This WACV 2022 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Co-Segmentation Aided Two-Stream Architecture for Video Captioning

Jayesh Vaidya, Arulkumar Subramaniam, Anurag Mittal
Department of Computer Science and Engineering,
Indian Institute of Technology Madras
{jvaidya, aruls, amittal}@cse.iitm.ac.in

Abstract

The goal of video captioning is to generate captions for a video by understanding visual and temporal cues. A general video captioning model consists of an Encoder-Decoder framework where Encoder generally captures the visual and temporal information while the decoder generates captions. Recent works have incorporated object-level information into the Encoder by a pretrained off-the-shelf object detector, significantly improving performance. However, using an object detector comes with the following downsides: 1) object detectors may not exhaustively capture all the object categories. 2) In a realistic setting, the performance may be influenced by the domain gap between the object-detector and the visual-captioning dataset. To remedy this, we argue that using an external object detector could be eliminated if the model is equipped with the capability of automatically finding salient regions. To achieve this, we propose a novel architecture that learns to attend to salient regions such as objects, persons automatically using a co-segmentation inspired attention module. Then, we utilize a novel salient region interaction module to promote information propagation between salient regions of adjacent frames. Further, we incorporate this salient region-level information into the model using knowledge distillation. We evaluate our model on two benchmark datasets MSR-VTT and MSVD, and show that our model achieves competitive performance without using any object detector.

1. Introduction

Video captioning task aims to generate human-understandable captions by understanding visual and temporal cues in the video. As we are witnessing an exponential increase in videos, this task assumes greater importance. Further, the ability of a machine/computer to generate text from a video has the potential to have an enormous impact on our day-to-day life. For example, a live sports game can have automatically generated live commentary, the machine-human interaction may become more natural by human-understandable text format, it can serve better in blind assistance and automatically creating video subtitles. However, this task requires understanding complex visual contents such as the spatiotemporal structure of events, the interactions between different objects, and then grammatically generating coherent text descriptions out of it. Further, the multi-modal nature of this task makes it even more challenging to map information from one modality (video) to another (text).

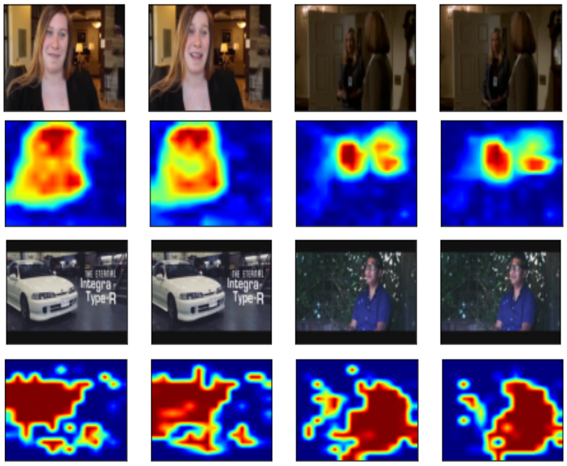


Figure 1: An illustration of spatial attention maps from proposed co-segmentation branch (CoSB) taken from the MSRVTT test dataset. The model is equipped with the capability to localize key salient regions (things and stuffs) necessitated by video-captioning task.

Classical methods used a template-based matching scheme by means of hand-crafted features and rules to determine the subject (S), verb (N), and Object (O). Then the sentences were generated using a sentence template [28, 2]. In these methods, nouns were detected predominantly by object detection methods [2]. However, these methods failed to show promising results on visuals involving complex scenes.

Recent advances in deep learning paved way for rejuvenated interest in solving this task. A typical deep learning pipeline for video captioning involves an encoder (2D [41, 19] and 3D CNNs [53, 3], transformer-based models

[44]) that takes the video frames as input and extracts features and a decoder (RNN [20, 14], Self-attention models [44]) to generate captions. Encoder features give global context in terms of spatial and temporal field of views of the video but fail to focus on local aspects and the relationship between them. For example, using a 2D CNN to extract a global descriptor for a frame does not account for different objects in the frame and therefore fails to model object-level interactions in the spatial and temporal domains.

To alleviate the issue mentioned above, some methods [59, 33, 60, 32, 62] not only focus on incorporating global descriptors but also consider the local aspects of the video overshadowed by the global features. They incorporate object-level features extracted from an off-the-shelf object detector such as YOLO [38], Faster RCNN [39], and Mask RCNN [18]. For instance, OA-BTG [59] extracts multiple object regions using an object detector and forms object trajectories by aligning the same object from different frames, thus capturing temporal dynamics of the scene. Zhou *et al.* [62] and Ma *et al.* [32] aggregate object features via pooling mechanism. One perceived downside of these approaches is the lack of object-to-object interactions modeling, as object interactions form a basis for the majority of the captions. To promote object-level interactions, some methods [13, 61, 55] use Graph convolutional networks (GCNs) [27] to form a spatiotemporal graph on object features. [33, 60] have shown that such an approach helps in boosting the performance.

Recent literature on video-captioning [32, 1, 58, 33, 60] task suggests that incorporating object-level information is important to achieve state-of-the-art performance. However, it needs an object detector pretrained on a large dataset. Though object detectors support video-captioning models with prior information on objects and their labels, we perceive the following downsides: 1) Object detectors may not exhaustively capture all the object categories. 2) The object detectors may bias the captioning model towards the object categories that it has been trained on. These categories are limited in numbers and can differ from the categories in the captioning dataset. Thus, the generalization may be affected, and performance may be capped by the performance of object detector in video captioning dataset, 3) Further, in a realistic setting, the performance may be influenced by the domain gap [5, 52] between the object detector's dataset and the visual-captioning dataset.

