

Multimodal Graph Convolution Network for Video Event Captioning

Anonymous Author(s)

Submission Id: 1751*

ABSTRACT

Video event captioning has attracted increasing attention which aims to generate detailed descriptive captions. To generate relevant and coherent captions for all events in the videos, one of the key points is to effectively capture the event dependency. However, existing video event captioning methods are hard to model the long-dependency between events well. Thus, in this paper, inspired by the recent success of graph neural networks in capturing the relations between none-sequential nodes, we propose a multimodal graph neural network for dense video captioning, called MGCN. Specifically, we first design two event-specific graphs for visual nodes and text nodes separately. Next, we construct a heterogeneous event-specific graph by combining the visual and text graph with the novel event-specific cross-modal alignment edges. Then based on the defined heterogeneous graph, we adopt the graph convolution network to enhance the original visual and text features with the context information preserved. Finally, we fuse the enhanced features and original features, and feed them to the transformer encoder-decoder framework for caption generation. Extensive experiments demonstrate the proposed MGCN is effective in generating coherent captions between events and achieving state-of-the-art performance on the Youcook2 benchmark dataset.

CCS CONCEPTS

• Computing methodologies → Video event captioning.

KEYWORDS

video event captioning, graph network

ACM Reference Format:

Anonymous Author(s). 2020. Multimodal Graph Convolution Network for Video Event Captioning. In *Multimedia '21: the 29th ACM International Conference on Multimedia*, Oct 12–16, Chengdu, China. ACM, New York, NY, USA, 9 pages. <https://doi.org/00.0000/0000000.0000000>

1 INTRODUCTION

Dense video event captioning is a fundamental research problem in video understanding, in which a sequence of events are segmented and a descriptive sentence is generated for each event in a long untrimmed video. For instance, the "how to cook pizza" video consists of these event sequences: 1) make dough 2) put on

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Multimedia '21, Oct 12–16, 2021, Chengdu, China

© 2020 Association for Computing Machinery.

ACM ISBN 978-0-0000-0000-0/00/00...\$15.00

<https://doi.org/00.0000/0000000.0000000>



Figure 1: A showcase. The "squid ring" is the global object, which appears in the 1_{st}, 4_{th}, and 5_{th} segments. Although it is relatively easy to recognize from the 1_{st} segment, it is hard to directly recognize from the 4_{th} and 5_{th} segment. From this case, we can see that long-range dependency and global context are important for generating coherent captions.

tops and cheese and 3) heat it in the oven. In this paper, we focus on the captioning task to generate descriptions for all event as in AT+Video[8], WLT [38], VideoBERT [33], MDVC [12], MART [18], TVPC [30], without the requirement of video segmentation generation. As mentioned in these works, generating coherent captions from a provided list of video segments can be very challenging. To generate relevant and coherent captions for all events in the videos, one of the key points is to effectively capture the surrounding context and event dependency especially for long-range videos.

Thus, early video captioning models [17] mainly adopt recurrent neural networks like LSTM [9], GRU [5] for the modeling of event sequence to capture surrounding context. Compared with recurrent neural networks based methods, transformer based methods are more effective for inner-event interaction, thus, this type of methods are becoming the prevailing architecture for the video captioning task recently. For example, Zhou etc.[46] proposed an end-to-end dense video captioning using vanilla transformer. AT+Video [8] and MDVC [12] further extended the model with multi-modal inputs

of both visual feature and speech text extracted through ASR tools to generate description for each event independently.

However, existing works are still infeasible for none-sequential context and especially difficult to model long-range dependency between events to capture global context information. Moreover, long-range dependency and global context are important for the video event captioning task. Figure 1 explains a showcase. The "squid ring" is the global object, which appears in the 1_{st} , 4_{th} , and 5_{th} segments. It is very hard to directly recognize the object from the 4_{th} and 5_{th} segment, which can be learned from the 1_{st} segment. Besides, as a prior knowledge to human, the actions "wash the carrot" and "take out knife" are likely to lead to the subsequent action "cut the carrot". From the above cases, we can see that long-range event dependency and global context are important for generating coherent captions.

Thus, in these paper, inspired by the success of graph convolution network [16] in capturing the relations between none-sequential nodes, we propose a multi-modal graph convolution network (MGCN) for video events captioning. Specifically, we first design two event-specific graphs for visual nodes and text nodes separately. In details, we design event-specific contextual edges and ordering edges for visual graph, and event-specific text edges for speech text. To be specific, the Event-specific contextual edges is to model long-range event dependency, event-agnostic ordering edges for temporal orders, and the text graph for global context. Next we integrate them into one unified heterogeneous event-specific graph through the novel event-specific cross-modal alignment edges for cross-modality message passing. Then based on the defined heterogeneous graph, we adopt the graph convolution network to enhance the original visual and text features with the context information preserved. Finally, we fuse the enhanced features and original features, and feed them to the general encoder-decoder framework for caption generation. Extensive experiments demonstrate the proposed MGCN is effective in generating coherent captions between events and achieves state-of-the-art performance on Youcook2[45] benchmark dataset.

Our contributions can be summarized as:

- 1) We propose a multi-modal graph convolution network (MGCN) for dense video captioning.
- 2) We define a novel heterogeneous event-specific graph to capture the long-range dependency between events and global context.
- 3) Experimental results on Youcook2 Captions dataset demonstrate the effectiveness of our model and outperform the context-agnostic model to a large extent.

