Received: 4 January 2023 | Revised: 22 March 2023 | Accepted: 16 April 2023

DOI: 10.1049/ipr2.12819

ORIGINAL RESEARCH

Dense video captioning based on local attention

Yong Qian[1] | Yingchi Mao[1,2] | Zhihao Chen[1] | Chang Li[1] | Olano Teah Bloh[1] | Qian Huang[1,2]

[1] School of Computer and Information, Hohai University, Nanjing, China

[2] Key Laboratory of Water Big Data Technology of Ministry of Water Resources, Hohai University, Nanjing, China

Correspondence
Yingchi Mao, School of Computer and Information, Hohai University, Nanjing 210098, China.
Email: yingchimao@hhu.edu.cn

Funding information
Key Research and Development Program of China, Grant/Award Number: 2022YFC3005401; Key Research and Development Program of Yunnan Province, Grant/Award Numbers: 202203AA080009, 202202AF080003; the Key Technology Project of China Huaneng Group, Grant/Award Number: HNKJ20-H46

Abstract

Dense video captioning aims to locate multiple events in an untrimmed video and generate captions for each event. Previous methods experienced difficulties in establishing the multimodal feature relationship between frames and captions, resulting in low accuracy of the generated captions. To address this problem, a novel Dense Video Captioning Model Based on Local Attention (DVCL) is proposed. DVCL employs a 2D temporal differential CNN to extract video features, followed by feature encoding using a deformable transformer that establishes the global feature dependence of the input sequence. Then DIoU and TIoU are incorporated into the event proposal match algorithm and evaluation algorithm during training, to yield more accurate event proposals and hence increase the quality of the captions. Furthermore, an LSTM based on local attention is designed to generate captions, enabling each word in the captions to correspond to the relevant frame. Extensive experimental results demonstrate the effectiveness of DVCL. On the ActivityNet Captions dataset, DVCL performs significantly better than other baselines, with improvements of 5.6%, 8.2%, and 15.8% over the best baseline in BLEU4, METEOR, and CIDEr, respectively.

1 | INTRODUCTION

Video captioning is a particularly challenging task in computer vision, aiming to generate a meaningful sentence according to the content of a short video [1–3]. In real-life scenarios, videos are often long, untrimmed, and comprise multiple events. In such situations, conventional video captioning methods are inclined to perceive fewer details since they generate only one sentence to describe the entire video. As a result, dense video captioning is developed [4, 5] with the purpose of generating a sentence for each event in the video.

Generally, dense video captioning follows a two-stage procedure, that is, event proposal and caption generation. Event proposal first predicts a set of event segments, and caption generation subsequently translates the results of the event proposal into natural language. Therefore, dense video captioning needs to focus not only on the temporal motion features of the video and the visual information of each frame, but also on establishing the multimodal feature relationship between frames and captions. Encoder-decoder framework is often applied to dense video captioning [6–8]. The encoders using Convolutional Neural Networks (CNNs) can effectively extract visual and motion features of videos but tend to ignore the context in the temporal dimension. In addition, decoders cannot directly establish the feature relationship between frames and captions, resulting in low accuracy and poor readability of the generated captions.

Recurrent Neural Networks (RNNs) are impressive at processing sequences with contextual relationships in the temporal dimension. Nevertheless, for videos with long durations, RNNs suffer from gradient disappearance and explosion, failing to establish the long-term dependencies of the input video. Previous approaches [9, 10] used the improved Long Short-Term Memory (LSTM) network in the encoder, which partially solved the long-term dependency problem. As compared to CNNs, however, the LSTM-based encoders are inefficient in processing fine-grained interactive motion in the video, resulting in lower accuracy of the generated captions.

In this paper, we utilize a CNN for initial feature extraction from video frames due to its superior image processing performance, consistent with the approach taken in most of the previous works. To compensate for CNN's inadequacies in building long-term dependency on the temporal dimension, we incorporate an attention mechanism while encoding the extracted features. In the decoder, the attention mechanism is likewise efficacious in establishing multimodal feature relationships between frames and captions [11]. Global attention considers all of the encoder's hidden states, but it significantly increases the computational cost and reduces the speed of model convergence. Instead, we introduce the local attention mechanism, which not only substantially decreases computational complexity but also allows the model to focus better on the video sequence at the corresponding input position, thus generating captions with higher accuracy.

Specifically, this paper proposes a novel model for Dense Video Captioning Based on Local Attention (DVCL). To enable the simultaneous perception of local motion and visual information in a single sampled frame, we first introduce a 2D temporal differential CNN. The extracted features are then fed into a deformable transformer to construct global feature dependencies. In this way, we are well-positioned to complete the video coding of dense video captioning, which is to ensure sufficient video features and context dependencies. Considering that the quality of generated captions and the performance of the event proposal are inextricably linked, Distance Intersect-over-Union (DIoU) and Tightness-aware Intersect-over-Union (TIoU) are incorporated into the event proposal match algorithm and evaluation algorithm during training. This allows us to focus on the location and completeness of event segments and improve the accuracy of the event proposal. Furthermore, for the critical step of establishing the multimodal feature relationship between frames and captions, we propose an LSTM based on local attention for generating captions so that each word in the captions can correspond to the relevant frame to improve the accuracy of captions.

The main contributions of this paper are summarized as follows:

1) We propose a local attention-based LSTM that dynamically focuses on frame features corresponding to each generated word, establishing context dependencies between frames and words to improve caption accuracy;
2) During training, we add DIoU and TIoU to the event proposal match algorithm and evaluation algorithm, respectively, to obtain more accurate event proposals and thus improve the quality of the generated captions;
3) We conduct extensive experiments on ActivityNet and YouCook2, and by comparing the result with the existing baselines, our model achieves higher state-of-the-art performance.

2 | RELATED WORK

Existing approaches to video captioning can be divided into two categories: template-based approaches and sequence-based approaches. Template-based approaches [12–14] require predetermined sentence templates, such as SVO, and subsequently capture semantic information and fill the corresponding words into the template to generate captions. While capable of generating syntactically accurate sentences, these methods are limited in their linguistic diversity due to their reliance on a single syntax. Sequence-based approaches [15–17], which leverage advancements in deep learning, utilize CNNs and RNNs to generate sentences with more flexible syntactic structures, and are currently the mainstream approaches for video captioning. Venugopalan et al. [18] modeled video captioning as a Seq2Seq task, extracted frame features by 2D convolution, and used LSTM networks to construct an encoding-decoding model to establish the frame-to-text feature relationship. Li et al. [19] draw on the attention mechanism in the image captioning task to build attention on input sequences along the temporal dimension. This is achieved by computing the attention weight of each feature as the output probability of the generated words. Shetty et al. [20] employed separate models for different types of features during training and utilized evaluation networks to determine the correlation between video features and captions. The model's final output was determined by selecting the model with the highest correlation.

All the above methods assume that the videos contain only a single event, but in reality, most videos are untrimmed and contain multiple events. Traditional video captioning methods cannot describe every event in a clear video, so Krishna et al. [21] proposed the first dense video captioning model. They divided the candidate events into short and long events, and each candidate event was represented by an independent feature as the input to the caption generation model. They then utilized the contextual relationship of adjacent events to generate captions. Shen et al. [22] proposed a model based on region-sequence, which first produced a series of region-sequences for the input videos and then generated the corresponding text. Wang et al. [23] proposed a bidirectional proposal method that can use past and future contexts, and subsequently fused hidden states and video features for generation of descriptions using a context gate. All the above methods utilized event-level contextual relationships, but do not consider the feature relationships between video frames. Wang et al. [24] extracted event-level, scene-level, and frame-level features as input to the description generation module, respectively, and generated text by combining feature information at different levels, but failed to focus on the contextual relationships of the graphical text. Li et al. [25] combined two modules, event proposal and caption generation, and reinforcement learning was incorporated to improve the model's performance. Suin et al. [26] employed reinforcement learning to select effective frames and reduce the computational cost. However, this approach led to fewer extracted features compared to other methods and lower accuracy in the generated captions.

