Spatio-Temporal Graph for Video Captioning with Knowledge Distillation

Boxiao Pan[1], Haoye Cai[1], De-An Huang[1],
Kuan-Hui Lee[2], Adrien Gaidon[2], Ehsan Adeli[1], Juan Carlos Niebles[1]
[1]Stanford University [2]Toyota Research Institute
{bxpan,hcaiaa,dahuang,eadeli,jniebles}@cs.stanford.edu {kuan.lee,adrien.gaidon}@tri.global

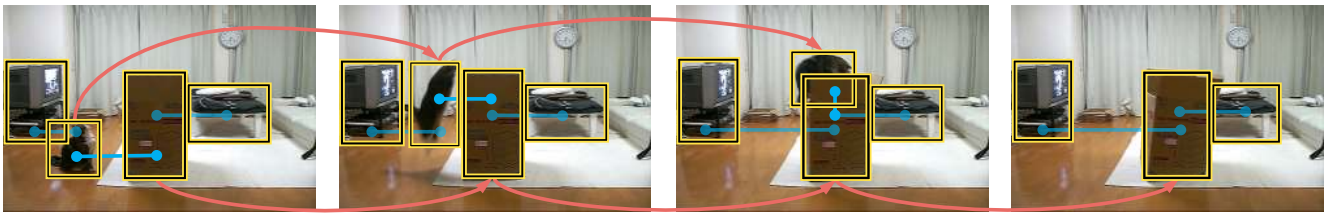

Figure 1: How to understand and describe a scene from video input? We argue that a detailed understanding of spatio-temporal object interaction is crucial for this task. In this paper, we propose a spatio-temporal graph model to explicitly capture such information for video captioning. Yellow boxes represent object proposals from Faster R-CNN [12]. Red arrows denote directed temporal edges (for clarity, only the most relevant ones are shown), while blue lines indicate undirected spatial connections. Video sample from MSVD [3] with the caption "A cat jumps into a box." Best viewed in color.

Abstract

Video captioning is a challenging task that requires a deep understanding of visual scenes. State-of-the-art methods generate captions using either scene-level or object-level information but without explicitly modeling object interactions. Thus, they often fail to make visually grounded predictions, and are sensitive to spurious correlations. In this paper, we propose a novel spatio-temporal graph model for video captioning that exploits object interactions in space and time. Our model builds interpretable links and is able to provide explicit visual grounding. To avoid unstable performance caused by the variable number of objects, we further propose an object-aware knowledge distillation mechanism, in which local object information is used to regularize global scene features. We demonstrate the efficacy of our approach through extensive experiments on two benchmarks, showing our approach yields competitive performance with interpretable predictions.

1. Introduction

Scenes are complicated, not only because of the diverse set of entities involved, but also the complex interactions among them. Consider the scene shown in Fig. 1. In order to understand that "A cat jumps into a box," we need to first identify "cat" and "box," then capture the transformation of "cat jumps into the box." It is also crucial to be able to ignore the "television" and "bed," since they mostly serve as distractors for understanding what is happening.

The task of video captioning [13, 37] approaches scene understanding by generating text descriptions from video input. However, current methods for video captioning are not able to capture these interactions. Rather than modeling the correlations among high-level semantic entities, current methods build connections directly on raw pixels and rely on the hierarchical deep neural network structure to capture higher-level relationships [19, 39]. Some works try operating on object features instead, but they either ignore cross-object interaction [49], or object transformation over time [27, 51]. Despite efforts in directly modeling local object features, the connections among them are not interpretable [27, 51], and hence sensitive to spurious correlations.

On the other hand, modeling object relations via video spatio-temporal graphs [34, 43] has been explored to explicitly construct links between high-level entities by leveraging the relation-modeling nature of graphs. Specifically, nodes represent these entities, such as body joints [47], objects / persons [8, 43, 45], and actions [34], while edges encode relationships among the entities. Although spatio-temporal

graph models have achieved great success on classification tasks [8, 17, 43, 45], the effect of relation modeling remains unclear, as the model would easily shortcut the classification problem by taking advantage of other cues (e.g., background). To the best of our knowledge, we are the first to explicitly model spatio-temporal object relationships for video captioning, and show the effect of graphical modeling through extensive experiments.

To provide the global context that is missing from local object features, previous work either merges them to another global scene branch through feature concatenation [43] or pooling [49], or adds scene features as a separate node in the graph [8, 11, 34]. However, because videos contain a variable number of objects, the learned object representation is often noisy. It thus leads to suboptimal performance. To solve this problem, we introduce a two-branch network structure, where an object branch captures object interaction as privileged information, and then injects it into a scene branch by performing knowledge distillation [18] between their language logits. Compared with previous approaches that impose hard constraints on features, our proposed method applies soft regularization on logits, which thus makes the learned features more robust. We refer to this mechanism as "object-aware knowledge distillation." During testing, only the scene branch is used, which leverages the distilled features with object information already embedded. As a bonus effect, this approach is also able to save the cost of running object detection at test time.