2 RELATED WORK

Dense Video Captioning There are two types of dense video events captioning: paragraph-level captioning [6, 18, 27, 40] and event-level captioning [3, 17, 23, 46]. The difference between these tasks is whether to generate multiple sentences for all events together or one sentence for each event separately given a list of event segments in the same video. In this paper, we focus on the event-level dense video captioning task which is more challenging due to the requirement of precise description for each event. Previous works [17, 19, 36] mainly exploited recurrent neural models such as long short-term memory network (LSTM) [9] or recurrent

unit (GRU) [5] to encode sequential context. However, recurrent model suffers from modeling long dependency effectively. Another and the recent popular models are transformer [34] based models [32, 33, 46]. Zhou et al. [43] proposed an end-to-end self-attention model, and Sun et al. [32, 33] further pre-train a self-attention model [34] to generates better captions. However, those model either employed recurrent sequential model for short-term context encoding or attention model to generate caption for each event solely. An intuitive idea is to simply feeding all context into the transformer model, but it will lead to two problems: 1) the transformer is hard to handle long sequence efficiently and effectively; 2) it is quite a challenge to select the relevant context and ignore the irrelevant context from all context. Thus, we proposed to leverage graph based network to capture long-dependency global context for dense video event captioning.

Multi-modal Video Captioning Video naturally has multi-modal signals including visual, speech text and audio. Previous works explore visual RGB, motion, optical flow features, audio features [10, 28, 38] as well as speech text features [8, 12, 29] for captioning. According to the work in [8, 12, 29], although the speech text is noisy and informal, it can still capture better semantic features and improve performance significantly especially for instructional videos. Later on, Lashin et al. [12] proposed to embed all visual, audio, and speech text for dense video event captioning. However, context-aware models are rarely investigated in multi-modal video event captioning.

GCN for Video Tasks GCN [16] has been widely used to effectively capture the graph based data structure, e.g. knowledge graph. Furthermore, GCN is effectively utilized for several visual related tasks including visual-text matching [4, 14], action classification [37], action localization [11, 41], grounding [1, 42], video QA [13], video summarization [26], video captioning [24, 44]. One research direction is to build the visual graph, and existing works [4, 4, 11, 26, 37, 41, 42] construct visual graph with either frame-frame, proposal-proposal or object-object relations. Another research direction is to build visual and language graph to perform matching. [1] proposed object-object visual graph and phrase-phrase language graph as well as a fusion graph for grounding. [13] proposed both visual and linguistic graph with fully connect for video question answering. Both works constructed modality specific graph and adapt attention based modality fusion method to predict the matching. Specifically for video captioning, [24] constructed spatial and temporal graph for both object-object and frame-frame relations. [44] built a relational graph between object. Those works demonstrated the effectiveness of GCN for single-sentence video clip captioning, which is inapplicable to dense video captioning without consideration of event dependency. In this paper, we propose a multi-modal heterogeneous graph for dense video captioning.

3 PROBLEM DEFINITION

Given a video $E = \{e_1, e_2, \dots, e_n\}$ represented by several temporally ordered event segments, where n denotes the total number events in the video and e_i denotes the i^{th} event, the goal of dense video event captioning is to generate a descriptive sentence y_i for each event e_i . Specifically, for each event $e_i = \{V_i, T_i\}$, it includes both video and transcript text features. $V_i = \{v_1^i, v_2^i, \dots, v_{|V_i|}^i\}$ denotes the set of video features in the event e_i , where $|V_i|$ represents

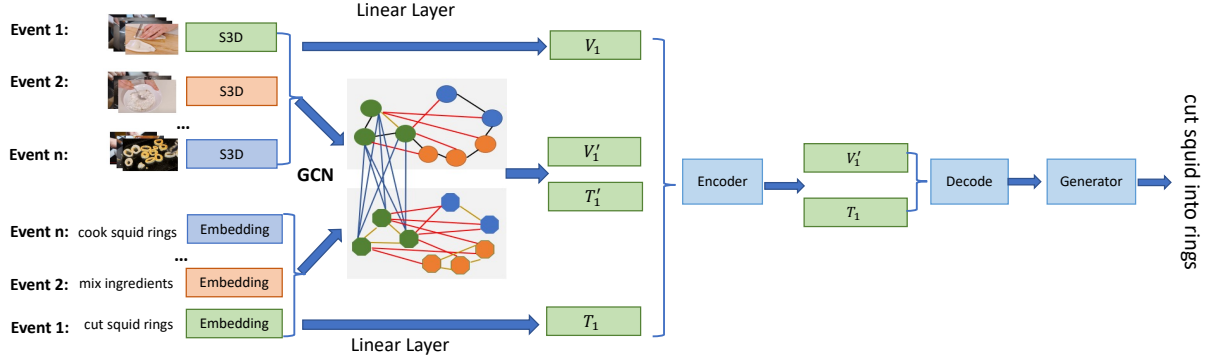


Figure 2: The workflow of our proposed MGCN. Suppose the Event 1 is the current event gonging to be captioned. First, the visual features of all events are extracted through an off-the-shelf 3D convolution model, and the text features of all events are represented by embedding features. Then, according to the features of all the events, a multi-modal heterogeneous event-specific graph is constructed for the current event e_1 whose target descriptive sentence are going to be predicted. After that, through the graph convolution network, we get the global context-aware features V'_1 and T'_1 for event e_1 , respectively. Finally, we combine the context-aware features V'_1 and T'_1 and event features V_1 and T_1 and feed them to the transformer-based encoder-decoder to get final representation, and adopt an autoregressive transformer to generate the caption for the event e_1 .

the number of total features in V_i and v_j^i denotes the j^{th} feature of V_i . $T_i = \{t_1^i, t_2^i, \dots, t_{|T_i|}^i\}$ denotes the transcript text features in the event e_i , where $|T_i|$ denotes the number of total frames in V_i and v_j^i denotes the j^{th} frame feature of V_i . Then, given all events $\{e_1, e_2, \dots, e_n\}$, the dense video event captioning model need to predict the target descriptive sentences $Y = \{y_i | i \in [1, n]\}$. y_i is a sequence of descriptive words corresponding to the event e_i . To optimize the model, this task usually minimizes the sum of the negative log-likelihood of the expected sentences Y :

$$L(Y|e) = - \sum_{i=1}^n \log P(y_i | e_i) \quad (1)$$

4 METHODOLOGY

The architecture of our proposed MGCN model is shown in Figure 2. Specifically, first the visual features of all events are extracted through an off-the-shelf 3D convolution model, and the text features of all events are represented by embedding features (see Section 4.1). Then, based on features of all the events, a multi-modal heterogeneous event-specific graph \mathcal{G}_i is constructed for the current event e_i whose target descriptive sentence are going to be predicted (see Section 4.2). After that, through the graph convolution network [7], the visual feature V_i and transcript text feature T_i are enhanced to be global context-aware features V'_i and T'_i , respectively (see Section 4.3). Finally, we combine the context-aware features V'_i and T'_i and event features V_i and T_i and feed them to the transformer-based encoder-decoder to get final representation, and adopt an autoregressive transformer to generate the caption y_i for the event e_i (see Section 4.4).

