Video Captioning Using Global-Local Representation

Liqi Yan, Siqi Ma, Qifan Wang, Yingjie Chen, Xiangyu Zhang, Andreas Savakis, Dongfang Liu*

***Abstract*—Video captioning is a challenging task as it needs to accurately transform visual understanding into natural language description. To date, state-of-the-art methods inadequately model global-local vision representation for sentence generation, leaving plenty of room for improvement. In this work, we approach the video captioning task from a new perspective and propose a GLR framework, namely a global-local representation granularity. Our GLR demonstrates three advantages over the prior efforts. First, we propose a simple solution, which exploits extensive vision representations from different video ranges to improve linguistic expression. Second, we devise a novel global-local encoder, which encodes different video representations including long-range, short-range and local-keyframe, to produce rich semantic vocabulary for obtaining a descriptive granularity of video contents across frames. Finally, we introduce the progressive training strategy which can effectively organize feature learning to incur optimal captioning behavior. Evaluated on the MSR-VTT and MSVD dataset, we outperform recent state-of-the-art methods including a well-tuned SA-LSTM baseline by a significant margin, with shorter training schedules. Because of its simplicity and efficacy, we hope that our GLR could serve as a strong baseline for many video understanding tasks besides video captioning. Code will be available.**

***Index Terms*—Computer vision, video captioning, video representation, natural language processing, visual analysis.**

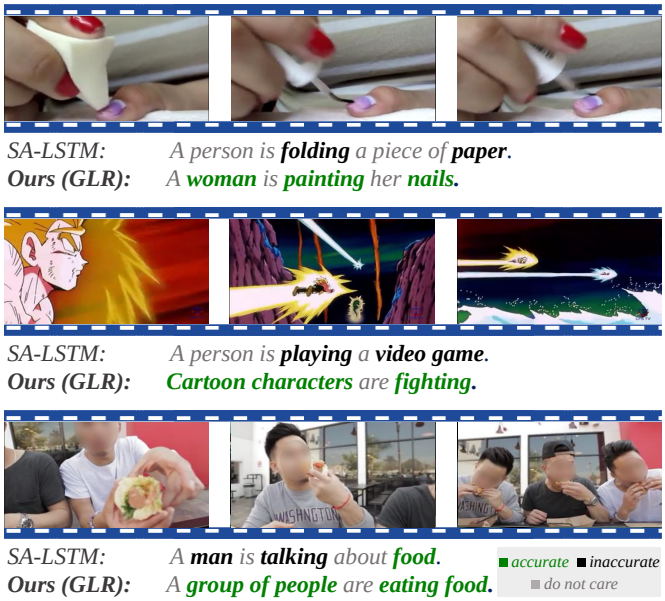


Fig. 1. Comparison with the state-of-the-art method SA-LSTM [1]. The three different examples demonstrate that our method can achieve fine-grained expression in the sentence to accurately describe the video frames.

I. Introduction

Video captioning has great societal value due to a wide array of real-world applications, *e.g.*, subtitle generation, blind person assistance, and autopilot narration [2]–[5]. However, isolated video frames may suffer from motion blur, occlusion, or truncation, which introduces great confusion in visual understanding for the captioning task. To address this above problem, many prior efforts [6]–[9] attempt to answer the principal problem: how to leverage the rich global-local features across video frames to close the gap from visual understanding to language expression?

Despite making significant progress, existing methods for video captioning inadequately capture the local and global representations. Rather than modeling the correlations of semantic entities across frames, a lot of methods simply apply the deep convolutional neural network on raw pixels to build higher-level connections [7], [10]. The primary focus of these methods is to operate on local object features, but this neglects object transformations or interactions [11]–[16]. The approach of modeling local object features is a limiting solution for video captioning, because the temporal connections across frames are not explored delicately and thus are sensitive to spurious associations [12], [13].

To study the problem of the global-local correlation, other related vision tasks leverage the graph representation using graph neural networks (GNNs). For instance, [17], [18] model object relations by using video spatio-temporal graphs and explicitly build links between high-level entities. Specifically, each node encodes a target entity (*i.e.*, objects persons [18]–[20], body joints [21], and actions [17]), while each edge represents correlations among the entities.

Inspired by the above success, recent video captioning studies extend the graph-based approach and use GNNs to model global-local reasoning [22], [23]. Among these works, [11], [18] merge local features with global features using concatenation; [17], [19], [24] add spatio-temporal features as a separate node in the graph. However, empirical results

This work was supported in part by the U.S. National Institute of Health (NIH) under Grant #1R25EY029127.

L. Yan is enrolled at Fudan University, China.
S. Ma and L. Yan are also with Westlake University, China (e-mail: {yanliqi,masiqi}@westlake.edu.cn).
Q. Wang is with Meta AI, USA. This work is done before joining Meta AI. (e-mail: wqfcr@fb.com).
Y. Chen and X. Zhang are with Purdue University, USA (e-mail: {victorchen,xyzhang}@purdue.edu).
A. Savakis and D. Liu are with Rochester Institute of Technology, USA (e-mail: {axseec,dongfang.liu}@rit.edu).
L. Yan is also with Rochester Institute of Technology, USA.
* is corresponding author.

indicate that using graphs to represent global-local correlation is a sub-optimal solution, as it often encounters the over-smoothing problem in training which leads to weak performance in sentence generation.

Alternatively, many video captioning methods intuitively exploit multi-modal fusion ($i.e.$, visual and audio features) to enrich the feature representations for prediction [25], [26]. However, these simple "lumping" approaches inefficiently exploit multi-modal features and encounter difficulty to perform joint optimization cross-modality, leaving large room for improvement.

To this end, we attempt to solve the video captioning in a more flexible approach. Concretely, we make the following contributions in this work:

- We devise a simple framework called the global-local representation granularity (GLR), which uses extensive vision representations for captioning generation.
- We propose a novel global-local encoder, which exploits rich temporal representation for video captioning. We encode the long-range frames to describe spatio-temporal correspondence and short-range frames to capture object motion and tendency, while using a local keyframe to preserve finer object appearance and location details.
- We introduce a progressive training strategy that includes two phases. In the first seeding phase, we propose a novel discriminative cross-entropy that addresses the problem of human annotation discrepancy. In the second boosting phase, we propose a discrepant reward for reinforcement learning (RL), which stably estimates a bias of the expected reward for each individual video.
- We assess our method on the MSR-VTT [1] and MSVD [27] datasets. Extensive evaluation results indicate that our method is competitive with the state-of-the-art systems while requiring shorter training schedules. Compared to [1], [11], [22], our method demonstrates improved captioning performance. We also use ablation studies to verify the power of our idea and the efficacy of our algorithm.

II. Related Work

In this section, we review representative work in video captioning as well as the technique of global-local representation and training strategies for the video caption task.

A. Video Captioning

Early video captioning works mainly focus on using template-based models for sentence generation to [28]–[30]. Inspired by the success of other vision tasks, the first work in [31] successfully extends the encoder-decoder architecture to develop a solution for the video captioning task. Following the same architectural paradigm, [31], [32] explore the temporal patterns on video using attention mechanisms to depict object movements. [33] develops a hierarchical attention module to apply content attention on each feature to select time intervals related to the semantic cues of the target word, and applies cross-modal syntax attention to model the feature importance of the target word under the guidance of syntax cues. [34] devises a MARN method, which generalizes descriptions from a single video to other videos with high semantic similarity. [8], [10], [35] develop an idea of feature fusion to guide sentence generation for video contents. [36] and [37] develop visual captioning models with semantic concepts. [38] and [39] attempt to generate diverse sentences for each video. [40], [41] and [42] explore video representation via visual reasoning. [43], [44] aim to develop boundary-aware sequence-to-sequence decoder for captioning. [45], [46] try to investigate the influence of the attention based on temporal components and semantics. [47] attempts to exploit multi-level semantic guidance via visual relation of objects. However, the above solutions capture global-local vision representation for sentence generation because they either neglect explicitly modeling the temporal content correspondences [12], [13] or the spatial object motion or tendency [11]. We instead explore the representations of both spatial interaction and temporal content features jointly using the proposed global-local encoder.