Vision Transformer [27] demonstrated that the transformer can effectively extract video features. Lin et al. [28] migrated it to the video captioning by using transformer-based video coding instead of the traditional 2D and 3D coding modes, which facilitated the adaptation to variable-length videos and eliminated the need to redesign the backbone network for different frame rates. Zhou et al. [29] proposed an end-to-end model named MT, which used a masked transformer in the caption

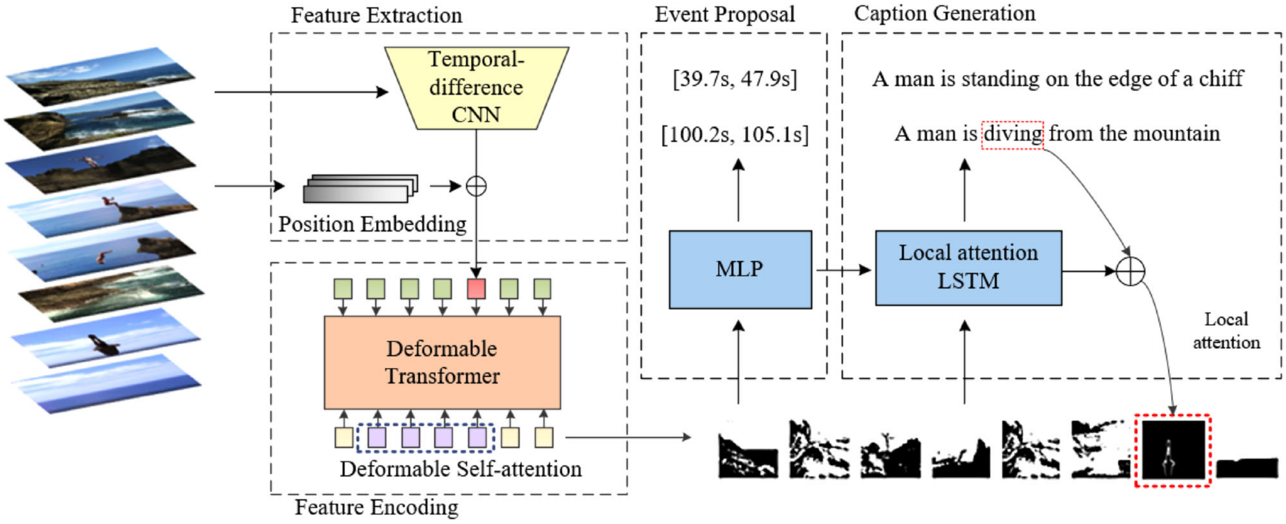

FIGURE 1 Overview of the proposed DVCL. First, we adopt a 2D temporal differential network to extract the video features and then input them to a deformable transformer for encoding. The following event proposal module acquires visual features from the encoder and decodes them to generate multiple event proposals. The final caption generation module based on local attention, enables each word in the generated captions corresponds to the corresponding frame to improve caption accuracy.

decoder module to transform the proposal events generated in the event proposal into a micro-maskable code. Although these approaches are end-to-end, they rely heavily on manual components. Therefore, Zhang et al. [11] designed a pure end-to-end model with parallel decoding, which established a global attention through a transformer that adaptively aggregates contextual feature information into the decoder and guided the generation of captions. While this method addressed the challenges of unifying modeling for two modalities and incorporating attention to temporal feature relationships, the use of global attention resulted in parameter redundancy and inaccurately corresponds the generated words to specific frames. This led to the loss of key information in the generated captions, and consequently, poor readability. The memory-augmented transformer is prevalent in video paragraph captioning [30, 31], where the memory-augmented module improves the coherence of the generated sentences. For dense video captioning; however, this may blur the boundary position of event proposal.

In summary, the establishment of the feature relationship between frames and captions is crucial for dense video captioning. Since the attention mechanism can establish the graphical dependency, we propose a dense video captioning model based on local attention, which enables the word generation to focus dynamically on the corresponding input frame features and obtain the captions with higher accuracy.

3 | METHOD

The proposed dense video captioning model based on local attention consists of four parts: feature extraction, feature coding, event proposal, and caption generation. Its structure is shown in Figure 1. The untrimmed video is extracted by a 2D temporal differential network for features and then input them to a deformable transformer to encode. The event proposal module acquires visual features from the encoder and decodes them to generate multiple event proposals. The caption generation module takes the output of the event proposal and video encoding as input, and utilizes local attention to ensure that each word in the generated captions corresponds to the relevant frame. Each module of the model is described in detail in the following section.

3.1 | Feature extraction

3.1.1 | Temporal differential network

In dense video captioning, dense sampling and coding of spatiotemporal information of all frames directly consume a large amount of computational resources, and the size of GPU memory limits the training speed of the model as the video length increases [32]. Since the visual features of adjacent frames are close to each other, with significant visual changes only around moving objects, we opt for a sparse sampling algorithm. A 2D convolution-based temporal differential encoder is built in order to extract features by capturing short-term motion information using temporal differential operators.

The sparse sampling only selects a fixed and limited number of frames in the video, which makes the model computationally invariant during the training process. The sampled frames are uniformly distributed in the temporal dimension, covering the whole video. In addition to the visual information extracted from the sampled frames, the encoder adds the temporal motion information before and after each frame, so that the sampled frames have both motion and visual features. This facilitates the long-distance dependence modeling of the video frame sequences in the subsequent step.

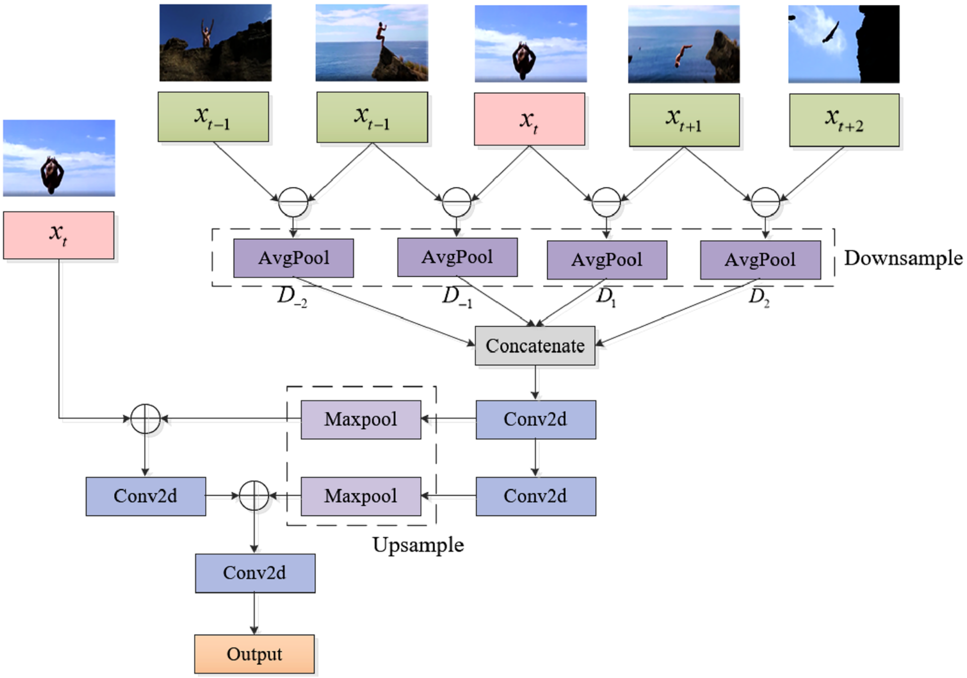

FIGURE 2 The details of the 2D temporal differential CNN. There are two main stages in the 2D temporal differential CNN, a down-sampling and an up-sampling. The former's goal is to effectively extract motion information while compressing the size of images, and the latter is meant to combine with the sampling frame such that single frame contains both visual and motion information.