4.1 Features

We extract features for vision and speech text separately.

Video Representation We directly extract the visual feature through an off-the-shelf pre-trained S3D [39] model given the video clip for each segment. In this paper, we sampled frames at 16 fps and create clips from 16-frame. We take the feature before the final linear classifier of the S3D backbone and applied 3D average pooling to obtain a 1024-dimension feature vector. That is, we got one feature for each second, and the video feature of the i^{th} event is $V_i \in \mathbb{R}^{|V_i| \times d}$. In our experiment, $|v_i|$ is 80 and the padding is applied for the short event.

Text Representation We extract the embedding features for each token through an embedding layer. The text feature of the i^{th} event is $T_i \in \mathbb{R}^{|T_i| \times d}$. The input token number $|T_i|$ is set to 80, and we directly pad for short sentences and trim for long sentences.

4.2 Event-specific Graph Construction

The graph construction is the key part for the graph convolution network. Specifically, we first build a event-specific visual graph and a event-specific text graph separately, and then merge the two graphs to build a heterogeneous event-specific graph through temporal alignment.

4.2.1 Event-specific Visual Graph. For a event e_k in the video D , we define the event-specific visual graph with N nodes as $\mathcal{G}_k^v = (V, \mathcal{E}_v)$, where V and \mathcal{E}_v denote the visual nodes and edge sets, N is the total number of visual frame features in the video D , i.e., $N = |V_1| + |V_2| + \dots + |V_n|$. Moreover, each visual frame feature in the video is a visual node, and the edge between i -th and j -th nodes, denoted as ε_{ij} , shows the dependency between the two nodes. To capture both the local sequential context and global non-sequential context, we design two types of edges: (i) event-specific contextual edges and (ii) ordering edges.

Event-specific contextual edges. The goal of event-specific contextual edges is to model the long-dependency between the frames of current event and the frames from other events. One way to construct such edges is linking each node of current event

with all nodes in the other events. However, through this way, it will incur redundant even noisy information which will damage the long-dependency modeling. Hence, in our paper, the event-specific contextual edges only connect a node with its top $z\%$ nearest neighbors in each event that it does not belong to. Specifically, suppose the current event, whose target descriptive sentence are going to be predicted, is e_i , for a node v_k^i which is the k^{th} feature of event e_k , there are event-specific contextual edges between v_k^i with its top $m\%$ similar visual features in every event e except e_i , then the adjacency matrix A_i^c of event-specific contextual edges for event e_i can be formulated as follows:

$$A_i^c(k, j) = \begin{cases} 1 & v_j \in nn(v_k) \text{ and } v_k \in V_i \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $A_i^c(k, j)$ is the k^{th} row and j^{th} column of the adjacency matrix $A_i^c \in \{0, 1\}^{N \times N}$, and N is the total number of visual features in the video. V_i is the set of visual feature of event e_i . v_k is the k^{th} visual feature in the video. Please note, if the index of the j^{th} visual feature of event e_i in the video level is k , then $v_k = v_j^i$, i.e., v_k and v_j^i are two type of notation of a same visual feature. Moreover, $nn(v_k) = nn_{e_1}(v_k) \cup \dots \cup nn_{e_{i-1}}(v_k) \cup nn_{e_{i+1}}(v_k) \cup \dots \cup nn_{e_n}(v_k)$, where $nn_{e_z}(v_k)$ denotes the top $m\%$ similar visual features of v_k in event e_z . When $A_v^{inter}(i, j) = 1$, it represents the node v_i is connected with the node v_j , otherwise they are not connected.

Ordering edges. As videos are temporal sequences of frames, it is essential and model the temporal order between the consecutive video frames. Then the temporal edges are defined to capture the neighborhood relation between the consecutive video frames, and its adjacency matrix A^o can be formulated as follows:

$$A^o(k, j) = \begin{cases} 1 & \text{if } |k - j| \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $A^o(k, j)$ is the k^{th} row and j^{th} column of the adjacency matrix $A^o \in \{0, 1\}^{N \times N}$, and N is the total number of visual features in the video, and when $A^o(k, j) = 1$, it represents the node v_k is connected with the node v_j , otherwise they are not connected. Please note, because the sequence of visual features in a video is fixed, thus the the adjacency matrices of different events in the same video are the same.

With the two types of edges defined well, the final adjacency matrix A_i^v of event-specific visual graph for event e_i can be formulated as follows:

$$A_i^v = A^o + A_i^c \quad (4)$$

4.2.2 Event-specific Text Graph. We define the event-specific text graph as $\mathcal{G}_t = (\mathcal{T}, \mathcal{E}_{\mathcal{T}})$, where \mathcal{T} and $\mathcal{E}_{\mathcal{T}}$ denote the node and edge sets, respectively. Each node corresponds to one textual token in the speech text for inner segment and meaningful semantic concepts for other segments. Regards to construction of the text graph, we only consider the representative semantic concepts instead of all tokens by removing those stop words. Those noisy and meaningless tokens especially those high frequent adverb or conjunction words like 'the', 'to' lead to lots of unnecessary edges for error message passing. Moreover, it is inefficient to calculate all tokens in the

speech text. Intuitively, the high frequency words are likely to be those conjunction tokens and the low frequent token are unlikely to be propagated during the message passing of the graph. Inspired by TVPC[30], We adopt several heuristic rules for filtering manually. In details, we first tokenize each sentence into tokens and build token dictionary. Next, we filter high frequent tokens (top 100) and remove all less frequent tokens (frequent less than 3).