B. Global-Local Representation

To model the global-local vision representation, many existing methods [7], [23], [31], [34], [48], [49] resort to the sequence learning strategy. [49] uses a temporal attention method to depict the global-local connections. [7] leverages the decoding hidden states to increase the temporal feature representation. More recently, [11], [50], [51] exploit the object features to model the object movement across frames. For instance, [11] employs a bidirectional temporal graph to capture detailed movements for the salient objects in the video; [50] devises a stacked LSTM to encode both the frame-level and object-level temporal information. At the same time, [52] employs a stacked multimodal attention network to process additional visual and textual historical information as context features; [35] proposes four fusion filters to fuse different visual feature representations. However, the aforementioned methods primarily focus on salient objects from the global contents without explicitly modeling the global-local representation reasoning.

To address the above limitations, we propose a novel global-local representation granularity, which simultaneously exploits long-range temporal correspondence, short-range object motion, and local spatial appearances on the video frames. Using the accumulative global-local representation, our method can achieve fine-grained descriptions for video captioning. Besides, the generated video representations from our encoders can be directly used (may need finetuning) to transfer to any other video analysis tasks, including video grounding [53] or video retrieval [54].

C. Training Strategies

One popular strategy for training video captioning models is "Teacher Forcing" [55], which has been widely used in training video captioning tasks [12], [56]–[58]. Despite its popularity, the "Teacher Forcing" supervision is empirically suboptimal [25], [59]. More recently, many research efforts attempt to explore different training methods to boost the

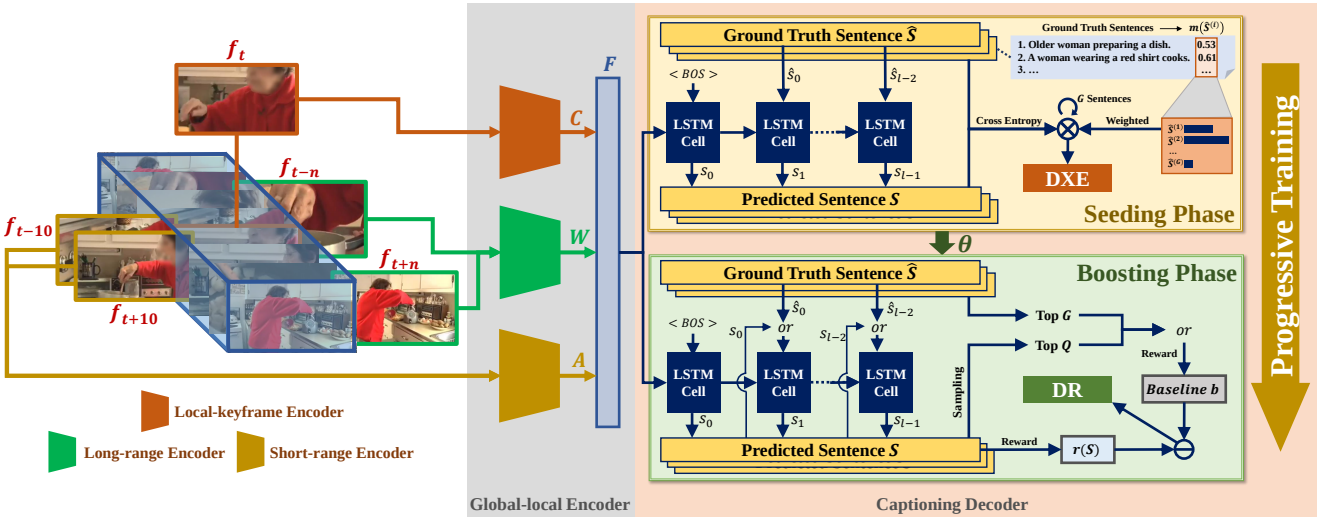


Fig. 2. The architectural framework of GLR. Our global-local representation encoder includes 1. the long-range encoder captures temporal correspondence among distant frames (t-n to t+n frames) and makes the cross-frame representations robust to appearance variations and shape deformations; 2. the short-range encoder focuses on motion and tendency, which depicts local consistency of object movement within a short moment (t±10 frames); 3. the local-keyframe encoder focuses on each object, which can preserve better object spatial information and finer details in terms of object appearances. In training, our method is trained by a progressive strategy which includes a seeding phase and then a boosting phase. The seeding phase supervises our method to obtain an entrance model which can be easily trained in the second boosting phase.

captioning performance [7], [10], [60]–[65]. For instance, [62] uses a mixed loss function to optimize the video captioning algorithm, which leverages the weighted combination of cross-entropy and reinforcement learning. Similarly, [60] adopts the paradigm of reinforcement learning and devises a self-critical baseline to reward the model learning to train the video captioning network. Although obtaining improvements over the prior methods [1], [49], [62], the above methods generally require a complicated pipeline to train with a heavy computation overhead for optimization. More recently, some works attempt to design a loss function to capture the location of temporal patterns [66] or spatial objects [14], but all of them fail to reconcile the demands of generating standard sentences and generating human-like sentences. Building on the lessons learned from the concurrent approaches [60]–[62], we propose a progressive training strategy, which can easily operate training on our proposed GLR. Empirical results indicate that the progressive training strategy can help us to achieve further performance than using the conventional optimization scheme.

III. Global-Local Representation Granularity

A. Overview

The framework of our GLR is demonstrated in Figure 2. Following [7], [23], our GLR adopts an encoder-decoder architecture. More specifically, we include a global-local encoder and a captioning decoder. The global-local encoder takes the long-range frames, the short-range frames, and the local keyframe as inputs and encodes them into different vocabulary features. All the obtained features are aggregated together to enrich global-local vision representations across video frames. Afterwards, the captioning decoder supervised by the progressive training strategy translates the vocabulary features into natural language sentences. We elaborate on our method in the following sections.

B. Global-Local Encoder

Our global-local encoder includes three essential parts: long-range encoder, short-range encoder, and keyframe encoder (Figure 2). Collectively, our novel encoder can help our method enrich global-local vision representations for video captioning tasks

1) Long-Range Encoder: Inspired by the random crop paradigm [67], we encode random global video frames to produce the global vocabulary based on a random keyframe f_t (see Figure 3) in training. Since each iteration will randomly choose different frames (the total number is fixed) from the videos, our training iteration will fully saturate the whole video clips. We encode random global video frames and produce the global vocabulary of the video content (Figure 3). Particularly, our long-range encoder first performs 2D convolutions on the inputs ($i.e.$, f_{t-n} and f_{t+n}[1]) to identify the relevant contextual features. The output features from the first step is processed by a 3D convolutional network (CNN) to capture global temporal correspondence. In order to increase consensus, we choose the top K word choices (highest frequency) from the ground truth sentences to guide the vocabulary generation as a K classifications task. Therefore, outputs of the dense layers are defined as:

$$W = \{w_1, w_2, ..., w_k, ..., w_K\}, w_k \in (0, 1), \quad (1)$$

[1]where n is a random range larger than 25 frames and t indicates the current keyframe.

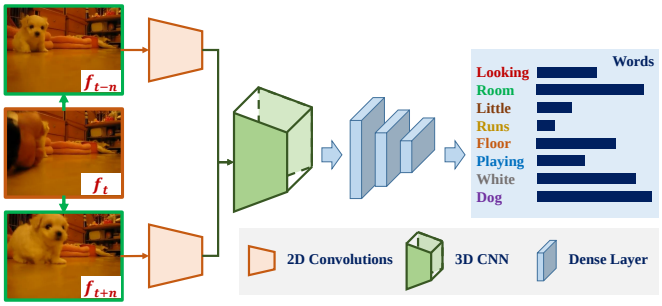


Fig. 3. The architecture of the long-range encoder. This encoder predicts the probability of whether a notional word will appear in the caption of the whole video.