The temporal differential network first uses the average pooling layer to down-sample each frame difference, compressing the image size of each frame to half of the original image, and then extracts the image features by 2D convolution. Finally, the max pooling layer is employed to upsample the extracted features to the same size as the sampled frames, and then merge them with the sampled frames to form a residual network structure that outputs the feature map. Its network structure is shown in Figure 2.

3.1.2 | Frame sampling and feature fusion

When sampling the original video, it is first divided into T equal segments without overlap, and then a frame x_t is randomly selected from each segment to form a set of $\mathrm{X} = [x_1, x_2, \ldots, x_T]$, where the dimension of X is $[T, C, H, W]$. Compared to sampling frames at a fixed location, random sampling increases the diversity of training and allows the network to learn different instances of the same event. These sampled frames are extracted by the 2D convolutional neural network to obtain the feature set $F = [F_1, F_2, \ldots, F_T]$, where F is the feature representation of the hidden layer of the network. The feature information represented by a single frame is shown in Equation (1), in which F_t contributes to the visual image information, and the feature stack $\mathrm{H}(x_t)$ contributes to the local motion information.

$$\hat{F}_t = F_t + \mathrm{H}(x_t), \tag{1}$$

where $\mathrm{H}(x_t)$ is the core of the encoder, which is obtained by extracting the features of n frames around the sampled frame from the average pooling layer and then stacking them, as shown in Equation (2).

$$\mathrm{H}(x_t) = \mathrm{Upsample}\big(\mathrm{CNN}\big(\mathrm{Downsample}(D(x_t))\big)\big), \tag{2}$$

where D denotes the RGB difference between the before and after n frames, centered on the sampled frame x_t, and stacked by channel, which is denoted as $D(I_i) = [D_{-2}, D_{-1}, D_1, D_2]$, and the calculation process follows the principle of low-resolution processing. The final sampled feature sequence $\hat{F}_t \in \mathbb{R}^{T \times C \times H \times W}$ is used as the input of the transformer network.

3.2 | Feature encoding

In order to enable the model to focus on the contextual global feature relations at the same time, we adopt a transformer network based on the multi-head self-attention mechanism for feature encoding. This network calculates the attention weight of each sampled image frame by the multi-head self-attention module, adaptively aggregates the semantic information in the video, and obtains the feature sequence with global self-attention information, as shown in Equation (3).

$$\mathrm{MulAtten}\big(z_q, x_t\big) = \sum_{m=1}^{M} W_m \left[\sum_{k \in \Omega_k} A_{mqk} \cdot W_m' x_k \right], \tag{3}$$

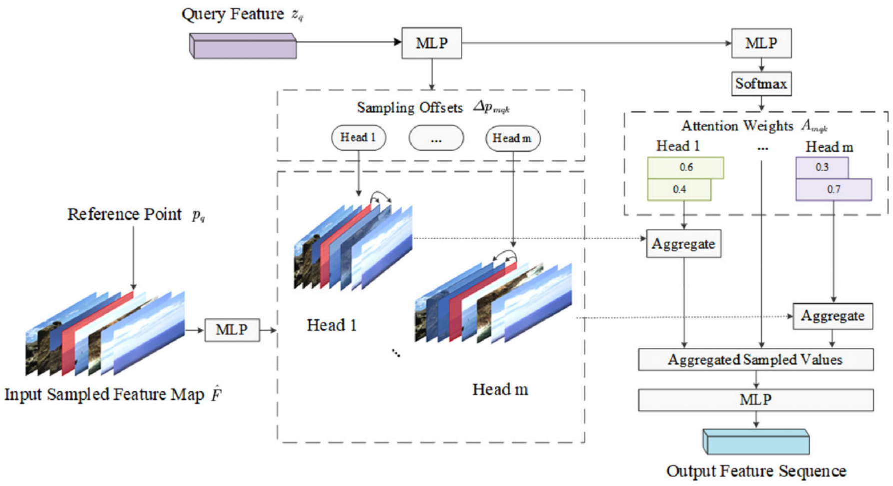

FIGURE 3 The architecture of the deformable transformer. Only some key frames around the current frame p_q are sampled to determine the attention weight, which is then calculated by MLP and finally linearly projected into the query vector.

where x_t is the feature representation for sampled frame, m is the number of multi-head self-attention of the transformer encoder, q and k are respectively the indexes of the query vector z_q and key vector x_k, W_m and W_m' are the weight learnable feature matrices. The self-attention weights A_{mqk} are calculated by Equation (4), and can be normalized to $\sum_{k\in\Omega} A_{mqk} = 1$.

$$\text{Attention}(q, k, v) = \text{soft max}\left(\frac{qk^T}{\sqrt{d_k}}\right)v. \tag{4}$$

For dense video captioning tasks, the input sequences are often long uncropped video clips, which can still have a large amount of data even after sampling. Calculating the global self-attention of each sampled frame consumes huge computational resources. There are many background frames in the video that no event information is contained, calculating the attention on all sampled frames two by two will cause parameter redundancy [33]. At the same time, large input sequences lead to slow gradient descent in training, which requires long training time and large training rounds to make the model converge, and the attention weights cannot focus on specific frames. In order to solve the above problem, we draw on the calculation mode of deformable convolution [34], and introduce the multi-head deformable transformer encoder with the structure shown in Figure 3. The network only samples a set of key frames around the current frame to calculate the attention weights, that is, assigning a certain number of key vectors k to the query vector q of each element in the sequence. The red image in Figure 3 is the current sampled frame, and the other sampled frames are combined with the current frame to calculate the attention weight, as shown in Equation (5):

$$\text{Atten}\left(z_q, p_q, x_t\right) = \sum_{m=1}^{M} W_m \left[\sum_{k=1}^{K} A_{mqk} \cdot W_m' x_k\left(p_q + \Delta p_{mqk}\right)\right], \tag{5}$$

where K is the number of key vectors, p_q is the position reference point of the current frame, Δp_{mqk} and A_{mqk} represents the sampling offset and attention weight of the k-th sample point in the m-th self-attention header, respectively, which are learned by the fully connected layer and finally linearly projected into the query vector. Unlike previous work [35, 36], which derived the attention weights from the dot product, our frame-level attention weights are computed by the fully connected layer.

3.3 | Event proposal

In the event proposal task, the event has characteristics such as a blurred boundary and sparse labeling. In this scenario, a strict one-to-one matching of the real event boundary location with the predicted event boundary location will make it difficult to output optimal results. The accuracy of the model is similar for several different candidate intervals of the same event prediction, and only the position of the beginning and end frames is slightly different, it is difficult to distinguish the superiority from the inferiority. Therefore, in this paper, the distance intersect-over-Union (DIoU) and the tightness-aware intersect-over-Union (TIoU) are added to the proposal matching algorithm and proposal evaluation algorithm, respectively, in the training process of event proposal, to improve the event proposal accuracy by focusing on the location and completeness of the event segment.

3.3.1 | Proposal matching algorithm

After generating the candidate event proposals, we use NMS to remove redundancy and filter the event proposals as the output of this section. The human-labeled event intervals in the dataset are denoted as $\Psi = \{\varphi_n = (t_{s,n}, t_{e,n})\}_{n=1}^{N_g}$, and the event intervals predicted by the event proposal task are denoted as $\hat{\Psi} = \{\hat{\varphi}_n = (\hat{t}_{s,n}, \hat{t}_{e,n})\}_{n=1}^{N}$, where N_g is the number of events in the original video, N is the predicted number of events in each video, $t_{s,n}$ and $t_{e,n}$ are the start and end frame positions of the real events φ_n , and $\hat{t}_{s,n}$ and $\hat{t}_{e,n}$ are the start and end frame positions of the predicted events $\hat{\varphi}_n$.