As a step forward, the \mathcal{E} is pair-wise connection between each nodes. Considering all semantic concepts are valuable for global context, we construct dense connection for the adjacent matrix. It means we link each node of current event e_i with all nodes in the other events. Moreover, we fully connect all the nodes in the same event. The final adjacency matrix of event-specific text graph for e_i is denoted as $A_i^t \in \{0, 1\}^{M \times M}$, and M is the total number of text tokens in the video:

$$A_i^t(k, j) = \begin{cases} 1 & t_k \in T_i \text{ or } condition_1 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where $condition_1$ is t_k and t_j belonging to the same event. t_k is the k^{th} text token/concept in the speech text. Moreover, please note, similar to the visual nodes, if the index of the j^{th} text token of event e_i in the video level is k , then $t_k = t_j^i$, i.e., t_k and t_j^i are two type of notation of a same text token.

4.2.3 Heterogeneous Event-specific Graph. We define the heterogeneous event-specific graph as $\mathcal{G}_{vt} = (\mathcal{H}, \mathcal{E})$, where \mathcal{H} is the node set with both visual and text nodes, and \mathcal{E} is the set of edges for visual-visual, text-text and visual-text nodes. Initially, the heterogeneous event-specific graph is a combination of event-specific visual graph and text graph, i.e., $\mathcal{H} = \mathcal{V} \cup \mathcal{T}$ and $\mathcal{E} = \mathcal{E}_{\mathcal{V}} \cup \mathcal{E}_{\mathcal{T}}$, respectively. To empower the message passing between visual node and text node, we define the **event-specific cross-modal alignment edges** which leverage the temporal correspondence to build fully connection between the visual nodes with the text nodes in the current event e_i , and its adjacency matrix A_i^{vt} is formulated as follows:

$$A_i^{vt}(k, j) = \begin{cases} 1 & v_k \in V_i \text{ and } t_j \in T_i \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where $A_i^{vt}(k, j)$ is the k^{th} row and j^{th} column of the adjacency matrix $A_i^{vt} \in \{0, 1\}^{N \times M}$ and N is the total number of visual frames in video and M is the total number of text tokens in the video.

Finally, by incorporating the event-specific cross-modal alignment edges into \mathcal{E} , the final adjacency matrix $A_i \in \{0, 1\}^{(N+M) \times (N+M)}$ of the event-specific heterogeneous event graph for event e_i is defined as follows:

$$A_i = \begin{bmatrix} A_i^v & A_i^{vt} \\ A_i^{vtT} & A_i^t \end{bmatrix} \quad (7)$$

$$A_i(k, j) = \frac{A_i(k, j)}{\sum_{z=1}^{N+M} A_i(k, z)} \quad (8)$$

4.3 Graph Convolution

To refine the representation of each node according to the defined heterogeneous event-specific graph, we apply graph convolution

network for message passing as in [16]. The detailed graph convolution operated on the predefined relations is performed as follows:

$$\hat{X} = A_i X W \quad (9)$$

$$Z = \text{ReLU}(\hat{X}) \quad (10)$$

where A_i is the adjacent matrix for event e_i . $X = [V, T]$ denotes the concatenation of all the visual frame features and all the text token features in the video. And the visual frame feature V_i and text token feature T_i for event e_i are part of V and T , respectively. W is the learnable weight matrix with $W \in \mathbb{R}^{d \times d}$. We adopt the ReLU activation function. Z is the final feature matrix after graph convolution, including the context-enhanced visual features V'_i and context-enhanced text features T'_i for the event segment e_i .

4.4 Caption Generation

Encoder After the graph convolution, for the current event e_i , the original visual feature V_i , text feature T_i , context-enhanced visual feature V'_i and text feature T'_i are important, then we fuse them together as the final representation. To fuse multi-modality information, dimension-wise add or product operation are quite intuitive and simple, which is widely used in previous work. Recently multi-head attention is the new paradigm for fusion. This multi-head attention is not only effective for cross-modal fusion, but also effective for enhanced feature with the original feature. Thus, we concatenate all the four types of features and then input them to Transformer [34] encoder to fuse them. The operation is specified as

$$E_{e_i} = [V_i; V'_i; T_i; T'_i] \quad (11)$$

$$T_{e_i} = \text{FFN}(\text{MultiHead}(E_{e_i})) \quad (12)$$

Where ";" represents the concatenation operation, E_{e_i} is the concatenated embedding of the four inputs. T_{e_i} is the results of multi-head attention operation for interaction between the four types of features. Next we only select the most effective features from the output of encoder as $\hat{E}_{e_i} = T_{e_i}[\text{index}_v : \text{index}_t]$, where index_v is the start index of the enhanced visual feature V'_i in T_{e_i} and index_t is the end index of the text feature T_i in T_{e_i} . The enhanced visual feature learns more information from both temporal and semantic context, while the text learns other context through the multi-head attention, which are the most effective features. Finally, \hat{E}_{e_i} is the final encoding of the features for the decoder.

Generation With the final representation of the encoder, we adopt the autoregressive transformer decoder to generate tokens of caption y_i one by one. We adopt the cross-entropy loss to minimize the negative log-likelihood over ground-truth words and apply the label smoothing strategy.

$$\mathcal{L} = -\sum_{i=1}^n \log P(y_i | e_i) \quad (13)$$

5 EXPERIMENT

5.1 Dataset and evaluation metrics

We run our experiments on Youcook2 dataset [45]. YouCook2 is an instructional video dataset for dense video captioning on recipe domain. Each video contains a sequence of steps to teach people

how to cook a dish. To make a fair comparison, we follow the data partition in VideoBERT [33] which uses 457 videos in the YouCook2 validation set as testing set and the rest for development. In all, we use 1,278 videos for training and validation. There are 89 cooking recipes, on average, each distinct recipe has 22 videos.