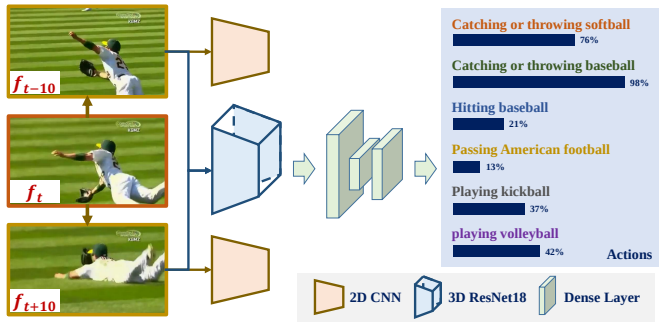


Fig. 4. The architecture of the short-range encoder. This encoder predicts the probability of short-term actions in the video.

where W is the collection of the predicted long-range vocabulary (including verbs, nouns and adjectives that may be used in sentences to describe the temporal contents in video) and w_k is the confidence of k_{th} word appearing in the predicted captioning for this video. This vocabulary is extracted from all annotated GT sentences of all videos in the video captioning dataset such as MSR-VTT [1] and MSVD [27], excluding function words such as "is", "be", "do", etc. Almost all previous methods lack attention to the adjectives ($i.e.$, "little" and "white") of the object or scene in the whole video, while this long-range adjective is an essential part of video description. Thus we include them in our long-range vocabulary.

2) Short-Range Encoder: Our short-range encoder is to capture object motion and tendency. We craft a 3D-Resnet18 [68] and two 2D CNNs into our architecture in parallel (Figure 4). By simultaneously taking two close neighbours (a.k.a. f_{t-10} and f_{t+10}) of the keyframe, 2D CNN and 3D-Resnet18 [68] yield the semantic and movement representations respectively. Afterward, these representations are stacked and fed into dense layers for action classifications. Given the number of action set in the dataset is J, our output of the short-range encoder is:

$$A = \{a_1, a_2, ..., a_j, ..., a_J\}, a_j \in (0, 1), \tag{2}$$

where A is the collection of the predicted short-range action vocabulary and a_j is the confidence of the j_{th} verb appearing in the predicted captioning for this video.

3) Local-Keyframe Encoder: The lexical knowledge for the local semantics is learned by a residual network [69], which extracts salient object features from the keyframe f_t. Given the number of image classes in the dataset is M, the output of our local encoder is:

$$C = \{c_1, c_2, ..., c_m, ..., c_M\}, c_m \in (0, 1), \tag{3}$$

where C is the collection of the object class vocabulary for the local frames and c_m is the confidence of the m_{th} class appearing in the predicted captioning for this video frame.

Once having all the vocabulary features from different ranges, we perform a fusion encoding. We first use a feature pool composed of linear layers φ to project each vocabulary feature into a same-size embedding and then aggregate them together to produce the fused feature F:

$$F = \text{Concat}(\varphi(W), \varphi(A), \varphi(C)) \tag{4}$$

C. Captioning Decoder

Our captioning decoder translates the fused feature into a l-word[2] sequence $S = (s_1, s_2, ..., s_y | y \in \{1, ..., l\})$ to form the predicted sentence. Specifically, we use language LSTM to generate the hidden state h_t and a cell state c_t at i_{th} step based on the fused feature F:

$$h_i, c_i = \mathbb{LSTM}([h_{i-1}, \Phi(s_{i-1}, \hat{s}_{i-1}, F)], c_{i-1}), \tag{5}$$

where $[\cdot, \cdot]$ denotes concatenation. $h_{i-1}, s_{i-1}, \hat{s}_{i-1}, F$, and c_{i-1} are the previous hidden state, the predicted word, the ground truth, the fused feature from encoding, and the cell state respectively. $\Phi(\cdot)$ is the annealing scheme which uses every previous token to predict the next word. We adopt a schedule sampling technique to randomly choose the token s_{i-1} or $\hat{s}_{i-1}$ by using a random variable $\xi \in \{0, 1\}$:

$$\Phi(s_{i-1}, \hat{s}_{i-1}, F) = \begin{cases} F, & (i = 1); \\ s_{i-1}, & (i > 1, \xi = 0); \\ \hat{s}_{i-1}, & (i > 1, \xi = 1), \end{cases} \tag{6}$$

when $i = 0$, the initial input of the LSTM is the fused feature F; when $i > 1$, we increase the probability of $\xi = 1$ gradually in every epoch until ξ is absolutely equal to 1. Then, we counter the process by decreasing the probability of $\xi = 1$.

As a widely used variant of the recurrent neural network (RNN), LSTM plays a crucial role in our captioning module to translate the encoder output into a sentence. It can output word by word according to input, and use long-term and short-term memory to learn grammar. A softmax function is used to perform sampling distribution for the next word. A trick called beam search [70] is used to find the sentences with the highest probability. Instead of choosing one word with the highest probability at t_{th} step, beam search records the top B words from the generated $B \times D$ (if $t \neq 0$) or D (if $t = 0$) words, where D is the vocabulary size. Therefore, it is more likely to find the global optimal solution than a greedy search (a special case of $B = 1$).

[2] l ($l = 30$) denotes the maximum length of a sentence.

Accordingly, the probability of a predicted word can be defined as:

$$p_\theta(s_i|h_i) = \text{softmax}(W_o \cdot h_i), \tag{7}$$

where h_i is the hidden state from Eq. (5) and W_o is the weight matrix [70] which maps the hidden state h_i to vocabulary-sized embedding, in order to find a context-matching word in the sentence. Thus, during decoding it defines a distribution over the output sequence $S = (s_1, s_2, ..., s_y)$ given the input sequence F as $p_\theta(S|F)$ is:

$$p_\theta(s_1, s_2, ..., s_y|F) = \prod_{i=1}^{y} p_\theta(s_i|h_i) \tag{8}$$

where $p_\theta(s_i|h_i)$ is defined in Eq. (7). So the encoding phase is to minimize this distribution according to the current parameter θ and the input feature F.

D. Training Objective for Global-Local Encoder

1) Training Objective for Long-Range Encoder: Supposing there are N_w videos in the video captioning dataset, the top K words (highest frequency) are chosen from the ground truth sentences. We consider the ground truth prediction for the i_{th} video is defined as $\hat{W}_i = \{\hat{w}_{i0}, \hat{w}_{i1}, ..., \hat{w}_{ik}, ..., \hat{w}_{i(K-1)}\} \in \{0,1\}^K$, where $\hat{w}_{ik} = 1$ means the k_{th} word exists in the ground-truth descriptions of this video, while $\hat{w}_{ik} = 0$ represents it does not. The long-range encoder computes the probability distribution of prediction $W = \{w_1, w_2, ..., w_k, ..., w_K\}, w_k \in (0,1)$. The long-range encoder network parameters are optimized by a multi-label cross-entropy loss function L_{long}:

$$\begin{aligned} L_{long} = \frac{1}{N_w}\frac{1}{K-1}\sum_{i=0}^{N_w}\sum_{k=0}^{K-1}[\hat{w}_{ik}\log w_{ik} \\ +(1-\hat{w}_{ik})\log(1-w_{ik})]. \end{aligned} \tag{9}$$

2) Training Objective for Short-Range Encoder: Let N_a denote the number of videos in the video action recognition dataset which has J actions in total. We consider the ground truth prediction for i_{th} video is defined as $\hat{A}_i = \{\hat{a}_{i0}, \hat{a}_{i1}, ..., \hat{a}_{ij}, ..., \hat{a}_{i(J-1)}\} \in \{0,1\}^J$, where $\hat{a}_{ij} = 1$ means j_{th} action can be used to describe this video, and $\hat{a}_{ij} = 0$ represents it does not. The short-range encoder computes the probability distribution of prediction $A = \{a_1, a_2, ..., a_j, ..., a_J\}, a_j \in (0,1)$. The short-range encoder network parameters are optimized by a multi-class cross-entropy loss function L_{short}:

$$\begin{aligned} L_{short} = \frac{1}{N_a}\frac{1}{J-1}\sum_{i=0}^{N_a}\sum_{j=0}^{J-1}[\hat{a}_{ij}\log a_{ij} \\ +(1-\hat{a}_{ij})\log(1-a_{ij})]. \end{aligned} \tag{10}$$