In the training process of event proposal, the predicted events $\hat{\varphi}_j$ are first matched with the real events φ_j and the interval error is calculated as the loss value optimization model. The objective function of matching is shown in Equation (6), and optimal matching π is calculated by maximizing the objective function.

$$\pi = \arg\max \sum_{i=1}^{N} F_{match}(\varphi_j, \hat{\varphi}_j), \tag{6}$$

where the matching function F_{match} is defined as shown in Equation (7), which consists of two parts, IoU and function L_1, and can focus on both the degree of event overlap and the degree of boundary matching. α is a hyperparameter and specified as 0.1 in experiments.

$$F_{match} = \alpha \mathrm{IoU}(\varphi_j, \hat{\varphi}_j) - L_1(\varphi_j, \hat{\varphi}_j). \tag{7}$$

Traditional event proposal algorithms use the IoU-based objective function as the basis for matching the predicted event interval with the true event interval, as shown in Equation (8). The positive and negative samples are determined by calculating whether the intersection of the two is above a threshold value. The IoU-based matching algorithm can also be used to measure the prediction quality of the event interval, that is, the larger the IoU value, the higher the degree of matching between the prediction interval and the real event interval, and the better the event proposal effect.

$$\mathrm{IoU} = \begin{cases} \dfrac{\min(t_{e,n}, \hat{t}_{e,n}) - \max((t_{s,n}, \hat{t}_{s,n}))}{\max(t_{e,n}, \hat{t}_{e,n}) - \min((t_{s,n}, \hat{t}_{s,n}))} & \varphi_n \cap \hat{\varphi}_n \neq 0 \\ 0 & \varphi_n \cap \hat{\varphi}_n = 0 \end{cases}. \tag{8}$$

However, in the actual event proposal model construction and training process, when there is no intersect between the predicted event interval and the real event interval, that is, $\hat{t}_{s,n} > t_{e,n}$ or $\hat{t}_{e,n} > t_{s,n}$, the value of IoU is 0, which is not conducive to gradient descent of the model. As a result, IoU is hard to optimize the case of disjoint predicted events and real events. It is difficult for the model to converge at the early stage of training, and the final output will have a large number of invalid events that affect the accuracy of the event proposal. Second, IoU can only represent the ratio of intersection to concurrent set, which carries less effective information and cannot reflect the location relationship between two intervals [37]. As shown in Figure 4, the predicted event intervals completely incompatible with the actual events intervals at all, and at this time all IoU are 0. Only the boundary-based objective function L_1 in Equation (7) is valid, which leads to a large event matching error.

In order to make the event proposal pay attention to the event proposal information during the training process, the location function is added to the IoU function, as shown in Equation (9). Speeding up the model convergence by minimizing the normalized distance between the predicted event interval and the real event interval. It is also ensured that the model can converge the predicted event interval to the real event interval even if the predicted events and the real events do not overlap.

$$\mathrm{DIoU} = \mathrm{IoU} - \frac{\rho^2(b, b^{t})}{c^2}, \tag{9}$$

where b and b^{t} represent the coordinates of the center point of the predicted event interval and the real event interval, respectively, while ρ represents the distance between the two points is calculated, c is the length of the smallest time interval that can cover both intervals.

Under the DIoU calculation rule, the DIoU value is 1 when the predicted event interval and the real event interval completely overlap, and when the two intervals do not intersect, DIoU $\rightarrow -1$. Since DIoU can directly optimize the distance between the two intervals, our event proposal module converges faster than traditional IoU-based matching models. With the introduction of DIoU, the two predicted events $\hat{\varphi}_1$ and $\hat{\varphi}_2$ in Figure 4 will be able to match with φ_2, φ_3 and converge toward the corresponding target events, respectively.

3.3.2 | Proposal evaluation algorithm

After matching the real events with the predicted events, the event proposal will be optimized by calculating the gradient with the proposal evaluation algorithm. In order to determine the precision of the predicted event intervals more accurately, a TIoU function is incorporated to the design of the proposal evaluation algorithm. The method focuses on each part of the predicted event intervals to ensure the completeness of the events, and a penalty is appended if other events or background noise are included, avoiding a threshold that determines the final result. For each predicted event interval $\hat{\varphi}_{\sigma(n)}$, the portion of the interval that is missed is denoted as $C_n = \hat{\varphi}_{\sigma(n)} - \hat{\varphi}_{\sigma(n)} \cap \varphi_{\sigma(n)}$, and the TIoU function is shown in Equation (10).

$$\mathrm{TIoU} = \mathrm{IoU} \cdot \left(1 - \frac{\hat{\varphi}_{\sigma(n)} - \hat{\varphi}_{\sigma(n)} \cap \varphi_n}{\hat{\varphi}_{\sigma(n)}}\right). \tag{10}$$

In addition, to maximize the overlap between predicted events and real events, the model should avoid splitting real events and reduce background frames in predicted events. The event proposal evaluation algorithm is designed as a bipartite equation, as shown in Equations (11)–(13).

$$L_{box} = \frac{1}{N} \sum_{n=1}^{N} (L_{bou} + L_{pre}), \tag{11}$$

$$L_{bou} = \|\hat{t}_s^{\sigma(n)} - t_s^{(n)}\|_{l_1} + \|\hat{t}_e^{\sigma(n)} - t_e^{(n)}\|_{l_1}, \tag{12}$$

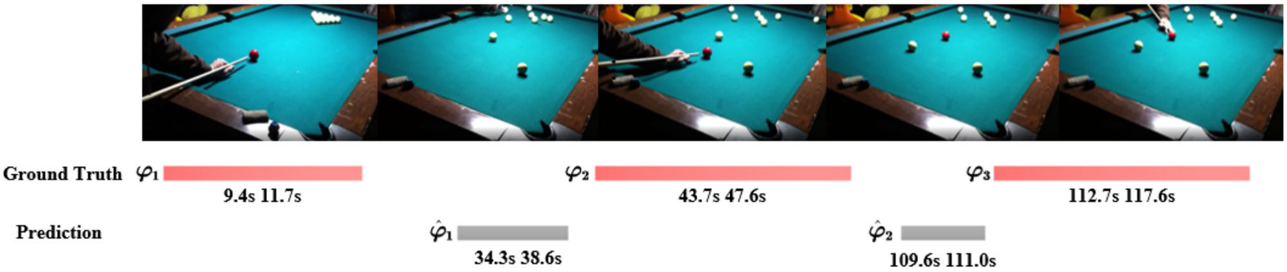

FIGURE 4 An example of a predicted event that does not intersect with the ground truth.

$$L_{pre} = 1 - \mathrm{TIoU}\left(\varphi_n, \hat{\varphi}_{\sigma(n)}\right), \tag{13}$$

where L_{bou} is the boundary loss, which measures the deviation of the starting and ending frames of the predicted event interval from the true event interval, and L_{pre} is the event loss, which measures the accuracy and completeness of the model's predicted event interval.

3.4 | Caption generator

From the event proposal part, we can get the predicted events $\hat{\Psi} = \{(\hat{t}_{s,n}, \hat{t}_{e,n})\}_{n=1}^{N}$, where N is the number of events in each video, $\hat{t}_{s,n}$ and $\hat{t}_{e,n}$ are the start and end frame positions of the *n-th* event, respectively. $\{x_{t,n}\}$ is the sequence of feature representations of each sampled image frame in the current event, and by decoding this sequence, the conditional probability of each word generation is calculated to obtain the description text $\{\boldsymbol{S}_{t',n}\}$ of the corresponding event, as shown in Equation (14). $S_{t'}$ denotes the t'*-th* word of the text, s denotes the input sequence.