In this paper, we propose a novel method to demonstrate the possibility of incorporating object-level information without using off-the-shelf object detectors. Specifically, we attempt to equip the model with the capability to find salient regions in terms of co-segmentation inspired attention [42] between adjacent frames in an end-to-end manner. To this end, we propose a two-branch network as follows: 1) *Global scene branch (GSB)* for capturing features of the global scene directly by means of pretrained 2D and 3D CNNs, 2) *Co-segmentation branch (CoSB)* to retrieve features from salient local regions in the video frames. To achieve this, we incorporate a co-segmentation inspired attention [42] module to determine salient regions, conditioned on the cost volume of features between adjacent frames. This cost volume is created by the normalized cross-correlation between features from the adjacent frames. Once the salient regions are localized (Fig.1), to encourage temporal information propagation between these salient regions, we propose a novel Salient Region Interaction Module (SRIM) that captures region/object-level interactions in temporal dimension through multi-head self-attention [44] mechanism. GSB and CoSB are connected by knowledge-distillation inspired KL-divergence constraint that distills the knowledge of salient regions from CoSB to GSB. Our main contributions are as follows:

- We propose a two-branch architecture to capture global scene features as well as salient local regions automatically without using object detector.
- Through qualitative visualizations, we show that the co-segmentation branch is able to capture interpretable object-level information to aid video-captioning.
- We evaluate our model on two benchmark datasets MSR-VTT and MSVD, and show that our model achieves competitive performance without using any object detector.

2. Related work

This section describes the deep learning solutions on video-captioning and co-attention based architectures in the literature.

Video captioning: Earlier deep learning based methods [34, 46, 40, 47] follow a straightforward approach to extract 2D/3D CNNs features and use LSTMs to model the temporal relationship between frames and caption generation [20]. Yao *et al.* [56] used 3D CNNs to extract features and attention mechanism to learn long-range temporal dependencies, whereas [57] used hierarchical RNNs to model long term dependencies. Recent works started using object detectors to infuse knowledge about objects in video-captioning models. For instance, Zhang *et al.* [60] extracts object features from an object detector and uses GCN to merge object features. Pan *et al.* [33] uses a two-branch network where one branch extracts global features using 2D/3D CNNs and the other branch applies GCN to model interaction between object features. Further, object-related information is distilled into the global branch via a KL-divergence knowledge distillation. Our method is similar to [33], however, we do not use an object detector and show that it is able to achieve on par or better performance by making the model learn salient regions in an end-to-end manner.

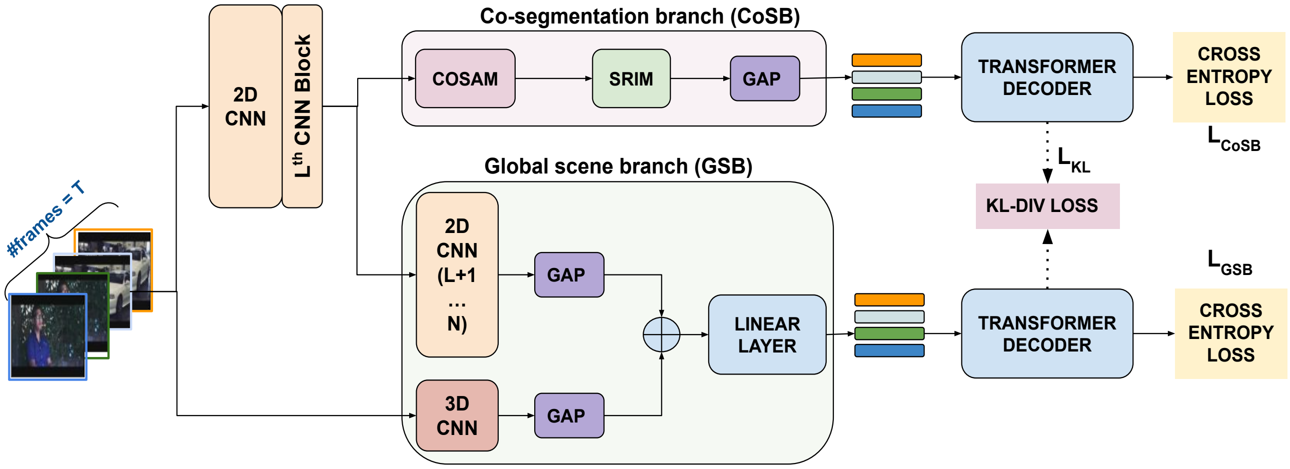


Figure 2: Illustration of the proposed model architecture. The model consists of two branch viz., 1) Global Scene branch (GSB) - Sec. 3.1, and 2) Co-segmentation branch (CoSB) - Sec. 3.2. Both branches have a separate Transformer as decoder (Sec. 3.3). GSB captures global scene by using 2D and 3D CNN global descriptors. CoSB attends to salient regions via COSAM module (Sec. 3.2.1) and promotes interaction among salient regions through SRIM (Sec. 3.2.2). Generated captions are evaluated using standard cross entropy loss. Further, the salient region information is distilled from CoSB to GSB via KL-divergence loss. Here, GAP = Global average pooling, $\bigoplus$ = feature concatenation.

Co-segmentation architectures: Co-segmentation is a task of localizing similar objects in one or more frames. Similarity can be based on several characteristics such as category, color, texture, and semantics. Earlier works relied on graph-based [4, 23, 29] or clustering [24, 43] approaches based on hand-crafted features to perform co-segmentation. Recent works employ deep networks for this task by means of spatial or channel-wise attention between frames. For instance, [7] proposed to use semantic similarity between regions to co-segment the images. [30] demonstrated the possibility of co-segmenting the images in an unsupervised way by constraining that foreground features to be similar to foreground features of other images and dissimilar their background features. [7] promoted common channel activations in bottleneck layer to aid co-segmentation. [42] repurposed the co-segmentation techniques to activate common regions between frames and shown that their model is able to learn salient regions for the underlying task. In this line of work, we attempt to eliminate the usage of pretrained object detector by studying the applicability of co-segmentation based attention [42]. Specifically, we propose to equip the network to find salient regions automatically, thus avoiding the need of an external object detection model.