In this paper, we propose a novel way to tackle video captioning by exploiting the spatio-temporal interaction and transformation of objects. Specifically, we first represent the input video as a spatio-temporal graph, where nodes represent objects and edges measure correlations among them. In order to build interpretable and meaningful connections, we design the adjacency matrices to explicitly incorporate prior knowledge on the spatial layout as well as the temporal transformation. Subsequently, we perform graph convolution [22] to update the graph representation. This updated representation is then injected into another scene branch, where we directly model the global frame sequences, as privileged object information via the proposed object-aware knowledge distillation mechanism. Afterward, language decoding is performed through a Transformer network [35] to obtain the final text description. We conduct experiments on two challenging video captioning datasets, namely MSR-VTT [46] and MSVD [3]. Our model demonstrates significant improvement over state-of-the-art approaches across multiple evaluation metrics on MSVD and competitive results on MSR-VTT. Note that although our proposed model is agnostic to downstream tasks, we only focus on video captioning in this work. Its application on other domains is thus left as future work.

In summary, our **main contributions** are as follows. (1) We design a **novel spatio-temporal graph network** to perform video captioning by exploiting object interactions. To the best of our knowledge, this is the first time that spatio-temporal object interaction is explicitly leveraged for video captioning and in an interpretable manner. (2) We propose an **object-aware knowledge distillation mechanism** to solve the problem of noisy feature learning that exists in previous spatio-temporal graph models. Experimental results show that our approach achieves a significant boost over the state-of-the-art on MSVD [3] and competitive results on MSR-VTT [46].

2. Related Work

General Video Classification. Spatio-temporal reasoning is one of the main topics for video understanding. With the success of deep Convolutional Neural Networks (CNNs) on image recognition [24], many deep architectures have been proposed correspondingly in the space-time domain. C3D [33] and I3D [2] construct hierarchical spatio-temporal understanding by performing 3D convolution. The two-stream network [10] receives additional motion information by fusing an extra optical flow branch. TSN [41], on the other hand, takes advantage of the fact that huge redundancy exists between adjacent video frames via sparse frame sampling. While arguing that previous methods fail to capture long-term dependency, several recent works [9, 42, 44, 50] attempt to model a wider temporal range. Specifically, TRN [50] extends TSN by considering multi-level sampling frequency. The non-local network [42] explicitly creates long-term spatio-temporal links among features. The SlowFast network [9] exploits multiple time scales by creating two pathways with different temporal resolutions. Alternatively, the long-term feature bank [44] directly stores long-term features and later correlates them with short-term features. However, all these models directly reason over raw pixels, which often fail to ground their predictions to visual evidence by simply collecting data bias. In contrast, we propose to model relationships over higher-level entities, which in our case, are the objects within scenes.

Spatio-Temporal Graphs. While the idea of graphical scene representation has been explored extensively in the image domain [20, 23, 48], its extension to videos has only been recently attracting attention. Among the earlier attempts, ST-GCN [47] models human body joint coordinates to perform action classification. Later works directly model the objects in a scene. The resulting representation is then used to perform various down-stream tasks, such as action classification [17, 43, 45], action localization [11, 28], relation prediction [34], and gaze prediction [8]. All these works aim for simple classification or localization tasks where capturing object interactions might not be as important. Thus the effect of spatio-temporal graph remains unclear. In this work, we target at the much harder task

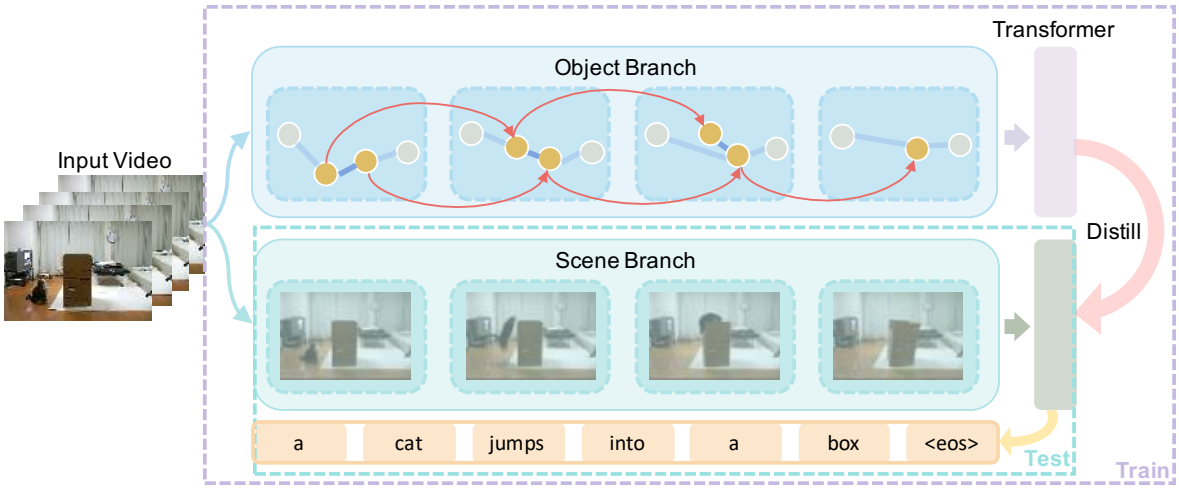

Figure 2: Overview of the proposed two-branch framework. During training, the object branch captures space-time object interaction information via the proposed spatio-temporal graph model, while the scene branch provides the global context absent from the object branch. The object-level information is then distilled into the scene feature representation by aligning language logits from the two branches. For clarity, we drop the arrow from the object branch Transformer to the output sentence, but it is also trained using a language loss. At test time, only the scene branch is needed for sentence generation.

of video captioning, and show the efficacy of our graph-based approach through extensive experiments and ablation study. While previous methods suffer from the noisy feature learning problem, we solve it via the proposed object-aware knowledge distillation mechanism.