We employ the widely used language generation metrics including BLEU4 [25], METEOR [2], ROUGE-L[20] and CIDEr[35] to evaluate the performance. To get stable results, we run 3 times for each experiment below, and report the averaged scores.

5.2 Implementation details

Setting The embedding size of video, hidden size of the multi-head, and feed-forward layer are 1024, 512, and 128 respectively. The number of head is 8 and the dropout rate is 0.1. We adopt the Adam optimizer [15] with learning rate of 1e-4, and set two momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.98$. For label smoothing, and the smoothing rate is 0.1. We set the batch size to 96. All models are trained on 1 Tesla P100 GPUs for 9 hours for Youcook2 dataset for 70 epochs.

Visual feature We did ablation studies on two commonly used visual feature extraction models through the off-the-shelf S3D model pre-trained on Knetics[47] and the S3D model pre-trained on Howto100M[22] dataset through MIL-NCE[21]. Since Youcook2 and Howto100M are both instructional videos and in-domain datasets, our experimental results indicate that the S3D model pre-trained on Howto100M dataset performs much better on Youcook2 dataset and used it for our experiment. The ASR transcript is automatically extracted from the off-the-shelf recognition tool¹.

5.3 Compare with State-of-the-art results

We present the results of baseline methods and our MGCN model in Table 1. The baseline methods are two types: UniModal input (video-only) based method and MultiModal input (video + transcript) based method. We compared MGCN models of both types with the corresponding baseline methods.

UniModal Input based methods. the input to these models are video features without consideration of ASR speech text.

(1) Bi-LSTM + TempoAttn [31] adopts Bi-LSTM language encoder and is the baseline of [46]. It used a Bi-LSTM context encoder given visual features to TempoAttn[31], and applies temporal attention on Bi-LSTM output for all the language decoder layers.

(2) End-to-End Masked Transformer (EMT) [46] is a transformer based model. It takes each video segment as input and generates a single sentence describing the given segment independently. Although Zhou et al. also propose a separate proposal generation module, we only focus on its captioning module.

(3) VideoBERT [33] adopts bidirectional transformer and pre-training strategy to further enhance the visual representation. VideoBERT is pre-trained on 320k instructional videos and can benefit the in-domain Youcook2 dataset.

(4) CBT [32] exploits contrastive bidirectional transformer and is further pre-trained on 1M instructional videos.

MultiModal Input based methods. the inputs to these models are both video and ASR speech text features. Based on previous

¹<https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/>

Methods	Modality	Feature	B-4	M	R-L	CIDEr
Bi-LSTM + TempoAttn [31]	V	Resnet34	0.87	8.15	-	-
EMT[45]	V	Resnet200	4.38	11.55	27.44	38
VideoBERT[33]	V	S3D(Knetics+320k)	4.04	11.01	27.50	49
VideoBERT (+S3D feature)[33]	V	S3D (Knetics+320k)	4.33	11.94	28.80	55
CBT[32]	V	S3D (Knetics+HTM)	5.12	12.97	30.44	64
MGCN w/o GCN	V	S3D(Knetics)	4.74	12.64	29.54	46.04
MGCN	V	S3D(Knetics)	5.20	13.49	30.42	53.08
MGCN w/o GCN	V	S3D(HTM)	8.80	18.39	37.54	98.80
MGCN	V	S3D(HTM)	8.91	18.55	37.99	101.08
DPC [29]	VT	Resnet32	2.76	18.08	-	-
AT+video[8]	VT	Resnet34	9.01	17.77	36.65	112.10
TVPC[30]	VT	Resnet50	10.42	18.31	41.56	112.00
MGCN w/o GCN	VT	S3D(Knetics)	9.68	19.45	38.27	108.48
MGCN	VT	S3D(Knetics)	10.30	20.12	39.39	114.05
MGCN w/o GCN	VT	S3D(HTM)	11.11	21.04	41.60	130.82
MGCN	VT	S3D(HTM)	11.74	21.77	42.40	136.59

Table 1: The dense video event captioning results on Youcook2 dataset. The column "Modality" indicating which feature used in the method. "V" is for visual feature, and "T" is for speech text feature. The column "Feature" means which backbone is adopted for feature extraction.

experimental results, although the ASR speech text is noisy and informal, it is quite informative for semantics.

(1) DPC [29] is a LSTM based sequence-to-sequence model for captioning. It encodes all transcript text into a global text embedding, and fuses this overall text embedding to each visual features for further processing.

(2) AT+Video [8] is a multi-modal transformer method. It concatenates all speech text tokens with all features inside each segment and uses transformer to perform the multi-modal fusion. This work verified the effectiveness of the speech text.

(3) TVPC [30] is a transformer based model. It employs mean pooling on all visual feature to get a global visual feature, and adds the global visual feature to each text token as final representation. This work indicated that temporal alignment between context is important.