3. Co-segmentation aided two-stream architecture

Recent state-of-the-arts predominantly use pretrained object detectors [38, 39, 18] to provide object-related cues to the visual captioning model. However, using object detectors has the following disadvantages: 1) Object detectors are trained with limited object categories, thus may not cover all the objects in captioning vocabulary. 2) The object detectors may bias the captioning model towards the object categories that it has been trained on. 3) Further, in realistic setting, the performance may be influenced by the domain gap [5, 52] between the object-detector's dataset and the visual-captioning dataset. In our work, we attempt to augment the model with end-to-end co-segmentation based learnable salient regions.

The overall model architecture is illustrated in Fig. 2. It follows a two-branch architecture: 1) *Global scene branch (GSB)* for capturing global scene cues, 2) *Co-segmentation branch (CoSB)* for capturing salient region features based on co-segmentation between frame-level spatial features. Each branch follows an encoder-decoder architecture. First, the input frames $\{F_i\}_{i=1}^{T}$ of a video V with dimension $T \times 3 \times H \times W$ are passed through GSB and CoSB simultaneously to predict the captions individually. Here F_i = i^{th} frame, T = number of frames, 3 channels belong to RGB, H = height, and W = width of the frames respectively. Next, we employ cross-entropy loss to evaluate the predicted captions from both the branches. Further, we employ a KL-divergence loss to impose a knowledge distillation constraint that requires both the branches to predict the same captions with similar confidence level. This ensures the knowledge of learned salient regions from CoSB is propagated to GSB. [33] follows a similar approach, however, they make use of an object-detection branch that requires a pretrained object detector. In the following sections, we describe the constituting components of the model.

3.1. Global scene branch (GSB)

The goal of GSB is to capture the global scene-level information to aid captioning task. To achieve this, we utilize

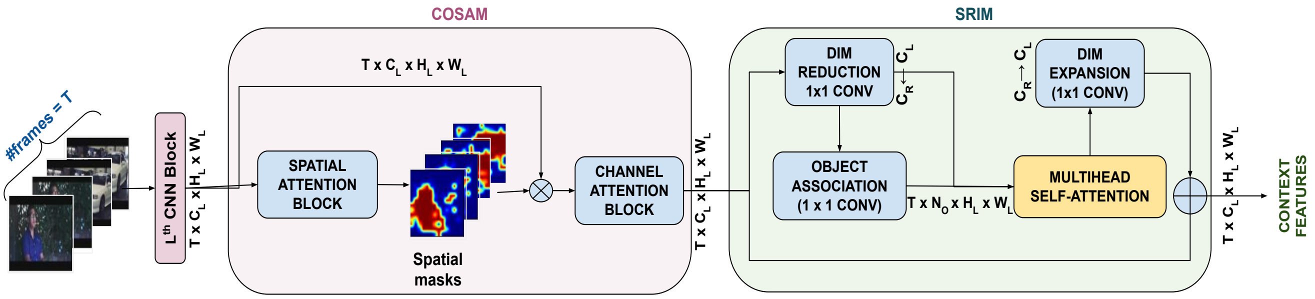


Figure 3: Co-Segmentation branch (CoSB) consists of two sub-modules COSAM and SRIM. 1) COSAM consists of two sub-blocks a) Spatial Attention Block (SAB) takes 2D-feature maps of dimension $T \times C_L \times H_L \times W_L$ as input an generates spatial masks signifying salient regions. b) Channel Attention Block (CAB) attends to common informative channels across the video frames. 2) Next, Salient Region Interaction Module (SRIM) promotes information propagation between salient regions via multi-head self-attention mechanism. Here, $\bigotimes$ = point-wise multiplication, $\bigoplus$ = feature addition.

an ImageNet [15] pretrained 2D CNN to capture frame-wise features and a Kinetics [25] pretrained 3D CNN to capture temporal features of the video. Specifically, given the input frames $\{F_i\}_{i=1}^{T}$ of dimension $T \times 3 \times H \times W$, we pass the frames through a 2D ResNet-101 [19] pretrained on ImageNet to obtain frame-wise features of dimension $T \times 2048$ after global-average pooling (GAP) layer. The input frames are also passed through a 3D ResNeXt-101 [53] pretrained on kinetics dataset to obtain temporal features of dimension $T \times 2048$ from its fully-connected layer before classification layer. These per-frame 2D and 3D features are concatenated together and are projected to 512 dimensions to result in features F_{GSB} of dimension $T \times 512$ for each video.

3.2. Co-segmentation branch (CoSB)

Global-scene information captured by GSB (Sec. 3.1) may not be sufficient for the task of video captioning which requires fine-grained understanding of the video. Hence, different from GSB, the goal of CoSB is to focus on salient spatial regions of the frames that serves the task better. To this end, we propose to employ a co-segmentation based attention module [42] to focus on salient regions. Specifically, CoSB branch (Fig.3) takes spatial feature maps of dimension $T \times C_L \times H_L \times W_L$ and uses a co-segmentation inspired attention (COSAM) [42] to capture salient regions of frames, followed by a salient-region interaction module (SRIM) to capture interaction between those regions. Here, C_L, H_L, W_L = number of channels, height, width of feature maps after L^{th} CNN block. In our work, we obtain these input feature maps from 2D ResNet-101 used in GSB to share weights and reduce model complexity. We will briefly mention the architecture of the COSAM and SRIM modules in the following paragraphs.