Knowledge Distillation. Knowledge distillation was first proposed in [18], where the distillation is performed from a large model to a small one by minimizing the KL divergence between their logits distributions. Later, Lopez-Paz *et al*. [26] generalize distillation to incorporate privileged information, which is some additional information that is available during training but not accessible during testing. One application of this approach is to treat the extra modality as the privileged information [14]. In our case, we innovatively regard object interactions as the privileged information. We leverage such information during training by distilling it into the scene branch, while only the scene branch is executed during testing.

Video Captioning. Earlier work on video captioning mainly focus on template-based language models [13, 31, 32]. Motivated by the success of the encoder-decoder architecture, Venugopalan *et al*. [38] extend it to the field of video captioning by globally pooling all frame features. The following works then try to exploit temporal patterns by introducing attention mechanisms [6, 37]. Very recently, Pei *et al*. [30] propose MARN, which attends to all semantically similar videos when generating descriptions for a single video. Wang *et al*. [39] and Hou *et al*. [19] provide the idea of predicting POS information before the actual sentence. While Recurrent Neural Networks (RNNs) are adopted as the language decoder for most of the models, Transformer [35] has been shown to be powerful as well [4, 51, 52]. Because it is faster and easier to train, we employ Transformer as the language decoder in our model.

Although most of the prior work directly operates on the global frames or video features, there have been a few attempts that try to model local object features. Zhou *et al*. [51] and Ma *et al*. [27] both use spatial pooling to aggregate object features. Zhang *et al*. [49] propose to perform object tracking and model object trajectories using GRU. However, they either ignore the temporal [27, 51] or the spatial [49] object interactions. We instead model both spatial and temporal object interactions jointly via our proposed spatio-temporal graph. Moreover, our approach is able to incorporate prior knowledge into the adjacency matrix, which provides better interpretability than the fully learned attention mechanism.

3. Method

An overview of our proposed two-branch network architecture is illustrated in Fig. 2. During the training process, given a video that depicts a dynamic scene, our goal is to condense it into a representation that fully captures the spatio-temporal object interaction. This is done via the proposed spatio-temporal graph network, which serves as the object branch. Afterward, this interaction information is distilled into another scene branch via the object-aware knowledge distillation mechanism. At test time, only the scene branch is retained to generate text descriptions. In the following, we will describe each part in detail.

3.1. Feature Representation

Given a sequence of RGB frames $\{x_1, x_2, \ldots, x_T\}$, we extract two types of features out of them: scene features and object features.

Scene Features. We follow the procedure in [30], where we first extract a sequence of 2D frame features $F_{2D} = \{f_1, f_2, \ldots, f_T\}$ using ResNet-101 [16], with each $f_t \in \mathbb{R}^{d_{2D}}$. We also extract a set of 3D clip features $F_{3D} = \{v_1, v_2, \ldots, v_L\}$ using I3D [2], where $v_l \in \mathbb{R}^{d_{3D}}$.

Object Features. We run Faster R-CNN [12] on each frame to get a set of object features $F_o = \{o_1^1, o_1^2, \ldots, o_t^j, \ldots, o_T^{N_T}\}$, where N_t denotes the number of objects in frame t and j is the object index within each frame. Each o_t^j has the same dimension d_{2D} as F_{2D}.

3.2. Spatio-Temporal Graph

Objects have radically different behaviors across the space and time domains. On the one hand, different objects interact with each other spatially. While on the other hand, the same objects transform (shape, location, pose, etc.) temporally. In order to capture these two types of correlations, we decompose our graph into two components: the spatial graph and the temporal graph. A unique undirected spatial graph is instantiated for each frame, whose adjacency matrix is denoted by G_t^{space} for time step t. For the temporal graph, in order to not overwhelm the model with noisy information, we only calculate temporal edges between an adjacent frame pair instead of in a fully-connected manner [11, 43]. Note that the temporal graph is still connected across all time steps in this way. The resulted temporal graph going from t to $t+1$ is represented as G_t^{time}, which is a directed graph following along the direction of time flow.

Spatial Graph. The goal of the spatial graph is to capture interactions among spatially related objects. Take the scene shown in Fig. 2 for example. With the help of the object detector, we know there is a "cat" as well a "box" in the scene, but how can we get a clue on whether the cat is interacting with the box? The crux of solving this problem lies in the relative spatial location of the objects. Based on the observation that objects which are close to each other are more likely to be correlated, we explicitly incorporate this information in the spatial graph by connecting objects using their normalized Intersection over Union (IoU) value:

$$G_{tij}^{space} = \frac{\exp \sigma_{tij}}{\sum_{j=1}^{N_t} \exp \sigma_{tij}}, \tag{1}$$

where G_{tij}^{space} is the (i,j)-th element of $G_t^{space} \in \mathbb{R}^{N_t \times N_t}$, which measures the spatial connectivity between the ith and jth objects at time step t. We adopt the Softmax function as the normalization function similar to [43, 45]. σ_{tij} denotes the IoU between the two objects.