$$\log p(S_n | x_n) = \sum_{j=1}^{t} \log p(S_{n,j} | S_{n,<j}, s). \tag{14}$$

To facilitate the computation of attention mechanism and implementation of neural network, the conditional probabilities are parameterized, and the probability of each word can be expressed as:

$$p(S_{n,j} | S_{n,<j}, s) = \mathrm{soft\,max}\,(g(h_j)), \tag{15}$$

$$h_j = f(h_{j-1}, s), \tag{16}$$

where h_j is the hidden layer of the recurrent neural network, and the function f calculates the hidden state at the current position based on the output of the hidden layer at the previous position and the current vector. Subsequently, the output is converted into a vector with the same dimension as the vocabulary by the function g. In order to better capture the contextual temporal relationship of the input sequence, this paper introduces the contextual relationship vector c_t. By splicing c_t and the hidden layer state h_t of the sequence, and then multiplying by the parameter matrix W_c whose weights can be learned [38], we can obtain the hidden state carrying the attention mechanism, as shown in Equation (17).

$$\tilde{h}_t = \tan\mathrm{h}(W_c[c_t : h_t]). \tag{17}$$

Finally, the corresponding word sequences are output by softmax function and fully connected neural network:

$$p(S_{n,j} | S_{n,<j}, s) = \mathrm{soft\,max}\,(W_s \cdot \tilde{h}_t). \tag{18}$$

In order to enable the model to focus on hidden state information and to generate better readable and accurate descriptions, an attention mechanism is added to the caption decoder. Before the introduction of the attention mechanism, c_t can only be used as a semantic vector containing the information of the input sequence; after the introduction of the attention mechanism, c_t at each time step changes dynamically according to the relevance of the current prediction sequence with the input sequence in the encoder, and c_t is recomputed by assigning different weights to the hidden vectors of the input sequence in the encoder at each time step. For example, when describing actions, attention prefers to assign larger weights to the frames in the sequence with larger variations. When attention is focused on the hidden state of the keyword combination vector, it can output statements describing the correlation between keywords, which can not only achieve accurate prediction but also control the semantic distinction between words. The network framework of the attention mechanism is shown in Figure 5.

In generating each target word, the center p_t of current attention is first calculated as shown in Equation (19):

$$p_t = S \cdot sigmoid(v_p^T \tan\mathrm{h}(W_p \cdot h_t)), \tag{19}$$

so that the output word can pay attention to the position of the input sequence associated with itself. The position matrix W_p and the penalty term v_p are the learnable feature parameters of the weights, S is the length of the input sequence, $p_t \in [0, S]$, and the attention window corresponding to the position is $[p_t - D, p_t + D]$. Then the the attention weights are obtained by calculating the hidden layer vectors of the input and output sequences using the align function, which is constrained by a

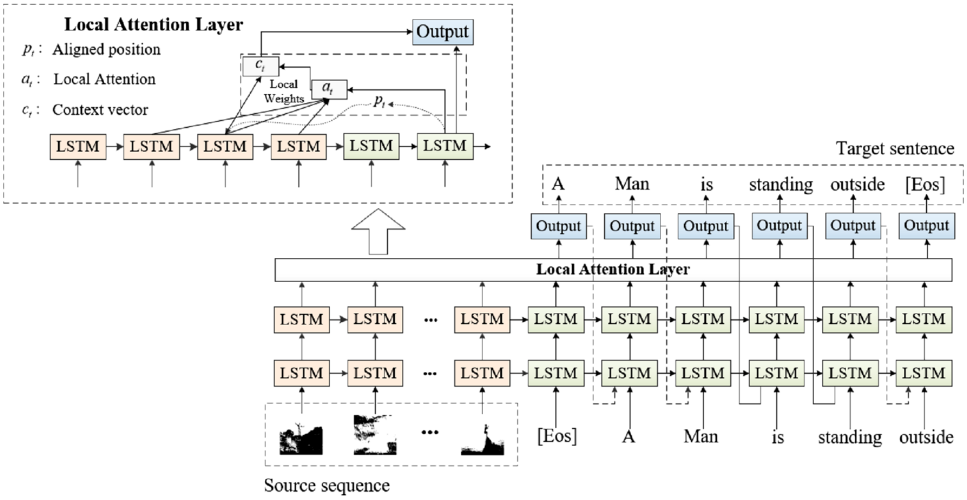

FIGURE 5 The LSTM based on local attention. When generating each target word, the current attention's center position p_t is calculated firstly so that the output word can focus on its relevant input sequence position.

Gaussian distribution, as shown in Equation (20).

$$a_t(s) = align(h_t, h_s) \exp\left(-\frac{(s - p_t)^2}{2\sigma^2}\right). \tag{20}$$

Finally, the LSTM network takes the concatenated input of contextual features, attention weights, and previously generated words, the output words at the current position are calculated by the fully connected network and softmax activation function.

4 | EXPERIMENTS

4.1 | Dataset and evaluation metrics

The performance of DVCL is evaluated on the ActivityNet Captions dataset [21] and the YouCook2 dataset [39]. ActivityNet Captions consists of 20,000 videos and corresponding captions, each video is 120 s and contains 3.65 events on average. The number of events in the videos follows a normal distribution, and increases with the length of the videos. The captions in the dataset contain 13.48 words per sentence on average, and the number of words also conforms to a normal distribution. In addition, there is about 10% overlap between the captions and the corresponding events, indicating that there is a partial intersection of the events in the dataset. This is in line with the application scenario of video captioning tasks. The YouCook2 dataset contains 2000 videos from 89 cooking scenarios. The cooking steps in each video are labeled with the start and end positions, and described by English imperatives. All videos are from YouTube and are shot from the third-person perspective. They are commonly used to verify the accuracy of the video captioning task in single-event scenario. In this paper, the training/validation/test splits for ActivityNet Captions dataset are 10,000/4,900/5,000 while for YouCook2 dataset are 1,300/450/210.

The essence of the dense video captioning task is to extract video feature information, locate all events in the video, and translate features of video into natural language. Therefore, in this paper, BLEU [40], METEOR [41] and CIDEr [42], common in the field of text translation, are used as the evaluation metrics of the model.

4.2 | Baselines

The experiments compare the DVCL model with previous state-of-the-art methods, as described below.

(1) DCE [21]: C3D is used as the backbone network to extract the feature information of the video, and dynamic step size is used to extract the events in the video. Finally an attention mechanism is added to the LSTM network to establish the context dependence of different events, ensuring the accuracy of the captions.

(2) MT [29]: The transformer network is used to implement end-to-end training for the video captioning task. It is verified that the self-attention mechanism can establish cross-modal feature relationships between text and images.

(3)) DVC [25]: The model predicts event and descriptive scores to filter events with higher complexity as candidate, and the coherence of the generated sentences is then improved through reinforcement learning.

(4) Efficient [26]: The caption generation in the work is reformulated as a classification problem. Given the input sequence and the context vector, the model selects a target from a series of candidate sentences. The model improves the accuracy of the generated captions by unsupervised learning.

5) ECHR [24]: An event-centered hierarchical feature representation is used to extract event-level, scene-level and frame-level features as input to the caption generation

TABLE 1 Hyperparameter setting of DVCL.

Parameters	Value
Self-attention head size	8
Transformer decoder network hidden size	512
Transformer decoder feed-forward layer size	512
Local attention LSTM window size	5
Dropout rate	0.5
Batch size	32
Epoch size	30
Learning rate	5e-05
Weight decay	0.0001
Optimizer	Adam

module, and sentences are generated by combining feature information at different levels.

(6) PDVC [11]: An end-to-end network structure with parallel decoding is used to generate both candidate events and captions. This method solves the problem that the back-propagation of the caption generation model cannot directly guide the optimization of the event proposal model.

4.3 | Experiment setup

The hyperparameters of DVCL are obtained by grid search on the validation set, and the values are shown in Table 1. In the image frame sampling part, all videos are uniformly cropped to 224 × 224 pixels and each image block is divided into 16 × 16 pixels with a frame rate of $1/32$. The transformer network contains eight self-attention heads, 512-dimensional hidden layers, and 2048-dimensional feedforward neural networks. The dropout rate is 0.5, and ReLU is used as the activation function. In the caption generation part, we adopt the LSTM network structure based on a local attention mechanism with 512-dimensional hidden layers and a window size of 5. The final word vector is output by a 1-layer, 512-dimension fully connected network with a softmax activation function.