3.2.1 Co-segmentation inspired attention module (COSAM)

We hypothesize that the usage of object-level information derived from an external object-detector could be avoided if the model is equipped with the capability to capture the notion of objects / salient regions on its own in an end-to-end manner. Towards this end, in our work, we re-use the attention method formulated by [42] to activate common regions between input frames. [42] demonstrated that co-segmentation inspired attention (COSAM) aids the model to capture salient regions (typically, regions corresponding to objects). We adapt the COSAM module to be applied within adjacent frames to capture salient regions for the underlying task (i.e., video captioning).

COSAM consists of two consecutive attention blocks namely: 1) *Spatial attention block (COSAM-SAB)* to activate common spatial salient regions and suppress non-informative regions, 2) *Channel attention block (COSAM-CAB)* to activate common informative channels. The design of COSAM-SAB and COSAM-CAB follows the architecture proposed in [42]. Note that spatial and channel attention concepts have been used in image-based tasks [22, 51, 26] and relatively under-explored in video-based tasks. Specifically, the input feature maps of dimension $T \times C_L \times H_L \times W_L$ are passed through COSAM-SAB to generate a spatial attention mask of dimension $T \times 1 \times H_L \times W_L$. For each frame F_i, COSAM-SAB correlates every location's feature with its adjacent frame features (i.e., F_{i-1}, F_{i+1}) and creates a cost volume. A summary convolution layer takes the cost volume as input and produces a spatial attention mask. The spatial mask is multiplied with corresponding original features to obtain spatially refined features. Further, COSAM-CAB takes the spatially refined features as input, then through a GAP followed by an MLP, it produces channel attention weights ($T \times C_L$). These channel attention weights are multiplied with spatially refined features to output co-attended features of dimension $T \times C_L \times H_L \times W_L$.

3.2.2 Salient-region interaction module (SRIM)

The COSAM module may have selected the salient regions containing potentially multiple objects. To promote interactions between individual object-like regions, we propose to use a self-attention based salient region interaction module inspired from GloRe [11]. It takes the co-attended spatial feature maps of dimension $T \times C_L \times H_L \times W_L$ from previous step (COSAM) as input, and passes it through a 1×1 2D convolution layer to perform dimension reduction on channels ($C_L \rightarrow C_R$, $C_R << C_L$) to reduce computations. Next, we assume that every frame has N_o objects and each pixel may belong to one of these objects. To figure out the object association of pixels, the dimension reduced feature maps (dimension $T \times C_R \times H_L \times W_L$) are passed through an object-association module to output *object-association maps* $\{O_i\}_{i=1}^{T}$ that associates every pixel of feature map to one of N_o objects. We utilize an 1×1 2D convolution with N_o output channels to achieve this. Further, based on this object association map, we apply weighted average pooling (WAP) on the dimension-reduced feature maps to output object feature descriptors of size $T \times N_o \times C_R$. i.e., for each object, the model outputs a descriptor of size C_R. In the next step, we promote interactions between these object features using multi-head self-attention [44]. Each frame F_i's object features ($N_o \times C_R$) are restricted to interact with the previous, current and next frame's (F_{i-1}, F_i, F_{i+1}) object features to avoid noisy feature interactions (refer to Sec. 4.6 for ablation studies).

Next, these self-attended object features are distributed back to pixel-space to obtain context-aware features. To achieve this, the self-attended features are passed through a dimension expansion block ($C_R \rightarrow C_L$) consisting of an 1×1 2D convolution followed by a reverse mapping module based on object association maps $\{O_i\}_{i=1}^{T}$. These redistributed features (dimension $T \times C_L \times H_L \times W_L$) are added to the original input features (dimension $T \times C_L \times H_L \times W_L$) to yield context-aware features $\{F_{c_i}\}_{i=1}^{T}$. Next, we apply GAP on $\{F_{c_i}\}_{i=1}^{T}$ to get per-frame feature descriptors F_{CoSB} (dimension $T \times C_L$) to pass them to the decoder for caption generation.

3.3. Caption generation decoder

For every video instance, after obtaining its features from GSB (F_{GSB}) and CoSB (F_{CoSB}), we use transformer based decoders to generate captions. Specifically, we use two separate decoders similar to [33]: one to predict captions from F_{GSB} and the other to predict captions from F_{CoSB}. The decoder transformer generates a word at every time-step by attending to the input features as well as previously generated words in the caption. The predicted caption from the decoders is evaluated using cross-entropy loss. Further, to transfer salient region specific learning from CoSB to GSB, we impose a knowledge distillation based constraint that both CoSB and GSB shall generate the caption words with similar confidence (refer Sec 3.4 for objective functions). During test time, we use the captions from Global scene branch for evaluation, as followed in [33].

3.4. Objective functions

We use standard cross entropy loss for evaluating captions generated from the transformer decoders. Further, to promote knowledge propagation of salient regions from CoSB to GSB, we apply online knowledge distillation constraint that GSB and CoSB shall predict same words with similar confidence. Specifically, knowledge distillation constraint is imposed via KL-divergence loss similar to [33]. Given the word vocabulary W and probabilities of caption words generated from global-scene branch P, co-segmentation branch Q, the knowledge distillation constraint is formulated using KL-divergence as follows:

$$D_{KL}(\, P||\, Q) \;= \sum_{x \in W} P(x) \log\left(\frac{P(x)}{Q(x)}\right) \qquad (1)$$

The overall loss function is given by,

$$L = L_{GSB} + \lambda L_{CoSB} + \lambda_{KL} L_{KL} \qquad (2)$$

Here, L_{GSB} and L_{CoSB} are the individual cross entropy losses for GSB and CoSB branches respectively. L_{KL} denotes KL-divergence between probability distributions of GSB and CoSB branch. λ and λ_{KL} are hyper-parameters.