Temporal Graph. While the spatial graph has the capability of capturing interactions among objects at one time step, it is unable to model the object transformations over time. In the example in Fig. 2, there is no way to tell what the cat is doing with the box with any single frame. To this end, we propose to connect all semantically similar objects in every adjacent frame pair by computing their pair-wise cosine feature similarity:

$$G_{tij}^{time} = \frac{\exp \cos (o_t^i, o_{t+1}^j)}{\sum_{j=1}^{N_{t+1}} \exp \cos (o_t^i, o_{t+1}^j)}, \tag{2}$$

where G_{tij}^{time} denotes the (i,j)-th element of $G_t^{time} \in \mathbb{R}^{N_t \times N_{t+1}}$, and $\cos(o^i, o^j)$ measures the cosine similarity between the two feature vectors.

Convolutions on the Spatio-Temporal Graph. After we get the topological graph structure following the procedure above, the next step is to update the node features based on this graph structure. We adopt Graph Convolution (GCN) [22] for this. In order to extend the original GCN to our space-time domain, we first merge all spatial and temporal graphs for a video into a single spatio-temporal graph G^{st}:

$$G^{st} = \begin{bmatrix} G_1^{space} & G_1^{time} & 0 & \ldots & 0 \\ 0 & G_2^{space} & G_2^{time} & \ldots & 0 \\ 0 & 0 & G_3^{space} & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & G_T^{space} \end{bmatrix} \in \mathbb{R}^{N \times N}, \tag{3}$$

where each G_t^{space} and G_t^{time} are the spatial and temporal adjacency matrices we defined above. Note that the 0s in Eq. 3 are zero-valued matrices, whose shapes are determined correspondingly by the neighboring space and time matrices. N is the total number of objects in the video, i.e., $N = \sum_{t=1}^{T} N_t$.

At this point, the graph can be updated via the standard graph convolution, which is formally defined as follows:

$$H^{(l+1)} = \text{ReLU}(H^{(l)} + \Lambda^{-\frac{1}{2}} G^{st} \Lambda^{-\frac{1}{2}} H^{(l)} W^{(l)}), \tag{4}$$

where $W^{(l)} \in \mathbb{R}^{d_{model} \times d_{model}}$ is the weight matrix of layer l. Λ is the diagonal degree matrix with $\Lambda_{ii} = \sum_j G_{ij}^{st}$. We follow [47] to add in the residual connection and use ReLU as the activation function. GCN is implemented by performing $1 \times 1 \times 1$ convolution on the input tensor $H^{(l)}$ followed by multiplying the resulting tensor with $\Lambda^{-\frac{1}{2}} G^{st} \Lambda^{-\frac{1}{2}}$. $H^{(l)} \in \mathbb{R}^{N \times d_{model}}$ is the activation from layer l. Particularly, $H^{(0)}$ are the stacked object features:

$$H^{(0)} = \text{stack}(F_o) W_o \in \mathbb{R}^{N \times d_{model}}, \tag{5}$$

where stack() stacks all object features in F_o along the first axis, and $W_o \in \mathbb{R}^{d_{2D} \times d_{model}}$ is the transformation matrix.

Then we perform spatial average pooling on the updated H^{N_l} (N_l is the number of graph convolution layers), after which we get the final object features as $F_o' \in \mathbb{R}^{T \times d_{model}}$.

3.3. Scene Branch

Similar to previous work [8, 11, 34, 43, 49, 51], we also directly model the frame sequence through a separate scene branch. This branch provides additional global context information that may be missing from the local object features, and is especially critical when a video has no or very few objects detected. In order to highlight the effect of our proposed spatio-temporal graph and isolate the performance from the progress in scene modeling, we keep this scene branch as simple as possible. Concretely, for every 16 consecutive non-overlapping frames, we extract one 3D feature. Then we replicate the 3D features 16 times along temporal dimension (as each 3D feature spans and provides the context across 16 time steps), and sample the T slices corresponding to the 2D features. Subsequently, we project 2D and 3D features to the same dimension d_{model}, then concatenate them together and project again to d_{model}:

$$F_s = [F_{2D}W_{2D}; F'_{3D}W_{3D}]W_{fuse} \in \mathbb{R}^{T\times d_{model}}, \quad (6)$$

where $W_{2D} \in \mathbb{R}^{d_{2D}\times d_{model}}$, $W_{3D} \in \mathbb{R}^{d_{3D}\times d_{model}}$ and $W_{fuse} \in \mathbb{R}^{2d_{model}\times d_{model}}$ are transformation matrices. F'_{3D} represents the 3D features after the process stated above. $[;]$ denotes concatenation along channel dimension.

3.4. Language Decoder

During training, we pass in both scene features F_s and object features F'_o to perform language decoding. At test time, only F_s is used to generate the predicted sentence. Again as our work focuses on the visual encoding component, we keep the language decoder as simple as possible. We directly adopt the TVT architecture [4]. Specifically, the encoder takes a temporal sequence of features (either F_s or F'_o) and produces an embedding. The decoder receives this embedding and the previous word encoding to generate the next word. To clarify our naming, we denote the original encoder-decoder Transformer structure as our language decoder. Please refer to [4] for further details on the language decoder. Note that we use two separate Transformers for our two branches, and train them simultaneously. We adopt the standard training procedure to minimize the language cross-entropy loss L_{o_lang} and L_{s_lang} for the object and scene branch, respectively.

