2018.

[56] J. Wang, W. Wang, Y. Huang, L. Wang, and T. Tan. Multimodal memory modelling for video captioning. In *CVPR*, 2018.

[57] X. Wang, W. Chen, J. Wu, Y.-F. Wang, and W. Y. Wang. Video captioning via hierarchical reinforcement learning. In *CVPR*, 2018.

[58] X. Wang, W. Chen, J. Wu, Y.-F. Wang, and W. Y. Wang. Video captioning via hierarchical reinforcement learning. In *CVPR*, 2018.

[59] X. Wang, Y.-F. Wang, and W. Y. Wang. Watch, listen, and describe: Globally and locally aligned cross-modal attentions for video captioning. In *NAACL-HLT*, 2018.

[60] J. Wanke, A. Ulges, C. H. Lampert, and T. M. Breuel. Topic models for semantics-preserving video compression. In *ICMIR*, 2010.

[61] J. Xu, T. Mei, T. Yao, and Y. Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, 2016.

[62] J. Xu, T. Yao, Y. Zhang, and T. Mei. Learning multimodal attention lstm networks for video captioning. In *ACM Multimedia Conference*, 2017.

[63] W. Yang, J. Feng, G. Xie, J. Liu, Z. Guo, and S. Yan. Video super-resolution based on spatial-temporal recurrent residual networks. *CVIU*, 2017.

[64] Z. Yang, Y. Han, and Z. Wang. Catching the temporal regions-of-interest for video captioning. In *ACM on Multimedia Conference*, 2017.

[65] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *ICCV*, 2015.

[66] G. Yu and J. Yuan. Fast action proposals for human action detection and search. In *CVPR*, 2015.

[67] H. Yu and J. M. Siskind. Grounded language learning from video described with sentences. In *ACL*, 2013.

[68] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*, 2016.

[69] T. Yu, H. Gu, L. Wang, S. Xiang, and C. Pan. Cascaded temporal spatial features for video action recognition. In *ICIP*, 2017.

[70] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015.

[71] M. Zanfir, E. Marinoiu, and C. Sminchisescu. Spatio-temporal attention models for grounded video captioning. In *ACCV*, 2016.

[72] C. Zhang and Y. Tian. Automatic video description generation via lstm with joint two-stream encoding. In *ICPR*, 2016.

[73] J. Zhang and Y. Peng. Object-aware aggregation with bidirectional temporal graph for video captioning. In *CVPR*, 2019.

[74] H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, and A. Torralba. The sound of pixels. In *ECCV*, 2018.