4. Experiments

In this section, we describe the training and test details, experiments and comparison of our model with state-of-the-arts.

4.1. Datasets

We use two challenging benchmark datasets: Microsoft Research Video-to-Text (MSR-VTT) [54] and Microsoft Video-Description Corpus (MSVD) [6].

MSTVTT: It contains 10,000 video clips from 20 wide-range of categories (cooking, animations, sports, etc) with an average video length of 20 seconds. Each video has 20 human-annotated English captions. We follow the standard split followed in [34, 37, 59] for train, validation and test as follows: 6513 video clips for training set, 497 for cross-validation set and remaining 2990 for the test set.

MSVD: It contains total of 1970 videos with approx. 40 English captions per video. We follow standard protocol split [34, 37, 59] of dividing 1970 videos into train = 1200, validation = 100, and test = 670 videos for performing experiments.

4.2. Evaluation metrics

We use the following metrics found in literature to quantitatively evaluate our models performance: BLEU@4 [35], METEOR [16], ROUGE_L [31] and CIDEr [45].

BLEU@N matches n-gram between the generated captions and the ground truth, while METEOR metric is based on word-to-word matching between the generated captions and the ground truth, ROUGE_L is Longest common subsequence based metric (LCS) and CIDEr calculates n-gram similarity between generated caption and ground truth. The above mentioned metrics are computed using Microsoft COCO evaluation server [10]

4.3. Implementation Details

We implement our model using PyTorch [36] deep learning framework[1]. During training, for every video, we use uniformly subsampled $T = 10$ frames as input with correct temporal order. The frames are resized to spatial resolution of 256×256 and are center cropped to 224×224. We use the 2D CNN ResNet-101 pretrained on ImageNet [15] dataset to extract 2D features in GSB. Also, we extract intermediate features after 5^{th} block (dimension $T \times 1024 \times 14 \times 14$) as input features to CoSB branch. For acquiring 3D features in GSB, we use 3D ResNeXt-101 pretrained on Kinetics dataset. For every frame, a clip of 16 adjacent frames is used as input for ResNeXt-101 to get per-frame 3D feature. The following hyper-parameters are used in network design: $C_R = 512, N_o = 5$.

For the transformer decoder, we adopt [9] as our transformer architecture. We build a vocabulary of size 11K and 6.5K for MSR-VTT and MSVD respectively. During this process, punctuations are removed from every sentence and a word is removed if its count is one or less in whole dataset to avoid imbalanced distributions. During training, the maximum length of the sentence is set to 20. We use 2 multi-head self attention layers (one to encode visual features and the other to generate captions) in the decoder. We greedly output words with beam size of 1. Self-attention module has 8 attention heads and an MLP to output features with 1024 dimension.

We use a cross validation set to tune the hyper-parameters. The hyper-parameters are set as follows: batch size = 64, learning rate = 10^{-4}, the learning rate is decayed by multiplying 0.8 after every 200 epochs, weight decay = 5×10^{-4}, dropout probability = 0.3. The model is optimized using Adam optimizer with default parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$) for 650 epochs with early stopping to get the best performing model. For MSRVTT dataset, we set $\lambda = 1$ and for MSVD dataset, $\lambda = 2$. λ_{KL} is set to 4 for both datasets.

4.4. Comparison with state-of-the-art methods

We compare our model with recent state-of-the-art methods: SA-LSTM [49], M3 [50] RecNet [49], PickNet [12], MARN [37], POS+VCT [21], ORG-TRL [60], OA-BGT [59], STG-KD [33].

SA-LSTM [56] captures global temporal structure using temporal attention mechanism. M3 [50] creates multimodal memory space that stores and retrieves both visual and textual information, to get mapping between words and visuals. RecNet [49] follows an encoder-decoder architecture along with a reconstruction module to generate visual features for the generated captions. PickNet [12] uses reinforcement Learning methods to pick video frames that are more visually diverse to get non-redundant features. MARN [37] establishes a mapping between words and visual cues related to that word over the spectrum of training data to get more robust context. OA-BTG [59] and STG-KD [33] introduce object features in the model with the help of a pretrained external object detector. Specifically, OA-BTG [59] aligns similar objects and computes object trajectories, whereas STG-KD [33] finds object level interactions using a spatiotemporal graph and further distilling this knowledge into their encoder-decoder network. Our model differs from STG-KD [33] by eliminating the need for external object detector and relying on model's ability to focus on salient regions. POS+VCT [21] and ORG-TRL [60] focus to improve the decoder. Specifically, POS+VCT uses <*POS*> tags to learn syntactic structure and uses a combination of features such as Inception-ResNetV2 with C3D, where as ORG-TRL [60] uses pretrained BERT [17] model to generate better distribution of vocabulary while predicting words in caption. Table 1 shows the comparison of our model with the state-of-the-art methods.