3.5. Object-Aware Knowledge Distillation

The problem with merging two branches through feature concatenation [43] or pooling [49], or adding scene features as a separate graph node [8, 11, 34] is that videos (and even frames in the same video) contain a variable number of objects, and this makes the learned features very noisy. This is because by either merging or adding an extra node, it imposes hard constraints on features that come from two intrinsically different spaces. By contrast, we only apply soft regularization on language logits, which are essentially probability distributions, thus being able to ensure a robust feature learning process and leverage the object information at the same time. The way of aligning language logits can be thought of as doing late fusion of the two branches, rather than early fusion as direct feature merging does. Concretely, we follow [18] to minimize the KL divergence between word probability distribution from the two branches. Let $P_o(x)$ be the probability distribution (pre-Softmax logits) across the vocabulary V from object branch and $P_s(x)$ be the distribution from scene branch. We minimize a distillation loss:

$$L_{distill} = -\sum_{x\in V} P_s(x)\log\left(\frac{P_o(x)}{P_s(x)}\right). \quad (7)$$

Note that we do not perform distillation by minimizing the L2 distance between features [14] as it is essentially putting hard constraints on features, and we will show through experiments that it yields inferior results.

3.6. Training

We freeze the scene and object feature extractors and only train the rest of the model. The overall loss function consists of three parts, i.e.:

$$L = L_{o\_lang} + \lambda_{sl}L_{s\_lang} + \lambda_d L_{distill}, \quad (8)$$

where λ_{sl} and λ_d are trade-off hyper-parameters.

4. Experiments and Results

We evaluate our proposed model on two challenging benchmark datasets: Microsoft Research-Video to Text (MSR-VTT) [46] and Microsoft Video Description Corpus (MSVD) [3]. To have a comprehensive evaluation, we report numbers on four commonly used metrics: BLEU@4, METEOR, ROUGE-L, and CIDEr.

4.1. Datasets

MSR-VTT. MSR-VTT is a widely used large-scale benchmark dataset for video captioning. It consists of 10000 video clips, each human-annotated with 20 English sentences. The videos cover a diverse set of 20 categories spanning sports, gaming, cooking, etc. We follow the standard data split scheme in previous work [30, 39, 49]: 6513 video clips in training set, 497 in validation, and 2990 in testing.
MSVD. MSVD is another popular video description benchmark, which is composed of 1970 video clips collected from YouTube. It supports multi-lingual description by annotating each video clip with sentences from multiple languages. Following the standard practice [30, 39, 49], we only select those English captions, after which we get approximately 40 descriptions per video, and 1200, 100, 670 clips for training, validation and testing, respectively.

4.2. Evaluation Metrics

In our experiments, we evaluate the methods across all four commonly used metrics for video captioning, namely BLEU@4 [29], ROUGE-L [25], METEOR [1], and CIDEr [36]. BLEU@4 measures the precision of 4-grams between the ground-truth and generated sentences. ROUGE-L computes a harmonic mean of precision and recall values on the longest common subsequence (LCS) between compared sentences. METEOR, on the other hand, uses a uni-grams-based weighted F-score and a penalty function to penalize incorrect word order, and it is claimed to have better correlation with human judgment. Finally, CIDEr adopts a voting-based approach, hence is considered to be more robust to incorrect annotations. We follow the standard practice to use the Microsoft COCO evaluation server [5].

4.3. Implementation Details

Feature Extractor. For scene features, we follow [30] to extract both 2D and 3D features to encode scene information. We use the ImageNet [7] pre-trained ResNet-101 [16] to extract 2D scene features for each frame. Specifically, we pass in a center-cropped frame patch with size 224×224, and take the output from the average pooling layer to get a flattened F_{2D} with $d_{2D} = 2048$. We also use the Kinetics [21] pre-trained I3D [2] for 3D scene feature extraction, where the input is a video segment consisting of 16 consecutive frames and we take the output from the last global average pooling layer to obtain a F_{3D} with $d_{3D} = 1024$.

To extract object features, we first apply a Faster-RCNN (with ResNeXt-101 + FPN backbone) [12] pre-trained on Visual Genome [23] to generate object bounding boxes for each frame. We set the confidence score threshold for a detection to be considered at 0.5. Given the output bounding boxes, we apply RoIAlign [15] to extract features of the corresponding regions. Specifically, we first project the bounding boxes onto the feature map from the last convolutional layer of ResNeXt-101, then apply RoIAlign [15] to crop and rescale the object features within the projected bounding boxes into the same spatial dimension. This generates a $7 \times 7 \times 2048$ feature for each object, which is then max-pooled to $1 \times 1 \times 2048$.

Hyper-parameters. For feature extraction, we uniformly sample 10 frames for both F_s and F_o (i.e., $T = 10$). We set the maximum number of objects in each frame to be 5. Specifically, we take the 5 most confident detections if there are more, and do zero-padding if there are less.

For the spatio-temporal graph, we stack 3 graph convolution layers, whose input and output channel number are all $d_{model} = 512$. In our language decoder, both the Transformer encoder and decoder have 2 layers, 8 attention heads, 1024 hidden dimension size, and 0.3 dropout ratio.