3) Training Objective for Local-Keyframe Encoder: Let N_c denote the number of videos in the image classification dataset which has M classes in total. We consider the ground truth prediction for i_{th} image is defined as $\hat{C}_i = \{\hat{c}_{i0}, \hat{c}_{i1}, ..., \hat{c}_{im}, ..., \hat{c}_{i(M-1)}\} \in \{0,1\}^M$, where $\hat{c}_{im} = 1$ means m_{th} class can be used to describe this image, and $\hat{c}_{im} = 0$ represents it does not. The local-keyframe encoder computes the probability distribution of prediction $C = \{c_1, c_2, ..., c_m, ..., c_M\}, c_m \in (0,1)$. The local-keyframe encoder network parameters are optimized by a multi-class cross-entropy loss function L_{local}:

$$\begin{aligned} L_{local} = \frac{1}{N_c}\frac{1}{M-1}\sum_{i=0}^{N_c}\sum_{m=0}^{M-1}[\hat{c}_{im}\log c_{im} \\ +(1-\hat{c}_{im})\log(1-c_{im})]. \end{aligned} \tag{11}$$

E. Training Objective for Decoder

In the first several time steps, the LSTM layer receives a sequence of features and there is no loss during this stage. After all the features for the video clip are exhausted, the LSTM layer is fed the beginning-of-sentence ($< BOS >$) tag, which prompts it to start decoding its current hidden representation into a sequence of words. Zeros are used as a $< PAD >$ tag when there is no input for the LSTM at this time step. While training in the decoding stage, the model maximizes for the log-likelihood of the predicted output sentence given the hidden representation and the previous words it has seen. From Eq. (8) for a model with parameters θ and output sequence $S = (s_1, s_2, ..., s_y)$, this is formulated as:

$$\theta^* = \arg\max_\theta \sum_{i=1}^{y} \log p_\theta(s_i|h_i) \tag{12}$$

This log-likelihood is optimized over the entire training split using stochastic gradient descent. The loss is computed only during the training stage.

F. Progressive Training

We propose a progressive training strategy to fulfill different learning objectives (See Algorithm 1). Our progressive training includes a seeding phase followed by a boosting phase (Figure 2). In the seeding phase, our learning is optimized by cross-entropy, which produces an entrance model to facilitate smooth training in the second phase. In the boosting phase, our training leverages reinforcement learning (RL) to achieve further performance gain.

1) Seeding Phase: The conventional models [1], [7], [11], [22], [71] are commonly trained with the cross-entropy (XE) loss, which measures the average similarity of the generated sentence and all the ground truth sentences. Since different annotators may interpret video content differently, the ground truth from the training dataset may include annotation bias. We argue that direct comparison between the captioning predictions to the ground truths may not yield the optimal training outcomes. Thus, we employ the metric scores $m(\hat{S})$ of all ground truths as a discriminative weight in computing cross-entropy to make our training biased towards those well-written ground truth sentences. Intuitively, manually annotated ground truths have severe bias, that is, some ground truth sentences are well-writen, while others are ambiguous or inappropriate. Metric scores encourage the training to focus on the well-written sentences. The $m(\hat{S})$ can use different options, such as BLEU_4 [72], METEOR [73], ROUGE_L [74], and CIDEr

Algorithm 1 Progressive Training (PT)

Input: Extracted feature list $\{F_1, F_2, ..., F_{N_w}\}$ for all N_w videos, captioning decoder parameters θ, and all annotation list $\{\hat{\mathbb{S}}_1, \hat{\mathbb{S}}_2, ..., \hat{\mathbb{S}}_{N_w}\}$ for all videos where $\hat{\mathbb{S}} = \{\hat{S}^{(1)}, \hat{S}^{(2)}, ..., \hat{S}^{(G)}\}$ is G human-labeled captions for each video.

Output: Learned decoder parameters θ^*

```
Initialize θ
# Seeding Phase
repeat
    for F in {F_1, F_2, ..., F_{N_w}} do
        Update θ* ⇐ arg min_θ L_DXE(θ) using Eq. 13
    end for
until reach the max epochs of seeding phase.
# Boosting Phase
repeat
    for F in {F_1, F_2, ..., F_{N_w}} do
        Update θ* ⇐ arg min_θ L_DR(θ) using Eq. 19
    end for
until the solution converges.
return Learned parameters θ*
```

[75][3]. The analysis of each option will be reported in the experiments.

Providing each video is annotated by G sentences $\hat{\mathbb{S}} = \{\hat{S}^{(1)}, \hat{S}^{(2)}, ..., \hat{S}^{(G)}\}$, the *discriminative cross-entropy (DXE)* loss function is:

$$L_{DXE}(\theta) = -\frac{1}{G}\sum_{i=1}^{G} m(\hat{S}^{(i)}) \log\, p(\hat{S}^{(i)}|F;\theta). \tag{13}$$

Our DXE loss increases the probability of generating captions with a high metric score by assigning higher weights to ground-truth sentences. The gradient of DXE is calculated by the weighted difference between the prediction and all the target descriptions. Consequently, the DXE loss encourages feature learning which increases the probability of generating captions with a high metric score. The result of $m()$ is considered a constant in our loss function, every GT sentence has a different computed value. Different from the weighted loss entropy (which manually assigns weights to classes to address the problem of unbalanced data), the weight $m(\hat{S})$ of our DXE is automatically calculated through metrics, evaluating the quality among all annotations. Our DXE assigns higher weights to high-quality annotations, helps the model generate sentences closer to them. For example, if CIDEr is selected as the metric $m(\hat{S}^{(i)})$, it assists the model to refer more to the sentence with high human consensus; then the model may be taught to generate more human-like captions. Empirical results resonate with our assumption.

2) Boosting Phase: After the seeding phase, we employ reinforcement learning [76] with a *discrepant reward (DR)* to further boost the performance of our GLR model. Rather than estimating a self-critical baseline [60] to fill the gap between training and testing, our DR harmonizes the model with respect to the distinctions of each video.

Because the video captioning model can be regarded as an agent which interacts with an environment of visual observation and natural language, optimizing it can be formulated as a reinforcement learning task. Considering the actor (LSTM cell) with parameters θ samples words and the trajectory is the generated sentence S, our *discrepant reward (DR)* loss is defined as the negative expected reward:

$$L_{RL}(\theta) = -\overline{R}_\theta = -\sum_S r(S)p(S|F;\theta), \tag{14}$$

where the reward $r(S)$ is the evaluation metric score of the sampled sentence and F is the feature generated by our global-local encoder. In order to optimize the actor, we use gradient descent to update the network parameters θ by computing the differential of the loss:

$$\nabla_\theta L_{RL}(\theta) = -\sum_S \nabla_\theta r(S)p(S|F;\theta). \tag{15}$$

Since the reward $r(S)$ is not a function that depends on θ, it is not differentiable with regard to θ, so the gradient in (15) can be rewritten as:

$$\begin{aligned}\nabla_\theta L_{RL}(\theta) &= -\sum_S r(S)\nabla_\theta p(S|F;\theta) \\ &= -\sum_S p(S|F;\theta)r(S)\frac{\nabla_\theta p(S|F;\theta)}{p(S|F;\theta)} \\ &= -\sum_S p(S|F;\theta)r(S)\nabla_\theta \log\, p(S|F;\theta) \\ &= -\mathbb{E}_{S\sim p}[r(S)\nabla_\theta \log\, p(S|F;\theta)].\end{aligned} \tag{16}$$

where $\mathbb{E}_{S\sim p}$ denotes the expected value of the distribution, the reward $r(S)$ is the evaluation metric score of the sampled sentence, and F is the fused feature extracted from our global-local encoder. One problem with this training strategy is that the reward function $r(S)$ is always positive because the metric score ranges between 0 and 1. Therefore, we can only encourage feature representations in learning but cannot perform suppression.