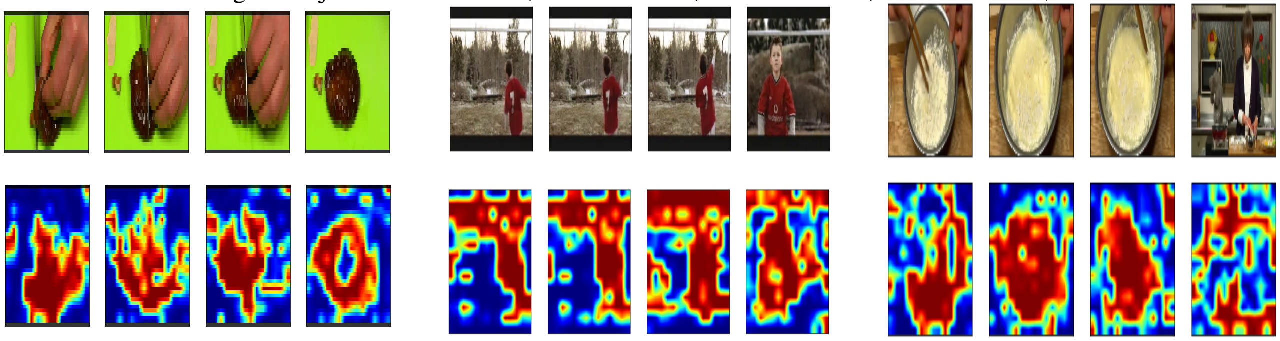
Table 1 is split into three logical groups that signifies different line of works in the video-captioning literature. First group (POS+VCT [21], ORG-TRL [60]) focuses on improving decoder in their video captioning model. As these models directly optimize decoder with <*POS*> tags and pretrained BERT [17] modules, according to the fair comparison policy followed in the literature [33], we compare our method only with those methods that worked on encoder part (i.e., logical group 3). The second group (OA-BTG [59], STG-KD [33]) employ a pretrained object detector to introduce the concept of objects inside the model. The third group (RecNet [49], PickNet [12], MARN [37], SA-LSTM [49], M3 [50]) focuses on encoders without the help of external pretrained models like object detectors / optical flow estimators. Our model belongs to third category where we focus on getting a better encoder representation without using external pretrained models and dependencies. In that, our model is most comparable with MARN, in which we use same global feature extractors as MARN i.e. ResNet-101 [19] and ResNeXt-101 [53]. Note that we follow the standard procedure in the literature [37] to not compare with models based on reinforcement Learning (RL) [48] techniques.

From the Table 1, in MSRVTT dataset, we can see that our model gets competitive performance in B@4,

[1] Code will be released in Github upon acceptance of the paper.

Methods	Year	Object detector	MSRVTT				MSVD			
			B	M	R	C	B	M	R	C
POS+VCT [21]	ICCV-2019	-	42.3	**29.7**	**62.8**	49.1	52.8	36.1	71.8	87.8
ORG-TRL [60]	CVPR-2020	Faster RCNN	**43.6**	28.8	62.1	**50.9**	**54.3**	**36.4**	**73.9**	**95.2**
OA-BTG [59]	CVPR-2019	Mask RCNN	**41.4**	28.2	-	46.9	**56.9**	36.2	-	90.6
STG-KD [33]	CVPR-2020	Faster RCNN	40.5	**28.3**	**60.9**	**47.1**	52.2	**36.9**	**73.9**	**93.0**
SA-LSTM [49]	CVPR-2018	-	36.3	25.5	58.3	39.9	45.3	31.9	64.2	76.2
M3 [50]	CVPR-2018	-	38.1	26.6	-	-	52.8	33.3	-	-
RecNet [49]	CVPR-2018	-	39.1	26.6	59.3	42.7	**52.3**	34.1	69.8	80.3
PickNet [12]	ECCV-2018	-	41.3	27.7	59.8	44.1	**52.3**	33.3	69.6	76.5
MARN [37]	CVPR-2019	-	40.4	**28.1**	60.7	**47.1**	48.6	35.1	71.9	92.2
Ours	-	-	**41.4**	27.8	**61.0**	46.5	50.7	**35.3**	**72.1**	**97.8**

Table 1: Performance analysis on MSRVTT and MSVD datasets. First group of methods optimize on decoder. Second group of methods enhance encoder visual features by making use of object detectors. Third group of methods enhance encoder features without the usage of object detectors. Here, B = BLEU@4, M = METEOR, R = ROUGE_L, C = CIDEr.

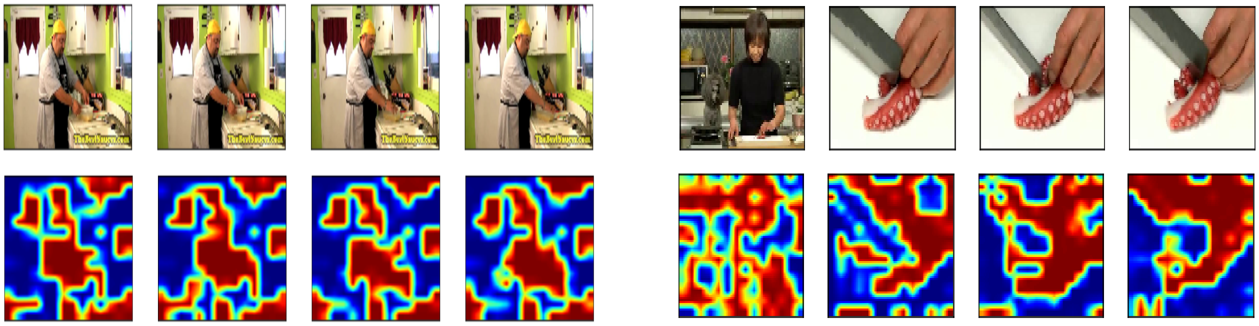


(a) **STG-KD [33]:** A woman is cooking.
Ours: A person is cutting a piece of meat.
GT: A person is cutting mushroom.

(b) **STG-KD [33]:** A boy kicks a goal.
Ours: A boy is kicking a soccer ball.
GT: A boy kicks a soccer ball.

(c) **STG-KD [33]:** A person is cooking.
Ours: A woman is mixing ingredients.
GT: A woman is mixing water and flour.