For the trade-off hyper-parameters in the loss function, we set λ_{sl} and λ_d to be 1 and 4, respectively. All hyper-parameters were tuned on the validation set.

Other Details. We adopt Adam with a fixed learning rate of 1×10^{-4} with no gradient clipping used. We train our models using batch size 64 for 50 epochs and apply early stopping to find the best-performed model. During testing, we use greedy decoding to generate the predicted sentences. All our experiments are conducted on two TITAN X GPUs.

4.4. Experimental Results

Comparison with Existing Methods. We first compare our approach against earlier methods, including **RecNet** [40], which adds one reconstructor on top of the traditional encoder-decoder framework to reconstruct the visual features from the generated caption, and **PickNet** [6] which dynamically attends to frames by maximizing a picking policy. We also compare to several very recent works that achieve strong performance. **MARN** [30] densely attends to all similar videos in training set for a broader context. **OA-BTG** [49] constructs object trajectories by tracking the same objects through time. While these works generally focus on the encoding side, **Wang *et al.*** [39] and **Hou *et al.*** [19] focus on the language decoding part and both propose to predict the POS structure first and use that to guide the sentence generation.

Note that among all these methods, we use the same scene features as MARN [30], i.e., ResNet-101 and I3D, so our method is most comparable to MARN. We also follow the standard practice [30] to not compare to methods based on reinforcement learning (RL) [39].

The quantitative results on MSR-VTT and MSVD are presented in Table 1 and Table 2, respectively. On MSVD, our proposed method outperforms all compared methods on 3 out of 4 metrics by a large margin. While on MSR-VTT, the performance of our model is not as outstanding. We summarize the following reasons for this: (1) MSR-VTT contains a large portion of animations, on which object detectors generally fail, thus making it much harder for our proposed spatio-temporal graph to capture object interactions in them; (2) The two very recent methods, i.e., Wang *et al*. [39] and Hou *et al*. [19] both directly optimize the decoding part, which are generally easier to perform well on language metrics compared to methods that focus on the encoding part, such as ours; (3) The more advanced features used (IRv2+I3D optical flow for Wang *et al*. [39] and IRv2+C3D for Hou *et al*. [19]) make it unfair to directly compare with them. Nonetheless, our method demonstrates a clear boost over other baselines, including the most comparable one MARN [30], as well as our own baseline, i.e., Ours (Scene), where only the scene branch is used. This manifests the effectiveness of our proposed method.

Ablation Study. At a high level, our proposed method consists of two main components: the spatio-temporal graph and the object-aware knowledge distillation. The spatio-

Table 1: Comparison with other methods on MSR-VTT (%). "-" means number not available. The first section includes methods that optimize language decoding, while the second is for those that focus on visual encoding.

Method	BLEU@4	METEOR	ROUGE-L	CIDEr
Wang *et al*. [39]	42.0	28.2	61.6	48.7
Hou *et al*. [19]	**42.3**	**29.7**	**62.8**	**49.1**
RecNet [40]	39.1	26.6	59.3	42.7
PickNet [6]	41.3	27.7	59.8	44.1
OA-BTG [49]	**41.4**	28.2	-	46.9
MARN [30]	40.4	28.1	60.7	**47.1**
Ours (Scene only)	37.2	27.3	59.1	44.6
Ours	40.5	**28.3**	**60.9**	**47.1**

Table 2: Comparison with other methods on MSVD (%).

Method	BLEU@4	METEOR	ROUGE-L	CIDEr
Wang *et al*. [39]	52.5	34.1	71.3	88.7
Hou *et al*. [19]	52.8	36.1	71.8	87.8
RecNet [40]	52.3	34.1	69.8	80.3
PickNet [6]	52.3	33.3	69.6	76.5
OA-BTG [49]	**56.9**	36.2	-	90.6
MARN [30]	48.6	35.1	71.9	92.2
Ours	52.2	**36.9**	**73.9**	**93.0**

temporal graph further contains two sub-components at a lower level, which are the spatial graph and the temporal graph. We evaluate the performance of several variants to validate the efficacy of each component. We first evaluate (1) **Scene Branch Only** where only the scene branch is used, (2) **Two Branch + Concat** where both branches are used, but the fusion of two branches is done by direct concatenation of features before passing into Transformers, and (3) **Two Branch + L2** which minimizes the L2 distance between features for distillation. These are intended to show the effectiveness of the two high-level components. In order to test different types of graph connection, we evaluate the performance of (4) **Spatial Graph Only** which only calculates the spatial graph G^{space} while setting G^{time} to all 0s, (5) **Temporal Graph Only** which similarly constructs only the temporal graph G^{time} and puts G^{space} to all 0s, as well as (6) **Dense Graph** which densely connects all objects with uniform weights (i.e., G^{st} is set to all 1s). (6) is also the method proposed in Wang *et al*. [43]. Note that we also compare with the spatial attention approach introduced in Ma *et al*. [27] and Zhou *et al*. [51], which is essentially equivalent to **Spatial Graph Only** because the attentive object aggregation only happens spatially and temporal modeling is done by passing the spatially attended object feature sequence into language decoder. The ablation study results on MSVD are shown in Table 3.