For our model, we conduct experiments on both "MGCN w/o GCN" and "MGCN" models. The "MGCN w/o GCN" model adopts the effective transformer encoder-decoder framework similar to AT+video [8] with multi-modal inputs, which is both simple and effective. The "MGCN" model mainly designs the multi-modal graph to effectively leverage context to enhance the representation. From the experimental results in table 1, we can see that: 1) the MGCN model can improve the baseline models by a large margin for both unimodal or multi-modal inputs, and achieves the state-of-the-art results; From Table 1, we can see that with the heterogeneous graph, the improvement of BLEU4 is around 0.6 points, METEOR for 0.7 points, Rouge for 0.8 points and CIDEr for 6 points compared with the strong baseline. 2) the feature extracted by S3D backbone pre-trained on Howto100M performs much better than that pre-trained on Knetics. Since the Howto100M dataset are instructional videos with 1/3 data of recipe domain, the S3D feature extractor pre-trained on this dataset can improve the visual representation for the in-domain Youcook2 videos. Through comparison of the different features for the video-only input model, we can see that the feature extractor impacts the results a lot. Nonetheless, the

Methods	GCN	B-4	M	R-L	CIDEr
MGCN _v	no	8.80	18.39	37.54	98.80
MGCN _v	yes	8.91	18.55	37.99	101.08
MGCN _t	no	8.70	18.47	36.18	99.75
MGCN _t	yes	9.27	18.60	36.49	102.07
MGCN _{vt}	no	11.11	21.04	41.60	130.82
MGCN _{vt}	yes	11.74	21.77	42.40	136.59

Table 2: Ablation study of visual graph, text graph and the heterogeneous graph. MGCN_v is the baseline model with video only input. MGCN_t is the baseline model with text only input. MGCN_{vt} is the baseline model with video and text input. The column GCN represents whether we adopt GCN enhanced representation for generation.

MGCN model consistently outperforms all baseline methods for both features.

5.4 Ablation Study

In this section we report our results on ablation studies to investigate the effectiveness of the constructed heterogeneous event-specific graph of the MGCN model from different perspectives.

The effect of GCN. To explore the capability of the GCN for each modality and multimodality, we conduct the experiments and present the results in the table 2. It can be found that all the video event captioning models, which are enhanced by the GCN, outperform all base models (w/o GCN) regardless of whether the input is video-only, text-only or video-text, which validate the GCN can learn useful information from the other event effectively to benefit the caption generating of current event.

The effect of event-specific ordering, contextual and cross-modal alignment edges. In this experiment, we investigate the effect of the three types of edges, and the results are shown in Table 3. Based on the Table 3, there are three observations: (1)

By adopting the event-specific ordering edges to construct the visual graph, it can make the model achieve better performance. It demonstrates that the event-specific ordering edges can capture the sequential local context to help the caption generating. (2) By adopting the event-specific contextual edges to construct the visual graph, MGCN can generate better captions for video events. For example, the performance of $MGCN_v$ with the event-specific contextual edges can outperforms the one without this type edges on all the four evaluation metrics. These results shows, with the event-specific contextual edges to model the long-dependency and capture the non-sequential global context, the visual features will be enhanced to help the event caption generating. (3) $MGCN_{vt}$ can achieve better performance by using the event-specific cross-modal alignment edges to align both visual and text graph. These result show that the event-specific cross-modal alignment edges can bridging more interaction between modality to enhance the feature representation well.

Methods	O	C	A	B-4	M	R-L	CIDEr
$MGCN_v$	-	-	-	8.80	18.39	37.54	98.80
$MGCN_v$	y	-	-	8.90	18.42	37.64	99.67
$MGCN_v$	-	y	-	8.82	18.47	37.74	100.39
$MGCN_v$	y	y	-	8.91	18.55	37.99	101.08
$MGCN_{vt}$	-	-	-	11.11	21.04	41.60	130.82
$MGCN_{vt}$	y	-	y	11.70	21.53	42.15	134.93
$MGCN_{vt}$	-	y	y	11.51	21.70	42.36	134.67
$MGCN_{vt}$	y	y	-	11.14	21.11	41.95	131.57
$MGCN_{vt}$	y	y	y	11.74	21.77	42.40	136.59

Table 3: Ablation study of event-specific ordering, contextual and cross-modal alignment edges. $MGCN_v$ is the baseline model with video only input. $MGCN_{vt}$ is the baseline model with video and text input. The columns "O", "C", and "A" represent the event-specific ordering, contextual and cross-modal alignment edges, respectively, and the "y" value in these columns denotes adopting the corresponding edges, otherwise denotes not adopting the corresponding edges.

Fusion methods. We investigate how to fuse the context enhanced features through graph convolution and present the results in Table 4. We also try several simple approaches including add or multiplication operation on the enhanced feature with the original feature. Besides, we also concatenate the features through dimension-wise concatenation or frame-wise concatenation. Through comparison with the baseline method $MGCN_{vt}$ w/o GCN in Table 3, we found that these baseline fusion methods even hurt the performance due to context confusion. Frame-wise concatenation used in the final MGCN model is an effective fusion methods, which concatenates the features and rely on the transformer model to perform attention to attentively learn the useful context.

Feature Ensemble for Decoder. An intuitive idea is to ensemble all the features for sufficient information including four types of features: V denoting visual feature, T denoting text feature, V' denoting context-enhanced visual feature and T' denoting context-enhanced text feature. Therefore, we concatenate all four types of

Fusion Methods	B-4	M	R-L	CIDEr
Add	10.52	20.25	40.04	123.07
Multiplication	11.32	20.36	40.54	126.28
Dimension-wise concatenation	9.50	18.32	37.85	103.22
Frame-wise concatenation	11.74	21.77	42.40	136.59

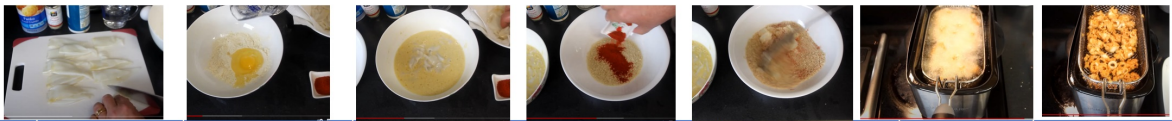
Table 4: Ablation study of Fusion methods. Features with Frame-wise concatenation followed by the transformer encoder is an effective attention mechanism to attend to the useful context.