To address this issue, our DR is equal to the original reward $r(S)$ subtracts a bias b, which is *baseline*. With the bias term, our learning can be more robust to variation in prediction. Then the policy gradient is given by:

$$\nabla_\theta L_{DR}(\theta) = -\mathbb{E}_{S\sim p}[(r(S) - b)\nabla_\theta \log p(S|F;\theta)], \tag{17}$$

where $b \approx E[r(S)]$. Our *baseline* b can be any arbitrary function, as long as it does not depend on the S so does not change the expected value of gradient:

$$\begin{aligned}\sum_S b\nabla_\theta p(S|F;\theta) &= b\nabla_\theta \sum_S p(S|F;\theta) \\ &= b\nabla_\theta 1 = 0.\end{aligned} \tag{18}$$

The self-critical method SCST [60] utilizes the reward of the greedy output at the test time as the baseline b, harmonizing the model with respect to its test-time inference procedure. But it incurs the time cost to run inference again in every training iteration. Using one greedy sample to estimate the expected

[3]Denotes as `B@4`, `M`, `R`, and `C` respectively in experiments.

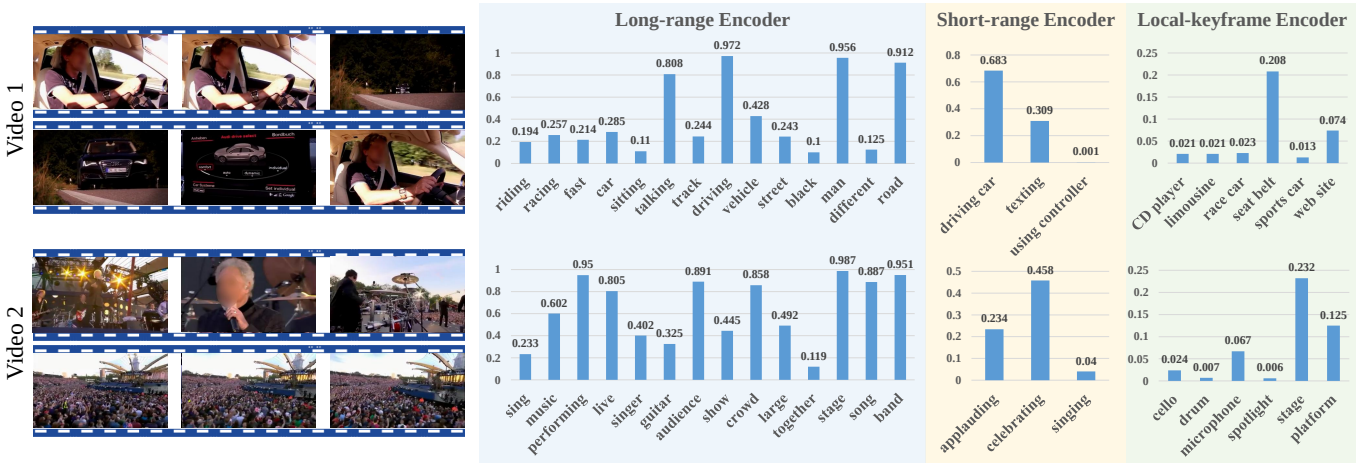


Fig. 5. Probability distributions of outputs from our long-range, short-range and local encoder. We only show top-14 results of our long-range encoder output, top-3 results of our short-range encoder output and top-6 results of our local-keyframe encoder output respectively. Those results show our global-local encoder can perceive the information of different aspects at various levels of video.

reward is too noisy, resulting in larger gradient variance. In our implementation, the baseline b has two variants: 1) b_1 obtained by the G ground-truth captions; and 2) b_2 the top Q sentences sampled by the model with the highest score during the forward step. Compared to the self-critical baseline, our discrepant baseline with sufficiently large G or Q establishes a relatively stable bias for each different input video, helps our method have a more robust and efficient estimation of expected reward. When updating, this gradient ∇_θ can be approximated by Monte-Carlo sampling through a single training example. So the final gradient of our discrepant reward is defined as:

$$\begin{aligned}\nabla_\theta L_{DR}(\theta) &= -\mathbb{E}_{S\sim p}[(r(S)-r(S^{b_j}))\nabla_\theta \log p(S|F;\theta)],\\ &\approx -(r(S)-r(S^{b_j}))\nabla_\theta \log p(S|F;\theta),\end{aligned} \tag{19}$$

where S^{b_j} can be used by the either baseline (b_1 or b_2). In our experiments, we carry out an ablation study to discover the impact of b_1 and b_2 on the captioning performance.

IV. Experiments

A. Dataset and Evaluation Metrics

1) MSR-VTT: The performance of our GLR approach is evaluated on the challenging MSR-VTT dataset [1], which consists of 1000 videos. Each video is associated with 20 ground-truth captions given by different workers. We follow the data split in the original publication, allocating 6513 videos for training, 497 videos for validation, and 2990 videos for testing.

2) MSVD: We also evaluate our GLR on the MSVD dataset [27], which consists of 1,970 Youtube video clips with 85K English descriptions. Following the previous works [31], [32], [77] , we split the dataset into a 1,200 training set, 100 validation set, and 670 testing set by the contiguous index number.

3) Evaluation Metrics: We evaluate our method on four commonly used metrics BLEU_4 [72], METEOR [73], ROUGE_L [74], and CIDEr [75], which are denoted as `B@4`, `M`, `R`, and `C` respectively. `B@4` measures the precision of 4-grams between the ground-truth and generated sentences. `M` uses a uni-gramsbased weighted F-score and a penalty function to penalize incorrect word order. `R` computes a harmonic mean of precision and recall between compared sentences on the longest common subsequence (LCS). `C` is a voting-based approach, which measures the consensus among sentences, is robust to incorrect annotations.

TABLE I
PERFORMANCE OF OUR GLR ON MSR-VTT VALIDATION SPLIT DURING PROGRESSIVE TRAINING. THE BOOSTING PHASE BEGINS FROM THE BEST MODEL (*e.g.*, WITH HIGHEST CIDER) BEFORE THE 30_{th} EPOCH. IN THIS TABLE THE BEST MODELS ARE ACQUIRED AT THE 28_{th} EPOCH, SO THE BOOSTING PHASE STARTS FROM THE 29_{th} EPOCH. THE ↑ AND ↓ ARROWS INDICATE THE INCREASE OR DECREASE FROM THE 28_{th} EPOCH.

	Epoch	B@4	M	R	C
Seeding	1	31.0	23.3	55.5	21.9
	5	42.1	28.1	61.1	45.8
	10	44.8	29.3	62.5	51.9
	20	46.7	30.7	63.8	53.2
	28	46.9	31.1	64.0	57.4
	30	47.0	31.0	64.4	56.3
Boosting	29	$45.0\downarrow_{1.9}$	$29.9\downarrow_{1.2}$	$63.6\downarrow_{0.4}$	$52.8\downarrow_{4.6}$
	39	$45.5\downarrow_{1.4}$	$30.3\downarrow_{0.8}$	$64.1\uparrow_{0.1}$	$60.0\uparrow_{2.6}$
	59	$45.3\downarrow_{1.6}$	$30.8\downarrow_{0.3}$	$64.5\uparrow_{0.5}$	$61.9\uparrow_{4.5}$
	79	$45.1\downarrow_{1.8}$	$31.0\downarrow_{0.1}$	$64.8\uparrow_{0.8}$	$63.6\uparrow_{6.2}$
	99	$46.9\downarrow_{0.0}$	$31.2\uparrow_{0.1}$	$65.7\uparrow_{1.7}$	$64.6\uparrow_{7.2}$

B. Implementation Details

1) Long-Range Encoder: Our long-range encoder is pre-trained on our proposed video-to-word dataset generated from MSR-VTT [1] dataset or MSVD [27] dataset. For our 2D CNN, We adopt ResNeXt [69] and use the 2048-dimension average pooling features from the `conv5_3` output as the 2D representation of videos. Then, 3D CNN uses ECO [78] followed by a global pool, which outputs 1536-dim features. Afterward, we use 3 layers of dense connection to predict K vocabulary feature classification, where we set $K = 300$,