We first investigate the effect of the two high-level components. Both "Two Branch + Concat" and "Two Branch + L2" perform worse than the "Scene Branch Only" baseline, which suggests that imposing hard constraints on features not only fails to exploit useful object-level information, but even hurts performance by overwhelming the model with noisy features. Once making the object branch regularize the learning of the scene branch via logit alignment (which is "Our Full Model"), the object-level information becomes useful and gives a significant performance boost. Then we analyze the role each sub-graph plays. "Spatial Graph Only" and "Temporal Graph Only" achieve similar results, but are both inferior to "Our Full Model." This validates that both sub-graphs capture important and distinct information. Finally, we would like to see how much effect prior knowledge has when creating the graph. We see a large performance margin between "Dense Graph" and "Our Full Model," which corroborates our argument that prior knowledge about spatial layout and temporal transformation provides the model with more helpful information.

Table 3: Ablation study on MSVD (%).

Method	BLEU@4	METEOR	ROUGE-L	CIDEr
Scene Branch Only	45.8	34.3	71.0	86.0
Two Branch + Concat	45.5	34.1	70.7	79.3
Two Branch + L2	46.1	33.7	70.6	80.3
Spatial Graph Only	50.8	36.1	72.9	91.8
Temporal Graph Only	50.7	36.1	73.1	92.1
Dense Graph	51.4	35.9	72.8	91.3
Our Full Model	**52.2**	**36.9**	**73.9**	**93.0**

Qualitative Analysis. In order to validate that after distilling knowledge from the object branch our model can indeed perform better visual grounding, we plot the saliency maps for 4 example videos from MSR-VTT. Concretely, we plot for both "Scene Branch Only" and "Our Full Model" for comparison. We also compare the captions generated by "Our Full Model" and Wang *et al*. [39]. We merge them together into Fig. 3.

We first observe that "Our Full Model" is able to attend to key regions much better than its "Scene Branch Only" counterpart. In the video at the top left corner, "Our Full Model" pays most of its attention to the man's face as well as the paddles, while "Scene Branch Only" rarely focuses on these key parts. Similarly, in the example at the top right corner, "Our Full Model" always keeps its attention to the group of people that are running, while the attention of "Scene Branch Only" is mostly diffused. This further proves that our proposed spatio-temporal graph, along with the object-aware knowledge distillation mechanism, grants the model better visual grounding capability.

We then compare the captions generated from "Our Full Model" with those from Wang *et al*. [39]. The captions from "Our Full Model" are generally better visually grounded