Figure 4: Qualitative visualization of CoSB branch's COSAM spatial attention on MSVD dataset (best viewed in color)



(a) **STG-KD [33]:** A man is pouring pasta on to a container.
Ours: A man is putting a lid on a plastic container.
GT: A man puts a lid on a plastic container.

(b) **STG-KD [33]:** A woman is slicing carrot.
Ours: A woman is slicing octopus.
GT: A woman is slicing octopus.

Figure 5: Qualitative visualization of CoSB branch's COSAM spatial attention on MSVD dataset (best viewed in color)

ROUGE_L measures. Specifically, our model performs 1 and 0.3 points better than its most comparable method MARN in B@4 and ROUGE_L respectively. Further, our model performs 1.2 and 2.4 points over PickNet [12] in ROUGE_L and CIDEr metrics respectively. We hypothesize that the moderate performance of our model in MSRVTT is due to the high number of discontinuous clips in these videos, thus it may be difficult for COSAM to determine salient regions.

In MSVD dataset, our model outperforms on 3 metrics in the third logical group: METEOR, ROUGE_L and CIDEr showing the effectiveness of combination of GSB and CoSB branches. We achieve state-of-the-art CIDEr score outperforming even the models that use external object detectors

[8] and those that focuses on both encoder and decoder [60]. CIDEr being a metric that is better correlated with human judgement, this performance improvement shows that our model is able to generalize to test set and generate better captions.

Components			B@4	M	R	C
GSB	CoSB					
	COSAM	SRIM				
✓	✗	✗	48.0	34.5	71.2	93.3
✗	✓	✓	38.3	28.3	64.7	49.1
✓	✓	✗	50.4	34.4	71.7	92.7
✓	✗	✓	49.2	35.2	71.9	92.7
✓	✓	✓	49.4	35.0	71.5	96.9
✓	✓	✓(m)	**50.7**	**35.3**	**72.1**	**97.8**

Table 2: Ablation study of the proposed model on MSVD dataset. Here, B@4 = BLUE@4, M = METEOR, R = ROUGE_L, C = CIDEr. 'm' in SRIM column indicates masked multihead self-attention.

4.5. Qualitative Analysis

In this section, we illustrate some of the interesting qualitative visualization from our model showing its ability to localize the important salient regions to aid video captioning. As our model is similar to [33][2] but without using external object detector, we also provide captions from [33] in the figures to have a relative comparison. In the qualitative visualizations (Fig. 4a, 4b, 4c, 5a and 5b), we show the visualizations of the spatial mask from COSAM module of CoSB branch. From the visualizations, it is evident from the masks that our model is able to attend key salient regions such as objects, persons in the video. Specifically, in Fig. 4a, our model attempts to generate more detailed caption by adding a quantifier like "a piece of". Similarly, in Fig. 4c, "mixing ingredients" is more detailing about the particular action than "cooking". In Fig. 5a, our model is able to capture the "closing" action between the person and the box lid, as well as it is able to generate a caption giving specific details about "plastic container". Further, in Fig. 4b, our model is able to attend on the cross bar of the goal post which is an important cue to showcase that the boy is hitting a "soccer ball". We notice a key observation that our model is better at classifying/recognizing objects that's important for captioning tasks. For example, Fig. 5b shows that our model is correctly able to classify "octopus", where as [33] classifies it as "carrot". We hypothesize that object detectors may face difficulty in detecting and recognizing such uncommon classes that's not present in their training dataset. Hence, letting the model to figure out the salient regions without biasing it with an external object detector is a potent approach.

4.6. Ablation studies

In this section, we show performance of our model by ablating on different components, especially the significance of the CoSB branch. The ablation study results are shown in the Table 2. We perform six experiments to isolate and analyze the contributions from different components of our model. In first experiment (Row 1), we first start with a simple model by having only the GSB with the cross entropy loss and show the performance. In the next experiment (Row 2), we train only CoSB with cross entropy loss which performs inferior to GSB-only model. It may be due to the use of only 2D intermediate features without temporal context. As GSB takes input as both 2D and 3D CNN features, it receives both spatial and temporal cues. Further, from the third experiment, along with GSB, we start integrating the components of CoSB one by one. Row 3 shows the performance of our model with GSB and COSAM. This variant improves the GSB-only performance by 2.4, 0.5 points in B@4 and ROUGE_L metrics respectively. Similarly (Row 4) we train GSB and SRIM to showcase relevance of spatial interactions, the results are similar to Row 3 with improvements in METEOR and ROUGE_L. Row 5 shows the performance of the model with GSB+COSAM+SRIM. It can be seen that this combination improves CIDEr metric significantly by 3.6 points over GSB-only model. In the next experiment (Row 6), we modify SRIM to restrict interaction of salient-region features (i.e., use a mask in multi-head self-attention) to be only within adjacent frames. i.e. every salient region interacts with salient regions from only the adjacent frames. We observe that this masked multi-head self attention improves the model's performance, Specifically, it improves 1.3, 0.9 points on B@4 and CIDEr metrics over the model where SRIM has all-to-all self attention. We hypothesize that constraining salient region interaction only to adjacent frames may avoid noisy feature interaction between frames with higher time gap, thus improving the performance.

5. Conclusion

In this paper, we proposed a unified framework to combine global scene features along with co-segmentation based salient regions and demonstrated that it performs competitive to the state-of-the-art methods that make use of external pretrained object detectors. Further, by means of qualitative visualization, it is shown that the co-segmentation is indeed able to capture salient regions / objects that's necessary for video captioning. Through a set of ablation studies, we quantified the contributions of individual components of our model and shown that co-segmentation based salient regions and salient region interaction module add value to the video captioning pipeline.

Acknowledgements: This work is supported by grants from PM's fellowship for Doctoral Research (SERB, India) & Google PhD Fellowship to Arulkumar Subramaniam.