4.4 | Result analysis

4.4.1 | Comparative analysis with existing methods

To demonstrate the performance of DVCL, it is compared with six baselines mentioned above, and the results are shown in Table 3. From Table 3, we can find that DVCL performs significantly better than other baselines on the ActivityNet Captions dataset. It achieves optimal results in three metrics, including BLEU4, METEOR, and CIDEr, with improvements of 5.6%, 8.2%, and 15.8%, respectively, over the best baseline.

The experimental results show that the DCE model pays no attention to the frame-level contextual information of the input video, leading to the generation of numerous number of redundant candidate events,and thus resulting in the generated captions with poor readability. The MT model uses a self-attention mechanism to capture the contextual feature relationships of the video, and combines the visual information of the image when generating the captions, so the accuracy is improved to a certain extent. DVC adopts the method of assigning scores to events and sorting to filter them, which effectively solves the problem of redundancy in extracting events from the model. Adding reinforcement learning to the caption generation module also greatly improves the readability of the text, resulting in significant improvement in both METEOR and CIDEr. Efficient models the video captioning task as a classification task and introduces unsupervised learning to optimize the task, which improves the readability and accuracy of the text. But the algorithm neglects the extraction of visual features, so the performance improvement is limited. The ECHR model utilizes an event-centered hierarchical feature representation and establishes interdependence between events and text, which enhances models performance. However, due to the lack of global dependency between image frames and events, the model accuracy is lower than PDVC. The feature modeling network in PDVC is a transformer-based decoder, which can establish long-range global feature dependencies, and the soft-attention-based LSTM network effectively leverages the self-attention information computed by the transformer network. PDVC verifies that the transformer network can effectively model the image visual information features, and the end-to-end training model can optimize the event extraction model by calculating the loss of the caption generation task. The performance of PDVC is slightly lower than that of DVCL, because the local attention mechanism in DVCL can make each generated word dynamically focus on several frames of related input images, which compensates for the inability that the model fails to directly calculate the feature relationship between words and frames.

The accuracy of the event proposal affects the quality of generated descriptions. The event proposal algorithms based on TIoU and DIoU enable DVCL to better localize the events in the video, as shown in Figure 6. The generated sentences also contain richer semantic information because of model's ability to directly establish the mulitimodal feature relationship.

DVCL achieves the best overall experimental performance, with three significantly improved metrics, such as BLEU4, METEOR, and CIDEr. The reasons for these improvements are: (1) The use of a temporal differential network to extract the features of the video, which incorporates the visual information of each sampled frame and the motion information of several frames around it. (2) The deformable transformer network ensures that the model can construct global feature dependencies, improving the network's ability to learn semantic information. (3) The LSTM network based on the local attention mechanism enables the model to dynamically focus on the input image features at the corresponding position when generating each word, and to establish contextual dependencies to improve the accuracy of caption generation.

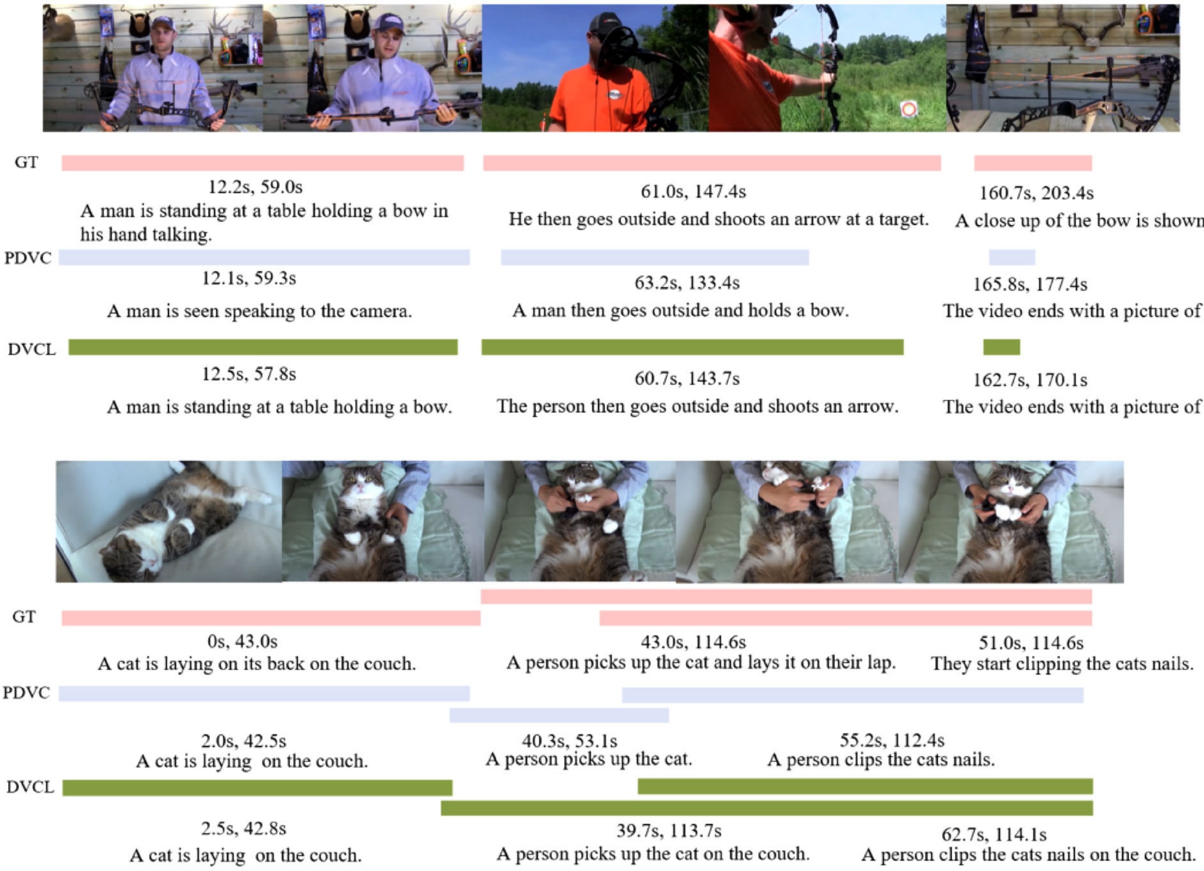
GT
12.2s, 59.0s
A man is standing at a table holding a bow in his hand talking.
61.0s, 147.4s
He then goes outside and shoots an arrow at a target.
160.7s, 203.4s
A close up of the bow is shown.
PDVC
12.1s, 59.3s
A man is seen speaking to the camera.
63.2s, 133.4s
A man then goes outside and holds a bow.
165.8s, 177.4s
The video ends with a picture of the video.
DVCL
12.5s, 57.8s
A man is standing at a table holding a bow.
60.7s, 143.7s
The person then goes outside and shoots an arrow.
162.7s, 170.1s
The video ends with a picture of the bow.
GT
0s, 43.0s
A cat is laying on its back on the couch.
43.0s, 114.6s
A person picks up the cat and lays it on their lap.
51.0s, 114.6s
They start clipping the cats nails.
PDVC
2.0s, 42.5s
A cat is laying on the couch.
40.3s, 53.1s
A person picks up the cat.
55.2s, 112.4s
A person clips the cats nails.
DVCL
2.5s, 42.8s
A cat is laying on the couch.
39.7s, 113.7s
A person picks up the cat on the couch.
62.7s, 114.1s
A person clips the cats nails on the couch.

FIGURE 6 Qualitative demonstration of generated captions on the ActivityNet Captions set.

TABLE 2 Comparison of experimental results for single-event scenario.