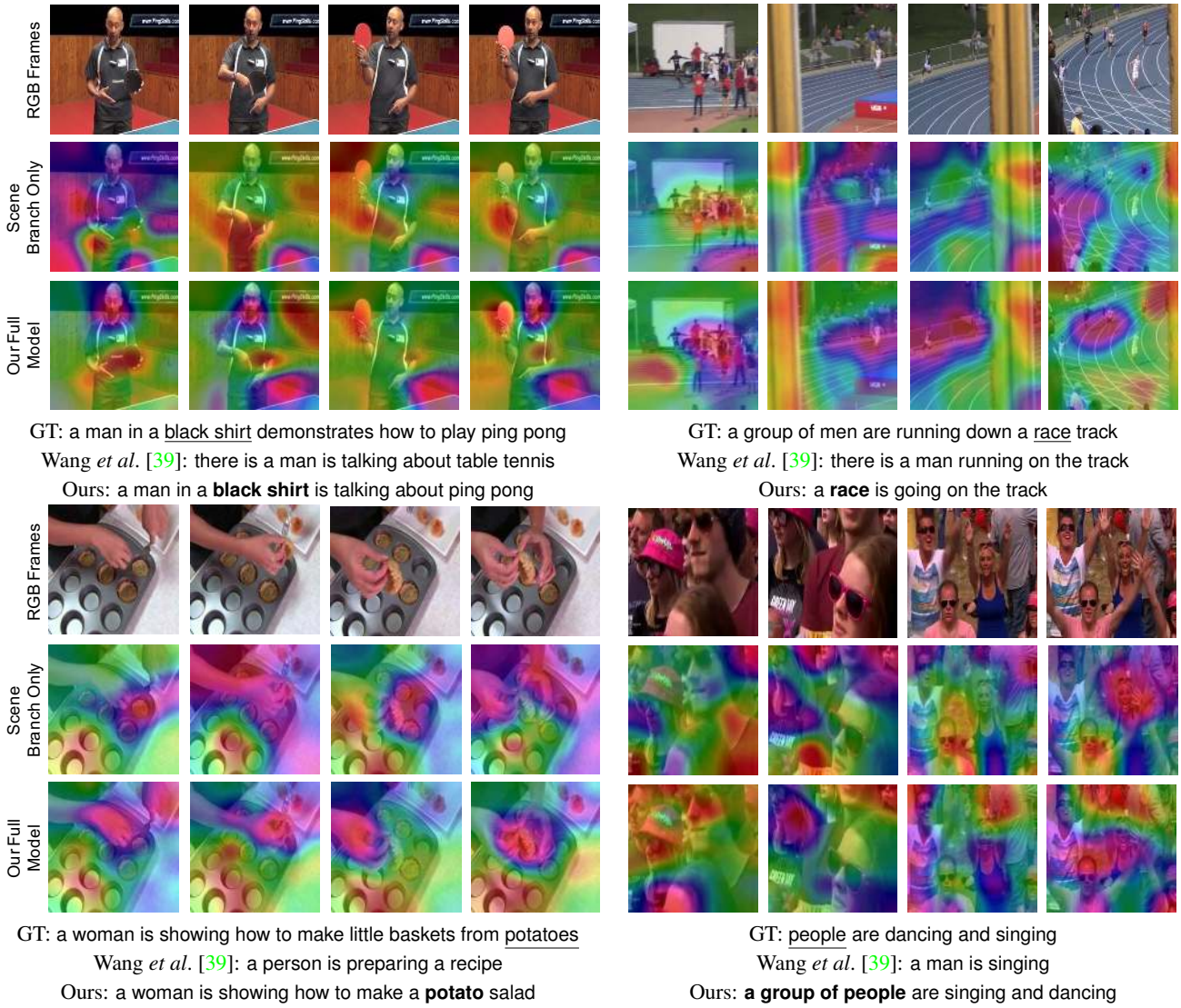

Figure 3: Qualitative results on 4 videos from MSR-VTT. (1) For each video, the first row shows its RGB frames, while the second and third rows are the saliency maps from our "Scene Branch Only" and "Our Full Model" variants (refer to "Ablation Study" for details), respectively. Specifically, red color indicates high attention scores, while blue means the opposite. We also present the ground-truth (GT), predicted sentences from both Wang *et al*. [39] and "Our Full Model" (Ours).

than Wang *et al*. [39]. For example, our model is able to predict very fine-grained details such as "black shirt" for the video at the top left corner, and "potato" for the video at the bottom left corner. It is also capable of grounding larger-scale semantic concepts, e.g., "race" (which indicates there is more than one person) for the top-right-corner video and "a group of people" for the bottom-right-corner one.

5. Conclusion

In this paper, we propose a novel spatio-temporal graph network for video captioning to explicitly exploit the spatio-temporal object interaction, which is crucial for scene understanding and description. Additionally, we design a two-branch framework with a proposed object-aware knowledge distillation mechanism, which solves the problem of noisy feature learning present in previous spatio-temporal graph models. We demonstrate the effectiveness of our approach on two benchmark video captioning dataset.

Acknowledgements Toyota Research Institute (TRI) provided funds to assist the authors with their research, but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity. We thank our anonymous reviewers, Andrey Kurenkov, Chien-Yi Chang, and Ranjay Krishna, for helpful comments and discussion.

References

[1] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[2] 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.

[3] David L 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-Volume 1*, pages 190–200. Association for Computational Linguistics, 2011.

[4] 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, 2018.

[5] 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.

[6] 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.

[7] 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.

[8] Lifeng Fan, Wenguan Wang, Siyuan Huang, Xinyu Tang, and Song-Chun Zhu. Understanding human gaze communication by spatio-temporal graph reasoning. *arXiv preprint arXiv:1909.02144*, 2019.

[9] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *arXiv preprint arXiv:1812.03982*, 2018.

[10] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016.

[11] Pallabi Ghosh, Yi Yao, Larry S Davis, and Ajay Divakaran. Stacked spatio-temporal graph convolutional networks for action segmentation. *arXiv preprint arXiv:1811.10575*, 2018.

[12] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[13] Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 2712–2719, 2013.

[14] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2827–2836, 2016.

[15] 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.

[16] 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.

[17] Roei Herzig, Elad Levi, Huijuan Xu, Eli Brosh, Amir Globerson, and Trevor Darrell. Classifying collisions with spatio-temporal action graph networks. *arXiv preprint arXiv:1812.01233*, 2018.

[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[19] Jingyi Hou, Xinxiao Wu, Wentian Zhao, Jiebo Luo, and Yunde Jia. Joint syntax representation learning and visual cue translation for video captioning. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[20] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015.

[21] 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.

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

[23] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

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

[26] David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*, 2015.

[27] 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.

[28] Effrosyni Mavroudi, Benjamín Béjar Haro, and René Vidal. Neural message passing on hybrid spatio-temporal visual and symbolic graphs for video understanding. *arXiv preprint arXiv:1905.07385*, 2019.

[29] 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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[30] 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 Conference on Computer Vision and Pattern Recognition*, pages 8347–8356, 2019.

[31] Anna Rohrbach, Marcus Rohrbach, Wei Qiu, Annemarie Friedrich, Manfred Pinkal, and Bernt Schiele. Coherent multi-sentence video description with variable level of detail. In *German conference on pattern recognition*, pages 184–195. Springer, 2014.

[32] Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. Translating video content to natural language descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 433–440, 2013.

[33] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[34] Yao-Hung Hubert Tsai, Santosh Divvala, Louis-Philippe Morency, Ruslan Salakhutdinov, and Ali Farhadi. Video relationship reasoning using gated spatio-temporal energy graph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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. *arXiv preprint arXiv:1908.10072*, 2019.

[40] 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.

[41] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.

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

[43] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 399–417, 2018.

[44] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2019.

[45] Jianchao Wu, Limin Wang, Li Wang, Jie Guo, and Gangshan Wu. Learning actor relation graphs for group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9964–9974, 2019.

[46] 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.

[47] 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.

[48] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 670–685, 2018.

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

[50] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–818, 2018.

[51] Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. Grounded video description. In *CVPR*, 2019.

[52] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8748, 2018.