Features	B-4	M	R-L	CIDEr
V	10.52	19.86	39.29	122.53
V'	9.91	19.53	39.45	117.91
T	11.70	21.64	42.42	136.20
T'	10.18	20.12	40.07	120.87
$V+T$	11.40	20.97	41.66	132.71
$V+T'$	10.96	20.15	40.38	124.68
$V'+T$	11.74	21.77	42.40	136.59
$V'+T'$	10.43	19.96	40.48	124.66
$V+V'+T$	11.36	20.96	41.63	131.64
$V+V'+T'$	10.76	20.42	40.63	125.30
$V+T+T'$	11.61	20.86	41.89	131.71
$V'+T+T'$	11.61	21.72	42.44	135.80
$V+V'+T+T'$	11.25	21.10	41.71	132.96

Table 5: Ablation study of feature ensemble. V denoting visual feature, T denoting text feature, V' denoting context-enhanced visual feature and T' denoting context-enhanced text feature. "+" denote the concatenating operator. For example, $V'+T = [V'; T]$. Note that those features are actually the output of the transformer encoder but not the original input.

features as input to the transformer encoder, and ensemble different outputs for decoder detailed in section 4.4. The transformer encoder performs the interaction between these features. Note that this study mainly focuses on features after the transformer encoder and before the decoder. From results shown in Table 5, we can see that $V'+T$ outperforms all other ensembled features. Thus, in the other experiments in our paper, we use the $V'+T$ as the input feature for the decoder.

Semantic Nodes Selection. As the defining of the event-specific contextual edges need to select some of the most similar nodes from other events, we conduct experiments to investigate the effect of the number of node being selected. Figure 4 shows the trends of each metric based on the selection percentage. Based on this figure, there are observations: (1) With selecting top 5% similar nodes to define the event-specific contextual edges, the model can achieve a good results on almost all metrics. Thus, we use this setting for the other experiments in our paper. (2) With increasing the percent of selected nodes, the curves of all the metrics are the zigzag lines. Its reason maybe that there are noisy or irrelevant nodes in the selected nodes, and such nodes may hurt the performance. Then when selecting more similar nodes, it introduces more noisy nodes



Segment	[00:39,01:03]	[01:19,01:30]	[01:47,02:10]	[02:25,03:05]	[03:09,03:28]	[03:59,04:07]	[04:13,04:21]
Ground Truth	cut squid into rings	add one egg and water to flour	put the squid in the mixture and stir	put some pepper red spice salt in a bowl and mix them	take out the squid and coat them with the mixed powder	deep fry the squid	take the squids out and drain on paper towel
MGCN w/o GCN	Cut the squid into rings	Mix flour egg and egg	Coat the squid rings with the batter	Add cayenne pepper salt and paprika to the breadcrumbs	Coat the squid rings with the breadcrumbs	Deep fry the squid in oil	Remove the shrimp from the oil
MGCN	Cut the squid into slices	Add the egg and egg to the bowl	Coat the squid rings in the batter and then the batter	Add cayenne pepper paprika salt and pepper to a bowl of flour and mix	Coat the squid rings in the batter and breadcrumbs	Deep fry the squid in the oil	Remove the squid rings from the oil and place on a paper towel

Figure 3: The qualitative results. We list the captions for each segment for groundtruth, baseline (transformer model) and our MGCN. The green token are the good cases showing the accurate results through long or short context. The yellow token is the bad case presenting the incorrect case without considering context. The blue token is the inaccurate case by both methods. The bold token is the referenced context information for generating the correct tokens.

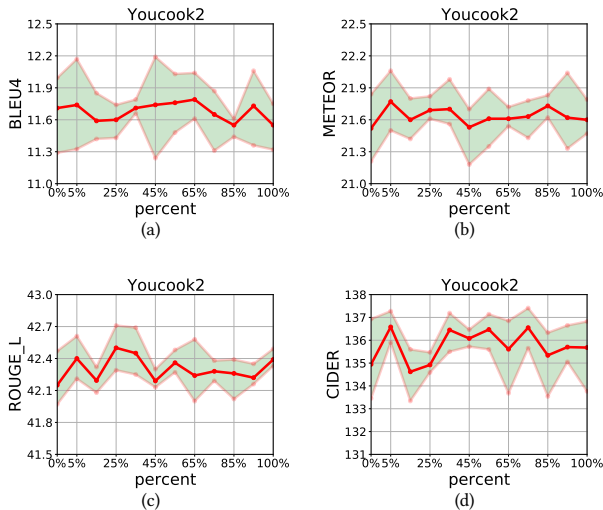


Figure 4: Ablation study of percentage for similar nodes selection. We run experiment 3 times and show the score range(dark red line) as well as the average score(bright red line).

than the useful ones, hence the video event captioning performance dropped.

5.5 Qualitative Analysis

Figure 3 presents a typical showcase. From this case, we can see that our proposed MGCN model can learn the "bow" and "squid rings"

through both the video clip in the current event and the global context. The "bow" in the 4_{th} event is generated with the global context features learned from the 2_{nd} event. Moreover, the "squid rings" in the 7_{th} event benefits from the global context features learned from the 3_{rd} event. Meanwhile, we can see that the MGCN model without using GCN method predicted "shrimp" by mistake which is hard to recognize directly from the current event without consideration of global context information. These results demonstrate that modeling the long-dependency to capture the global context information will greatly benefit the video event captioning. Besides, in these cases, we also noticed that the key ingredient "flour" in the 2_{nd} event is missed in the two predicted captions. It means that the fine-grained object recognition is still a very challenging problem to explore in the future.

6 CONCLUSION AND DISCUSSION

Generating coherent captions from a provided list of video segments can be very challenging to generate accurate and coherent captions. Therefore how to effectively use the context information is an essential research topic. Although previous attempts show that sequential local context is important, it is even challenging to model long-dependency and capture the global context information which is important to video event captioning task. Thus, in this work, we propose a multi-modal graph neural network for dense video captioning, called MGCN, which designs a heterogeneous event-specific graph. Based on the graph, a GCN is used to model the long-dependency between events and capture global context information to enhance the feature representation, and finally benefit the generating of video event caption. Our intensive experimental results demonstrate that our MGCN model can learn better context and achieves SOTA results.