[2] We reproduce and train the model on our own for producing these results.

References

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

[2] Andrei Barbu, Alexander Bridge, Zachary Burchill, Dan Coroian, Sven Dickinson, Sanja Fidler, Aaron Michaux, Sam Mussman, Siddharth Narayanaswamy, Dhaval Salvi, et al. Video in sentences out. *arXiv preprint arXiv:1204.2742*, 2012.

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

[4] Haw-Shiuan Chang and Yu-Chiang Frank Wang. Optimizing the decomposition for multiple foreground cosegmentation. *Computer Vision and Image Understanding*, 141:18–27, 2015.

[5] Chaoqi Chen, Zebiao Zheng, Xinghao Ding, Yue Huang, and Qi Dou. Harmonizing transferability and discriminability for adapting object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8869–8878, 2020.

[6] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 190–200, 2011.

[7] Hong Chen, Yifei Huang, and Hideki Nakayama. Semantic aware attention based deep object co-segmentation. *arXiv preprint arXiv:1810.06859*, 2018.

[8] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[9] Ming Chen, Yingming Li, Zhongfei Zhang, and Siyu Huang. Tvt: Two-view transformer network for video captioning. In *Asian Conference on Machine Learning*, pages 847–862. PMLR, 2018.

[10] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[11] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 433–442, 2019.

[12] Yangyu Chen, Shuhui Wang, Weigang Zhang, and Qingming Huang. Less is more: Picking informative frames for video captioning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 358–373, 2018.

[13] Yu Cheng, Bo Wang, Bo Yang, and Robby T Tan. Graph and temporal convolutional networks for 3d multi-person pose estimation in monocular videos. *arXiv preprint arXiv:2012.11806*, 2020.

[14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[16] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380, 2014.

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

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

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

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

[21] Jingyi Hou, Xinxiao Wu, Wentian Zhao, Jiebo Luo, and Yunde Jia. Joint syntax representation learning and visual cue translation for video captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8918–8927, 2019.

[22] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[23] Koteswar Rao Jerripothula, Jianfei Cai, and Junsong Yuan. Image co-segmentation via saliency co-fusion. *IEEE Transactions on Multimedia*, 18(9):1896–1909, 2016.

[24] Armand Joulin, Francis Bach, and Jean Ponce. Multi-class cosegmentation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 542–549. IEEE, 2012.

[25] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[26] Lei Ke, Yu-Wing Tai, and Chi-Keung Tang. Occlusion-aware video object inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14468–14478, 2021.

[27] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[28] Atsuhiro Kojima, Takeshi Tamura, and Kunio Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *International Journal of Computer Vision*, 50(2):171–184, 2002.

[29] Lina Li, Zhi Liu, and Jian Zhang. Unsupervised image co-segmentation via guidance of simple images. *Neurocomputing*, 275:1650–1661, 2018.

[30] Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Deep object co-segmentation. *arXiv preprint arXiv:1804.06423*, 2018.

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

[32] Chih-Yao Ma, Asim Kadav, Iain Melvin, Zsolt Kira, Ghassan AlRegib, and Hans Peter Graf. Attend and interact: Higher-order object interactions for video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6790–6800, 2018.

[33] Boxiao Pan, Haoye Cai, De-An Huang, Kuan-Hui Lee, Adrien Gaidon, Ehsan Adeli, and Juan Carlos Niebles. Spatio-temporal graph for video captioning with knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10870–10879, 2020.

[34] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4594–4602, 2016.

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

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

[37] Wenjie Pei, Jiyuan Zhang, Xiangrong Wang, Lei Ke, Xiaoyong Shen, and Yu-Wing Tai. Memory-attended recurrent network for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8347–8356, 2019.

[38] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[40] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *International Journal of Computer Vision*, 123(1):94–120, 2017.

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

[42] Arulkumar Subramaniam, Athira Nambiar, and Anurag Mittal. Co-segmentation inspired attention networks for video-based person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 562–572, 2019.

[43] Zhiqiang Tao, Hongfu Liu, Huazhu Fu, and Yun Fu. Image cosegmentation via saliency-guided constrained clustering with cosine similarity. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

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

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

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

[47] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014.

[48] Bairui Wang, Lin Ma, Wei Zhang, Wenhao Jiang, Jingwen Wang, and Wei Liu. Controllable video captioning with pos sequence guidance based on gated fusion network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2641–2650, 2019.

[49] Bairui Wang, Lin Ma, Wei Zhang, and Wei Liu. Reconstruction network for video captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7622–7631, 2018.

[50] Junbo Wang, Wei Wang, Yan Huang, Liang Wang, and Tieniu Tan. M3: Multimodal memory modelling for video captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7512–7520, 2018.

[51] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[52] Aming Wu, Yahong Han, Linchao Zhu, and Yi Yang. Instance-invariant domain adaptive object detection via progressive disentanglement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[53] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[54] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[55] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[56] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. De-

scribing videos by exploiting temporal structure. In *Proceedings of the IEEE international conference on computer vision*, pages 4507–4515, 2015.

[57] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4584–4593, 2016.

[58] Mihai Zanfir, Elisabeta Marinoiu, and Cristian Sminchisescu. Spatio-temporal attention models for grounded video captioning. In *asian conference on computer vision*, pages 104–119. Springer, 2016.

[59] Junchao Zhang and Yuxin Peng. Object-aware aggregation with bidirectional temporal graph for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8327–8336, 2019.

[60] Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zheng-Jun Zha. Object relational graph with teacher-recommended learning for video captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13278–13288, 2020.

[61] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3425–3435, 2019.

[62] Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. Grounded video description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6578–6587, 2019.