	ActivityNet Captions			YouCook2		
Model	**BLEU4**	**METEOR**	**CIDEr**	**BLEU4**	**METEOR**	**CIDEr**
DCE [21]	1.60	8.88	25.12	0.4	3.01	4.98
MT [29]	2.71	11.16	47.71	0.30	3.18	6.10
ECHR [24]	1.96	10.58	39.73	–	3.82	–
PDVC [11]	3.07	11.27	52.53	0.80	4.74	22.71
DVCL	**3.32**	**11.25**	**56.94**	**1.12**	**6.26**	**25.46**

TABLE 3 Comparison with state-of-the-art methods on the ActivityNet Captions dataset.

Model	**BLEU4**	**METEOR**	**CIDEr**
DCE [21]	0.17	5.69	12.43
MT [29]	1.15	4.98	9.25
DVC [25]	0.73	6.93	12.61
Efficient [26]	1.35	6.21	13.82
ECHR [24]	1.29	7.19	14.71
PDVC [11]	1.78	7.96	25.87
DVCL	**1.88**	**8.61**	**29.97**

4.4.2 | Single-event scenario comparison analysis

The above section verifies that DVCL can accurately locate events and generate the corresponding descriptions in dense video captioning task. This section splits the videos in the ActivityNet Captions dataset so that each video contains only a single event and no background frames. It is to verify the accuracy of the model only generates the corresponding descriptive text without event proposal. The accuracy of the model is also verified in the YouCook2 dataset for single-event scenario. From Table 2, we can find that DVCL generates the best text accuracy compared to other baselines for single-event scenario of videos. This directly proves that DVCL can not only effectively extract the rich semantic information in the video, but also the LSTM network based on the local attention mechanism ensures that the generated words can correspond to the feature information, resulting in a more accurate descriptive text.

4.4.3 | Ablation studies

DVCL consists of four components: 2D temporal differential convolution-based feature extraction; deformable transformer-based feature coding; improved event proposal in the training process; and local attention-based caption generation. In order

TABLE 4 Results of ablation studies.

Model	BLEU4	METEOR	CIDEr
DVCL	**1.88**	**8.61**	**29.97**
-Temporal Difference Conv	1.36	6.90	24.91
-Deformable Transformer	1.79	7.99	26.01
-TIoU&DIoU	1.66	7.89	26.85
-Local attention	1.27	6.74	22.81

to determine the impact of these four components on the overall model performance and generated text quality, one of them is removed or swapped, respectively, and compared with the original model effect. For the 2D temporal differential convolution-based feature extraction, we replaced it with random frame sampling. For the video coding model, we removed the dynamic attention extraction part and used the transformer network structure for feature coding. For the event proposal, we replaced the DIoU-based matching algorithm and the TIoU-based evaluation algorithm with the traditional IoU computation model. For the caption generator, the local attention mechanism is removed, only the vanilla LSTM network is used to generate captions.

From Table 4, it can be revealed that adding local attention to the caption generation model has the greatest impact on the quality of the text generated by the model. When the vanilla LSTM network is adopted, the BLEU4, METEOR and CIDEr metrics decrease by 32.4%, 16.9% and 23.9%, respectively. The results show that the LSTM network can only focus on event-level feature information and cannot capture the correspondence between words and frames, leading to insufficient semantic information for the model. When the 2D temporal differential convolution is replaced with random frame sampling, the BLEU4, METEOR, and CIDEr metrics decrease by 27.7%, 14.9%, and 16.9%, respectively, trailing only the LSTM network based on the local attention mechanism. It is evident that the temporal differential convolution can effectively extract the visual features and motion information of the video. When traditional IoU-based algorithm is used, the indicators respectively decreased by 11.7%, 2.7%, and 10.4%, indicating that the event proposal algorithm can capably improve the event proposal accuracy and thus the quality of the generated captions. The BLEU4, METEOR, and CIDEr metrics of the model decrease by 4.0%, 1.5%, and 15.2%, respectively, after removing the deformable attention and keeping only the transformer self-attention encoder, indicating that the global self-attention causes redundancy in the parameters of the model. Meanwhile, it is difficult to converge to the global optimum during the training process, which makes the caption generator unable to focus on the effective feature information and affects the accuracy of the model.

The comparative experiments on the ActivityNet Captions dataset demonstrate that DVCL is optimal in BLEU4, METEOR, and CIDEr, and outperforms the current baselines significantly. Therefore, it is verified that DVCL can fully extract the visual and motion features of the original video, and improve the accuracy of the captions by capturing the contextual relationship in the temporal dimension.

5 | CONCLUSION

In this paper, we propose a novel dense video captioning model named DVCL to address the problems of low accuracy and poor readability of the generated captions. The feature extraction module employs a 2D temporal differential network, which can effectively extract video features, and subsequently constructs the global feature dependency of the input sequence through a deformable transformer. The event proposal module utilizes TIoU and DIoU to access more accurate proposals. The final caption generator module adopts an LSTM based on local attention, enabling the model to focus dynamically on the input frame features at the corresponding locations when generating words and improving the accuracy of captions. Finally, the experimental results on the ActivityNet Captions and YouCook2 datasets show that the proposed method can significantly improve the quality and accuracy of the generated captions.

AUTHOR CONTRIBUTION

Conceptualization: Yong Qian, Yingchi Mao and Qian Huang. *Methodology*: Yong Qian and Yingchi Mao. *Validation*: Yong Qian and Zhihao Chen. *Formal analysis*: Yingchi Mao and Qian Huang. *Investigation*: Yong Qian, Zhihao Chen, Chang Li and Olano Teah Bloh. *Writing-original draft preparation*: Yong Qian. *Writing-review and editing*: Chang Li and Olano Teah Bloh. *Supervision*: Yingchi Mao. *Project administration*: Yingchi Mao and Qian Huang. *Funding acquisition*: Yingchi Mao.

FUNDING INFORMATION

Funding information: Key Research and Development Program of China, Grant/Award Number: 2022YFC3005401; Key Research and Development Program of Yunnan Province, Grant/Award Number: 202203AA080009, 202202AF080003; the Key Technology Project of China Huaneng Group, Grant/Award Number: HNKJ20-H46

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in [ActivityNet Captions] at [https://cs.stanford.edu/people/ranjaykrishna/densevid/], reference number [21], [YouCook2] at [http://youcook2.eecs.umich.edu/], reference number [39]

ORCID

Yong Qian https://orcid.org/0000-0003-3142-6891
Yingchi Mao https://orcid.org/0000-0002-9884-8100
Qian Huang https://orcid.org/0000-0001-5625-0402