REFERENCES

- [1] Mohit Bajaj, Lanjun Wang, and Leonid Sigal. 2019. G3graphground: Graph-based language grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4281–4290.
- [2] Satandeep Banerjee and Alon Lavie. 2005. 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*. 65–72.
- [3] Shizhe Chen, Yuqing Song, Yida Zhao, Qin Jin, Zhaoyang Zeng, Bei Liu, Jianlong Fu, and Alexander Hauptmann. 2019. ActivityNet 2019 task 3: Exploring contexts for dense captioning events in videos. *arXiv preprint arXiv:1907.05092* (2019).
- [4] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. 2020. Fine-grained video-text retrieval with hierarchical graph reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10638–10647.
- [5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).
- [6] Simon Geng, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. 2020. Coot: Cooperative hierarchical transformer for video-text representation learning. *arXiv preprint arXiv:2011.00597* (2020).
- [7] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216* (2017).
- [8] Jack Hessel, Bo Pang, Zhenhai Zhu, and Radu Soricut. 2019. A Case Study on Combining ASR and Visual Features for Generating Instructional Video Captions. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*.
- [9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [10] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R Hershey, Tim K Marks, and Kazuhiko Sumi. 2017. Attention-based multimodal fusion for video description. In *Proceedings of the IEEE international conference on computer vision*. 4193–4202.
- [11] Yifei Huang, Yusuke Sugano, and Yoichi Sato. 2020. Improving action segmentation via graph-based temporal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14024–14034.
- [12] Vladimir Iashin and Esa Rahtu. 2020. Multi-modal Dense Video Captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 958–959.
- [13] Pin Jiang and Yahong Han. 2020. Reasoning with heterogeneous graph alignment for video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 11109–11116.
- [14] Chenchen Jing, Yuwei Wu, Mingtao Pei, Yao Hu, Yunde Jia, and Qi Wu. 2020. Visual-Semantic Graph Matching for Visual Grounding. In *Proceedings of the 28th ACM International Conference on Multimedia*. 4041–4050.
- [15] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations* (2015).
- [16] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.
- [17] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*. 706–715.
- [18] Jie Lei, Liwei Wang, Yelong Shen, Dong Yu, Tamara Berg, and Mohit Bansal. 2020. MART: Memory-Augmented Recurrent Transformer for Coherent Video Paragraph Captioning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2603–2614.
- [19] Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. 2018. Jointly localizing and describing events for dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7492–7500.
- [20] Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 605.
- [21] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. 2020. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *CVPR*.
- [22] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. *ICCV* (2019).
- [23] Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, and Bohyung Han. 2019. Streamlined dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6588–6597.
- [24] Boxiao Pan, Haoye Cai, De-An Huang, Kuan-Hui Lee, Adrien Gaidon, Ehsan Adeli, and Juan Carlos Niebles. 2020. Spatio-temporal graph for video captioning with knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10870–10879.
- [25] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 311–318.
- [26] Jungin Park, Jiyoung Lee, Ig-Jae Kim, and Kwanghoon Sohn. 2020. SumGraph: Video Summarization via Recursive Graph Modeling. In *16th European Conference on Computer Vision, ECCV 2020*. Springer, 647–663.
- [27] Jae Sung Park, Marcus Rohrbach, Trevor Darrell, and Anna Rohrbach. 2019. Adversarial inference for multi-sentence video description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6598–6608.
- [28] Tanzila Rahman, Bicheng Xu, and Leonid Sigal. 2019. Watch, listen and tell: Multi-modal weakly supervised dense event captioning. In *Proceedings of the IEEE International Conference on Computer Vision*. 8908–8917.
- [29] Botian Shi, Lei Ji, Yaobo Liang, Nan Duan, Peng Chen, Zhendong Niu, and Ming Zhou. 2019. Dense Procedure Captioning in Narrated Instructional Videos. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*. 6382–6391.
- [30] Botian Shi, Lei Ji, Zhendong Niu, Nan Duan, Ming Zhou, and Xilin Chen. 2020. Learning Semantic Concepts and Temporal Alignment for Narrated Video Procedural Captioning. In *Proceedings of the 28th ACM International Conference on Multimedia*. 4355–4363.
- [31] Zheng Shou, Dongang Wang, and Shih-Fu Chang. 2016. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1049–1058.
- [32] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. 2019. Contrastive Bidirectional Transformer for Temporal Representation Learning. *arXiv preprint arXiv:1906.05743* (2019).
- [33] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. Videobert: A joint model for video and language representation learning. *Proceedings of the IEEE international conference on computer vision* (2019).
- [34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
- [35] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4566–4575.
- [36] Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. 2018. Bidirectional attentive fusion with context gating for dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7190–7198.
- [37] Xiaolong Wang and Abhinav Gupta. 2018. Videos as space-time region graphs. In *Proceedings of the European conference on computer vision (ECCV)*. 399–417.
- [38] Xin Wang, Yuan-Fang Wang, and William Yang Wang. 2018. Watch, listen, and describe: Globally and locally aligned cross-modal attentions for video captioning. *arXiv preprint arXiv:1804.05448* (2018).
- [39] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 305–321.
- [40] Yilei Xiong, Bo Dai, and Dahua Lin. 2018. Move forward and tell: A progressive generator of video descriptions. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 468–483.
- [41] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. 2020. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10156–10165.
- [42] Runhao Zeng, Wenbing Huang, Minghui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. 2019. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7094–7103.
- [43] Bowen Zhang, Hexiang Hu, and Fei Sha. 2018. Cross-modal and hierarchical modeling of video and text. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 374–390.
- [44] Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zheng-Jun Zha. 2020. Object relational graph with teacher-recommended learning for video captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 13278–13288.
- [45] Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018. Towards automatic learning of procedures from web instructional videos. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- [46] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. 2018. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8739–8748.
- [47] Andrew Zisserman, Joao Carreira, Karen Simonyan, Will Kay, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, et al. 2017. The kinetics human action video dataset. (2017).