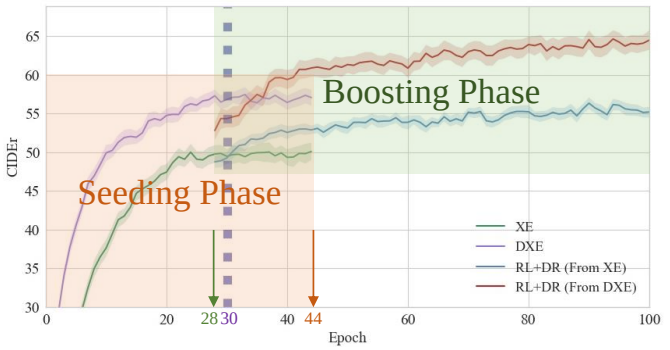


Fig. 6. Smoothed CIDEr score curve on MSR-VTT validation split during progressive training. We set the max epoch number of the seeding phase to be 30 in progressive training. But in order to prove the effectiveness of the boosting phase, we have run the seeding phase more than 30 epochs ($e.g.$, 44 epochs) to show that the CIDEr score is lower than the boosting phase after 30 epochs. The boosting phase begins from the best model ($e.g.$, with highest CIDEr) before the 30_{th} epoch. In this figure the best models are acquired at the 28_{th} epoch.

namely, 300 candidate words chosen from the ground truth of the MSR-VTT [1] or MSVD [27]. And we allocate 7010 videos for training (6513 original training + 497 original validation) and 2990 videos for testing for MSR-VTT [1], while 1300 videos for training (1200 original training + 100 original validation) and 670 videos for testing for [27]. The dimension of the dense layer is set to 512. We set the learning rate to 0.0002, batch size to 64, and use Adam to optimize the network parameters. The dropout rate is set to 0.5 during training. The demo results are shown in Figure 5. We select 20 keyframes evenly in the time sequence for each video. Our long-range encoder outputs the probability distribution of all candidate words, including fine-grained nouns like "audience" (a subcategory of "people" in special context) and adjectives such as "fast", "black" and "large".

2) Short-Range Encoder: The short-range encoder is pre-trained on Kinetics-400 dataset [79]. The 2D CNN in the short-range encoder adopts the first part of the BN-Inception architecture (until inception-3c layer) [80]. The outputs from the 2D CNN have feature size of 28×28 with 96 dimensions. Following that, our 3D-Resnet18 [68] uses 3 conv layers (with 3×3×3 kernel and dimensions of 128, 256 and 512 respectively), which output a one-hot vector for the J action class labels, where $J = 400$, namely, the confidence distribution of 400 actions. The initial learning rate for the short-range encoder is set to 0.001 and is decreased by a factor of 10 when validation error saturates for 4 epochs. We train the network with a momentum of 0.9, a weight decay of 0.0005, and mini-batches of size 32. We initialize the weights of the 2D-Net weights with the BN-Inception architecture [80] pre-trained on Kinetics, as provided by [81]. In the same way, we use the pre-trained model of 3D-Resnet-18 [68], as provided by [82] for initializing the weights of our 3D-Net. Afterwards, we train our whole short-range encoder on the Kinetics-400 dataset [79] for 10 epochs. We select 20 keyframes evenly in the time sequence for each video. The demo results are shown in Figure 5, our short-range encoder outputs the probability distribution of all actions appearing in the short-term split of the video, such as "applauding".

TABLE II
THE FINAL CIDER SCORES OF USING DIFFERENT FRAME INTERVALS IN LONG-RANGE AND SHORT-RANGE ENCODER.

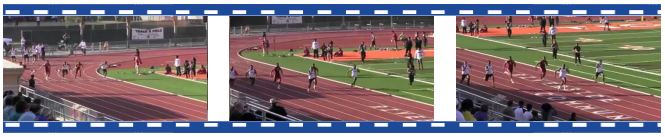
Long \ Short	4	8	10	12	16
$n > 20$	56.4	58.0	58.9	56.8	55.9
$n > 25$	57.2	59.1	60.6	58.7	57.5
$n > 30$	56.8	58.6	60.3	57.4	56.1

TABLE III
COMPARISON OF USING DIFFERENT FEATURES (LOCAL, SHORT, AND LONG-RANGE FEATURES) IN THE SEEDING AND BOOSTING PHASE. DXE LOSS IS USED. ↑ INDICATES THE INCREASE FROM THE PRIMITIVE METHOD WHICH ONLY USES THE LOCAL FEATURE. THE SETTINGS OF LONG-RANGE, SHORT-RANGE AND LOCAL-KEYFRAME ENCODERS ARE THE SAME WITH SECTION IV-B.

	Features			B@4	M	R	C
	Local	Short	Long				
Seeding Phase	Single feature						
	✓	×	×	31.7	24.0	54.5	35.3
	×	✓	×	32.3	23.9	54.1	34.8
	×	×	✓	43.8	28.7	61.2	51.7
	Combined features						
	✓	✓	×	36.7↑5.0	26.0↑2.0	57.9↑3.4	42.3↑7.0
	✓	×	✓	45.1↑13.4	29.3↑5.3	62.0↑7.5	53.0↑17.7
	×	✓	✓	45.6↑13.9	29.3↑5.3	62.9↑8.4	53.9↑18.6
	✓	✓	✓	46.9↑15.2	30.4↑6.4	63.9↑9.4	55.0↑19.7
Boosting Phase	Single feature						
	✓	×	×	34.5↑2.8	26.1↑2.1	58.1↑3.6	43.8↑8.5
	×	✓	×	34.4↑2.1	25.8↑1.9	57.9↑3.8	41.0↑6.2
	×	×	✓	45.8↑2.0	30.4↑1.3	64.4↑3.2	57.9↑6.2
	Combined features						
	✓	✓	×	38.2	27.1	60.1	50.5
	✓	×	✓	46.1	30.6	64.8	59.5
	×	✓	✓	46.2	30.6	64.7	59.6
	✓	✓	✓	46.9	31.2	65.7	60.6

3) Local-Keyframe Encoder: We employ ResNeXt-101 [69] with 64 paths in each block, which generates the probabilities of M objects ($M = 1000$) for each input video frame. Following standard practice [22], the local-keyframe encoder is pre-trained on ImageNet [83] and we train the ResNeXt-101 [69] on ImageNet. On the ImageNet dataset, the input image is 224×224 randomly cropped from a resized image using the scale and aspect ratio augmentation of [84]. We use SGD with a mini-batch size of 256. The weight decay is 0.0001 and the momentum is 0.9. We start from a learning rate of 0.1, and divide it by 10 for three times. We adopt the weight initialization of [85]. We select 30 keyframes evenly in the time sequence for each video. The demo results are shown in Figure 5, our local-keyframe encoder outputs the probability distribution of all objects categories, such as "seat belt", "microphone" and "platform", which provides more finer objects for our captioning decoder.

4) Feature Fusion: The inputs for feature fusion are 300-dim from the long-range encoder, 400-dim from the short-range encoder, and 1000-dim from the local-keyframe encoder respectively. The dense layers in the feature pool convert each of them into a 512-dim embedding. We use the ReLU



Local: A **man** is **playing** a **football game**.
Local + Short: A **man** is **running** on a **track**.
Local + Short + Long: A **group of people** are **running** on a **race track**.

Local: A **woman** is **cooking food**.
Local + Short: A **woman** is **cutting** a **potato**.
Local + Short + Long: A **man** is **slicing** a **potato**.

accurate
inaccurate
do not care

Fig. 7. Qualitative results of adding features short and long-range feature. Incremental training is used.

GLR (XE): There is a **woman** is **walking** down the **runway**.
GLR (DXE): **Models** are **walking** down the **runway**.
GLR (RL+DR): **Models** are **walking** down a **runway** in a **fashion show**.