REFERENCES

1. Yan, Y., Zhuang, N., Ni, B., Zhang, J., Xu, M., Zhang, Q., Zhang, Z., Cheng, S., Tian, Q., Xu, Y., Yang, X., Zhang, W.: Fine-grained video captioning via graph-based multi-granularity interaction learning. IEEE Trans. Pattern Anal. Mach. Intell. 44(2), 666–683 (2022)
2. Li, L., Gao, X., Deng, J., Tu, Y., Zha, Z., Huang, Q.: Long short-term relation transformer with global gating for video captioning. IEEE Trans. Image Process. 31, 2726–2738 (2022)
3. Man, X., Ouyang, D., Li, X., Song, J., Shao, J.: Scenario-aware recurrent transformer for goal-directed video captioning. ACM Trans. Multim. Comput. Commun. Appl. 18(4), 104:1–104:17 (2022)
4. Deng, C., Chen, S., Chen, D., He, Y., Wu, Q.: Sketch, ground, refine: Top-down dense video captioning. In: Computer Vision and Pattern Recognition Conference, pp. 234–243. IEEE, Piscataway (2021)
5. Chang, Z., Zhao, D., Chen, H., Li, J., Liu, P.: Event-centric multi-modal fusion method for dense video captioning. Neural Netw. 146, 120–129 (2022)
6. Yu, M., Zheng, H., Liu, Z.: Dense video captioning with hierarchical attention-based encoder-decoder networks. In: International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE, Piscataway (2021)
7. Ji, L., Guo, X., Huang, H., Chen, X.: Hierarchical context-aware network for dense video event captioning. In: ACL/International Joint Conference on Natural Language Processing (IJCNLP), pp. 2004–2013. Association for Computational Linguistics, Stroudsburg, PA (2021)
8. Yu, Z., Han, N.: Accelerated masked transformer for dense video captioning. Neurocomputing 445, 72–80 (2021)
9. Srivastava, N., Mansimov, E., Salakhutdinov, R.: Unsupervised learning of video representations using lstms. In: ICML, ser. vol. 37, pp. 843–852. International Machine Learning Society, Madison, WI (2015). JMLR Workshop and Conference Proceedings (2015)
10. Yu, H., Wang, J., Huang, Z., Yang, Y., Xu, W.: Video paragraph captioning using hierarchical recurrent neural networks. In: Computer Vision and Pattern Recognition Conference, pp. 4584–4593. IEEE Computer Society, Los Alamitos, CA (2016)
11. Wang, T., Zhang, R., Lu, Z., Zheng, F., Cheng, R., Luo, P.: End-to-end dense video captioning with parallel decoding. In: International Conference on Computer Vision, pp. 6827–6837. IEEE, Piscataway (2021)
12. Rohrbach, M., Qiu, W., Titov, I., Thater, S., Pinkal, M., Schiele, B.: Translating video content to natural language descriptions. In: International Conference on Computer Vision, pp. 433–440. IEEE Computer Society, Los Alamitos, CA (2013)
13. Guadarrama, S., Krishnamoorthy, N., Malkarnenkar, G., Venugopalan, S., Mooney, R.J., Darrell, T., Saenko, K.: Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In: International Conference on Computer Vision, pp. 2712–2719. IEEE Computer Society, Los Alamitos, CA (2013)
14. Krishnamoorthy, N., Malkarnenkar, G., Mooney, R.J., Saenko, K., Guadarrama, S.: Generating natural-language video descriptions using text-mined knowledge. In: Proceedings of the AAAI Conference on Artificial Intelligence. AAAI Press, Menlo Park, CA (2013)
15. Jin, T., Zhao, Z., Wang, P., Yu, J., Wu, F.: Interaction augmented transformer with decoupled decoding for video captioning. Neurocomputing 492, 496–507 (2022)
16. Tang, M., Wang, Z., Liu, Z., Rao, F., Li, D., Li, X.: Clip4caption: CLIP for video caption. In: ACM Multimedia, pp. 4858–4862. ACM, New York (2021)
17. Ryu, H., Kang, S., Kang, H., Yoo, C.D.: Semantic grouping network for video captioning. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 2514–2522. AAAI Press, Menlo Park, CA (2021)
18. Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R.J., Darrell, T., Saenko, K.: Sequence to sequence - Video to text. In: International Conference on Computer Vision, pp. 4534–4542. IEEE Computer Society, Los Alamitos, CA (2015)
19. Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C.J., Larochelle, H., Courville, A.C.: Video description generation incorporating spatio-temporal features and a soft-attention mechanism. CoRR, vol. abs/1502.08029 (2015)
20. Shetty, R., Laaksonen, J.: Frame- and segment-level features and candidate pool evaluation for video caption generation. In: ACM Multimedia, pp. 1073–1076. ACM, New York (2016)
21. Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Niebles, J.C.: Dense-captioning events in videos. In: International Conference on Computer Vision, pp. 706–715. IEEE Computer Society, Los Alamitos, CA (2017)
22. Shen, Z., Li, J., Su, Z., Li, M., Chen, Y., Jiang, Y., Xue, X.: Weakly supervised dense video captioning. In: Computer Vision and Pattern Recognition Conference, pp. 5159–5167. IEEE Computer Society, Los Alamitos, CA (2017)
23. Wang, J., Jiang, W., Ma, L., Liu, W., Xu, Y.: Bidirectional attentive fusion with context gating for dense video captioning. In: Computer Vision and Pattern Recognition Conference, pp. 7190–7198. Computer Vision Foundation / IEEE Computer Society, Los Alamitos, CA (2018)
24. Wang, T., Zheng, H., Yu, M., Tian, Q., Hu, H.: Event-centric hierarchical representation for dense video captioning. IEEE Trans. Circuits Syst. Video Technol. 31(5), 1890–1900 (2021)
25. Li, Y., Yao, T., Pan, Y., Chao, H., Mei, T.: Jointly localizing and describing events for dense video captioning. In: Computer Vision and Pattern Recognition Conference, pp. 7492–7500. Computer Vision Foundation / IEEE Computer Society, Los Alamitos, CA (2018)
26. Suin, M., Rajagopalan, A.N.: An efficient framework for dense video captioning. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 12 039–12 046. AAAI Press, Menlo Park, CA (2020)
27. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: 9th International Conference on Learning Representations, ICLR 2021. OpenReview.net (2021)
28. Lin, K., Li, L., Lin, C., Ahmed, F., Gan, Z., Liu, Z., Lu, Y., Wang, L.: Swinbert: End-to-end transformers with sparse attention for video captioning. In: Computer Vision and Pattern Recognition Conference, pp. 17 928–17 937. IEEE, Piscataway (2022)
29. Zhou, L., Zhou, Y., Corso, J.J., Socher, R., Xiong, C.: End-to-end dense video captioning with masked transformer. In: Computer Vision and Pattern Recognition Conference, pp. 8739–8748. Computer Vision Foundation / IEEE Computer Society, Los Alamitos, CA (2018)
30. Lei, J., Wang, L., Shen, Y., Yu, D., Berg, T.L., Bansal, M.: MART: memory-augmented recurrent transformer for coherent video paragraph captioning. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, pp. 2603–2614. Association for Computational Linguistics, Stroudsburg, PA (2020)
31. Cardoso, L.V., Guimarães, S.J.F., do Patrocínio, Z.K.G.: Enhanced-memory transformer for coherent paragraph video captioning. In: 33rd IEEE International Conference on Tools with Artificial Intelligence, pp. 836–840. IEEE, Piscataway (2021)
32. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Gool, L.V.: Temporal segment networks: Towards good practices for deep action recognition. In: European Conference on Computer Vision (ECCV). Lecture Notes in Computer Science, vol. 9912. pp. 20–36. Springer, Cham (2016)
33. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: deformable transformers for end-to-end object detection. In: Proceedings of International Conference on Learning Representation (ICLR). ICML, San Diego (2021)
34. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: International Conference on Computer Vision, pp. 764–773. IEEE Computer Society, Los Alamitos, CA (2017)
35. Choi, W., Chen, J., Yoon, J.: Parallel pathway dense video captioning with deformable transformer. IEEE Access 10, 129899–129910 (2022)
36. Chen, J., Pan, Y., Li, Y., Yao, T., Chao, H., Mei, T.: Temporal deformable convolutional encoder-decoder networks for video captioning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 8167–8174. AAAI Press, Menlo Park, CA (2019)
37. Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., Ren, D.: Distance-iou loss: Faster and better learning for bounding box regression. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 12993–13000. AAAI Press, Menlo Park, CA (2020)

38. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1412–1421. The Association for Computational Linguistics, (2015)
39. Zhou, L., Xu, C., Corso, J.J.: Towards automatic learning of procedures from web instructional videos. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 7590–7598. AAAI Press, Menlo Park, CA (2018)
40. Papineni, K., Roukos, S., Ward, T., Zhu, W.: Bleu: a method for automatic evaluation of machine translation. In: ACL, pp. 311–318. ACL, Stroudsburg, PA (2002)
41. Banerjee, S., Lavie, A.: METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In: ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pp. 65–72. Association for Computational Linguistics, Stroudsburg, PA (2005)
42. Vedantam, R., Zitnick, C.L., Parikh, D.: Cider: Consensus-based image description evaluation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 4566–4575. IEEE Computer Society, Los Alamitos, CA (2015)

How to cite this article: Qian, Y., Mao, Y., Chen, Z., Li, C., Bloh, O.T., Huang, Q.: Dense video captioning based on local attention. IET Image Process. 17, 2673–2685 (2023). https://doi.org/10.1049/ipr2.12819