GLR (XE): A **man** is **talking** about a **video game**.
GLR (DXE): A **man** is **talking** about a **toy**.
GLR (RL+DR): A **man** is **talking** about a **SpongeBob**.

accurate
inaccurate
do not care

Fig. 8. Qualitative results of progressive training compared with using only XE training.

TABLE IV
COMPARISON OF USING DIFFERENT WEIGHTING METRIC $m(\hat{S}^{(i)})$ FOR DXE IN THE SEEDING PHASE. ↑ AND ↓ INDICATES THE INCREASE OR DECREASE RESPECTIVELY FROM THE METHODS TRAINED BY XE.

	$m(\hat{S}^{(i)})$	B@4	M	R	C
XE	-	45.5	30.1	62.6	51.2
DXE	B@4	45.5$\uparrow_{0}$	29.7$\downarrow_{0.4}$	63.0$\uparrow_{0.4}$	51.4$\uparrow_{0.2}$
	M	44.5$\downarrow_{1.0}$	29.8$\downarrow_{0.3}$	62.9$\uparrow_{0.3}$	52.4$\uparrow_{1.2}$
	R	45.2$\downarrow_{0.3}$	29.0$\downarrow_{1.1}$	63.5$\uparrow_{0.9}$	52.5$\uparrow_{1.3}$
	C	46.9$\uparrow_{1.4}$	30.4$\uparrow_{0.3}$	63.9$\uparrow_{1.3}$	55.0$\uparrow_{3.8}$

activation of each dense layer in the feature pool and adopt a dropout of 0.5 to prevent overfitting. Finally, the output of the feature fusion (by concatenation of 512×3) has a dimension of 1536.

5) Decoder Setting: During the decoding stage, the initial step takes the fused features as inputs (as in Eq. (6)) to predict the first word. Afterward, the previously predicted word is embedded into a 512-dimensional vector then fed into the LSTM cell with a 512-dimensional hidden state to produce the next token.

TABLE V
COMPARISON OF USING XE OR DXE-TRAINED ENTRANCE MODEL IN THE BOOSTING PHASE. ↑ INDICATES THE INCREASE FROM THE SEEDING PHASE. IN BOOSTING PHASE, BASELINE b_2 IS USED IN OUR REWARD.

	Start From	R	C
Seeding Phase	-	62.6	51.2
Boosting Phase	XE	63.3$\uparrow_{0.7}$	55.3$\uparrow_{4.1}$
	DXE	65.7$\uparrow_{3.1}$	60.6$\uparrow_{9.4}$

TABLE VI
THE PERFORMANCE OF USING DIFFERENT BASELINES b_1 OR b_2 AS DISCREPANT REWARD (AS IN EQ. (17)). ↑ INDICATES THE INCREASE FROM "-" (WITHOUT BASELINE). WE USE DXE IN THE SEEDING PHASE.

b	B@4	M	R	C
-	13.5	16.1	46.1	12.7
b_{scst}	44.6$\uparrow_{31.1}$	30.2$\uparrow_{14.1}$	64.3$\uparrow_{18.2}$	56.4$\uparrow_{43.7}$
b_1	46.4$\uparrow_{32.9}$	30.5$\uparrow_{14.4}$	65.0$\uparrow_{18.9}$	58.1$\uparrow_{45.4}$
b_2	46.9$\uparrow_{33.4}$	31.2$\uparrow_{15.1}$	65.7$\uparrow_{19.6}$	60.6$\uparrow_{47.9}$

TABLE VII
COMPARISON OF USING RNN, GRU AND LSTM AS CAPTIONING DECODER. ALL THE SETTINGS INCLUDING THE EMBEDDING SIZE AND THE HIDDEN STATE OF THEM ARE THE SAME.

Method	B@4	M	R	C
RNN	41.5	28.3	61.7	52.1
GRU	43.6$\uparrow_{2.1}$	29.3$\uparrow_{1.0}$	63.5$\uparrow_{1.8}$	55.4$\uparrow_{3.3}$
LSTM	46.9$\uparrow_{5.4}$	31.2$\uparrow_{2.9}$	65.7$\uparrow_{4.0}$	60.6$\uparrow_{8.5}$

6) Seeding Phase Setting: We use the learning rate of 0.0003 in this phase with the Adam optimizer. When MSR-VTT is used, we operate training on all 20 ground-truth captions of each video at the same time. Accordingly, we compute the weighted metrics with ground-truth sentences ($G = 20$) for DXE. When MSVD is used, we operate training on all 17 ground-truth captions of each video at the same time. Accordingly, we compute the weighted metrics with ground-truth sentences ($G = 17$) for DXE. Beam search [70] is used to find the sentences with the highest probability, where the beam size B in search is set to be 5 with the max sequence length $l_{max} = 30$ for sentence inference.

7) Boosting Phase Setting: We use the learning rate of 0.0001 in this phase. Our model is also trained on all 20 ground-truth captions for MSR-VTT or 17 ground-truth captions for MSVD of each video at the same time. For baseline b_1, we compute the average reward of those 20 or 17 ground-truth sentences; for baseline b_2, we compute the average reward of $Q = 100$ sentences sampled from our trained model. Beam search [70] is also used in this phase. Recent work on video captioning [7], [32], [60], [61] has shown that CIDEr as a reward outperforms other evaluation metrics (*e.g.*, CIDEr, BLEU, or METEOR) to gain the largest improvement for video captioning. Following [7], [32], [61], we also use CIDEr score to compute the reward in training.

TABLE VIII
COMPARISONS WITH STATE-OF-THE-ART METHODS ON MSR-VTT BENCHMARK. THE BEST AND THE SECOND-BEST METHODS ARE HIGHLIGHTED. IN THE TRAINING COLUMN, "XE" IS CROSS-ENTROPY; "DXE" IS DISCRIMINATIVE CROSS-ENTROPY; "RL" IS REINFORCEMENT LEARNING; "DR" IS THE DISCREPANT REWARD. "EPOCH" INDICATES THE TRAINING SCHEDULE FOR EACH COMPARED METHOD. "FEATURE" CORRESPONDING TO DIFFERENT GLOBAL-LOCAL VISION REPRESENTATIONS ARE LISTED. "PT" IN OUR METHOD STANDS FOR PROGRESSIVE TRAINING, WHICH OPTIMIZES THE CIDER METRIC IN BOOSTING PHASE. METRICS FOR OTHER METHODS ARE OBTAINED USING THEIR OFFICIAL CODE AND TRAINED MODEL.

Training	Method	Epoch	Features			B@4	M	R	C
			Local	Short	Long				
XE	SA-LSTM [1]	100	✓	✓	×	40.5	29.9	-	-
	RecNet [86]	-	✓	×	×	39.1	26.6	59.3	42.7
	POS_{XE} [7]	-	✓	✓	×	42.0	28.1	61.1	49.0
	MARN [71]	500	✓	✓	×	40.4	28.1	60.7	47.1
	OA-BTG [11]	-	✓	×	×	41.4	28.2	-	46.9
	$SAAT_{XE}$ [87]	100	✓	✓	×	40.5	28.2	60.9	49.1
	ORG-TRL [22]	-	✓	✓	×	43.6	29.7	62.1	50.9
	STGraph [23]	50	✓	×	×	40.5	28.3	60.9	47.1
	Ours (GLR)	**30**	✓	✓	✓	**45.5**	**30.1**	**62.6**	**51.2**
DXE	**Ours (GLR)**	**30**	✓	✓	✓	**46.9**	**30.4**	**63.9**	**55.0**
RL	HRL [61]	-	✓	×	×	41.3	28.7	61.7	48.0
	PickNet [32]	300	✓	×	×	38.9	27.2	59.5	42.1
	POS_{RL} [7]	-	✓	✓	×	41.3	28.7	62.1	53.4
	VRE [77]	-	✓	×	×	43.2	28.0	62.0	48.3
	$SAAT_{RL}$ [87]	200	✓	✓	×	39.9	27.7	61.2	51.0
RL+DR	**Ours (GLR + PT)**	**100**	✓	✓	✓	**46.9**	**31.2**	**65.7**	**60.6**

TABLE IX
COMPARISONS WITH STATE-OF-THE-ART METHODS ON MSVD BENCHMARK. THE BEST AND THE SECOND-BEST METHODS ARE HIGHLIGHTED.

	Method	B@4	M	R	C
XE	RecNet [86]	52.3	34.1	69.8	80.3
	POS_{XE} [7]	52.5	34.1	71.3	88.7
	MARN [71]	48.6	35.1	71.9	92.2
	OA-BTG [11]	56.9	36.2	-	90.6
	ORG-TRL [22]	54.3	36.4	73.9	95.2
	STGraph [23]	52.2	36.9	73.9	93.0
DXE	**Ours (GLR)**	**57.7**	**38.6**	**74.9**	**95.9**
RL	PickNet [32]	46.1	33.1	69.2	76.0
	POS_{RL} [7]	53.9	34.9	72.1	91.0
	VRE [77]	51.7	34.3	71.9	86.7
RL+DR	**Ours (GLR + PT)**	**60.5**	**38.9**	**76.4**	**101.0**

C. Training Logs

We train the captioning decoder on an NVIDIA GeForce GTX 1080 Ti, while the parameters of the global-local encoder are kept frozen, the average iteration time in training for XE, DXE and RL+DR is 0.212, 0.226 and 1.75 respectively when batch size is 32. We check performance on the validation set for every epoch and report all the scores in Table I and the CIDEr scores in Figure 6. Table I shows all the scores are increasing whenever in the seeding phase or boosting phase, and our full model achieves improvements over the primitive model which only uses seeding phase by 0.1% on M, 1.7% on R, and 7.2% on C. Figure 6 shows that the CIDEr score increases fast in the seedind phase but is stable and lower than the boosting phase after 30 epochs, while using RL+DR can continue to train the model in the second phase. When switching from XE training to RL training, all the RL models have difficulty converging in the first few iterations, but quickly recover and reach higher performance levels.

D. Ablation Study

We conduct extensive ablation studies to discover the optimal settings related to our global-local encoding as well as the seeding and boosting phase in the training of our system.

1) Determine Global-Local Range: The effects of choosing different frame intervals for long-range and short-range encoder are illustrated in Table II. These results show that using $n > 25$ for the long-range encoder and choosing the f_{t-10}, f_{t+10} for the short-range encoder achieves the best performance.

2) Global-Local Features: We measure the performance of our model using different global-local features (as shown in Table III). We first evaluate the performance of different methods which use individual features for captioning prediction. Results indicate that using the long-range has the highest performance on all metrics in comparison with the other two methods. We also examine the impact of progressively combining different features together. Our full model using all three features outperforms the models which only use local features or short-term features by significant margins. For instance, our full model achieves improvements over the primitive model which only uses local features, by 15.2% on B@4, 6.4% on M, 9.4% on R, 19.7% on C in the seeding phase, and by 12.4% on B@4, 5.1% on M, 7.6% on R, 16.8% on C in the boosting phase.

As Figure 7 shows, adding short and long-range feature in the boosting phase can get more fine-grained captions than only using local-keyframe features, for example, "race track" is more fine-grained than "track" and "slicing" is more fine-grained than "cutting". In particular, after adding the long range encoder, the model is more robust to capturing the overall perception of video.

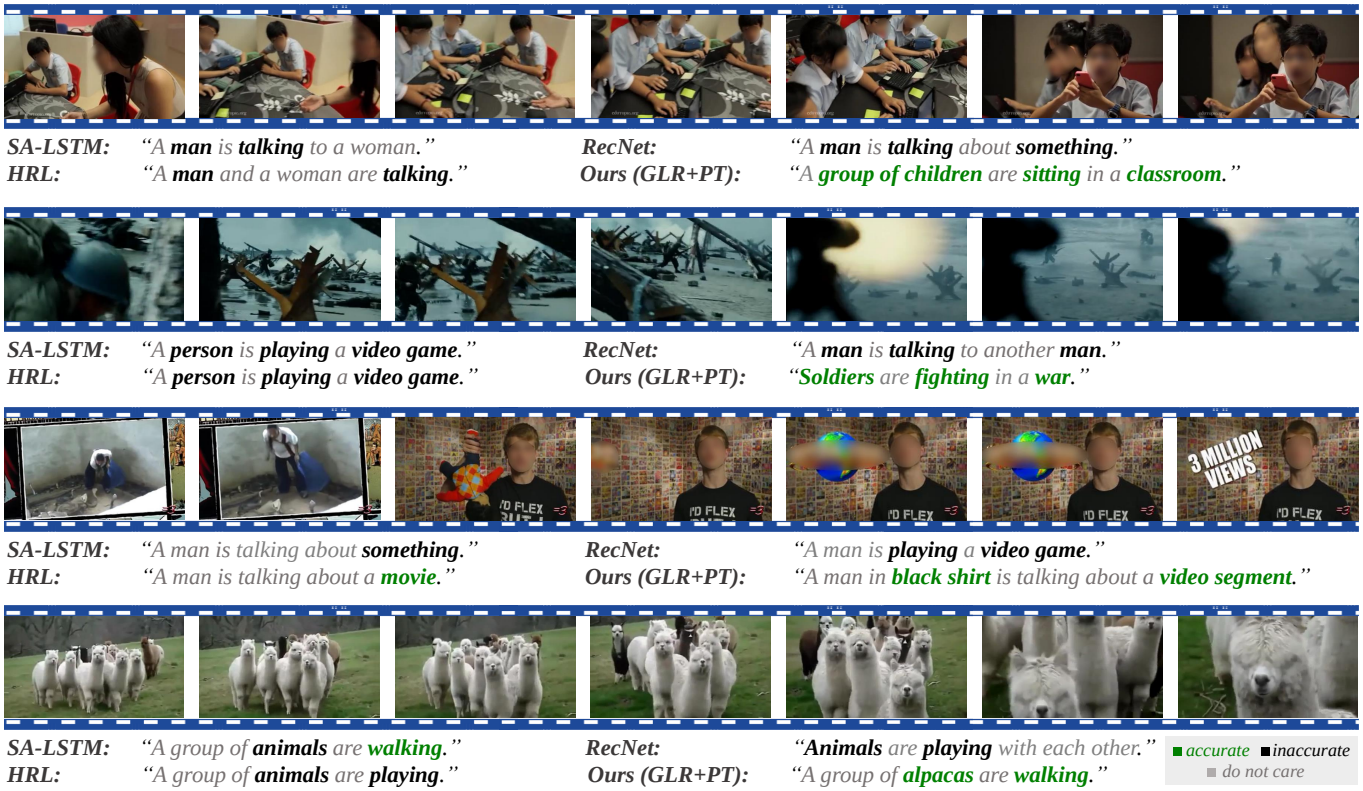


Fig. 9. Qualitative results on MSR-VTT [1]. We present comparisons with state-of-the-art methods SA-LSTM [1], RecNet [86] and HRL [61]. Based on the results, our method can generate more accurate captioning sentences to describe the context of the given video.

3) Different Weighted Metrics in Seeding Phase: The seeding phase training is important as it produces the entrance model for the following boosting phase. Hence, we evaluate the impact of using different weighting metric (a.k.a. `B@4`, `M`, `R`, and `C`) in training. Results are shown in Table IV. Models trained by different DXE loss all outperform the counterpart trained by XE. Meanwhile, using CIDEr as the metric weight in DXE training obtains the best results on all metrics. We use all global-local features (long-range, short-range, local features) here. As Figure 8 shows, in the seeding phase, using our DXE loss can get more correct words than the XE loss. For example, using the DXE loss can comprehend the "models" in the video rather than just output a "woman", and the "toy" is found by using DXE while it is mistaken for a "video game" by the XE method.

4) Progressive Training Analysis: We investigate if the progressive training could effectively improve our method in predicting captioning. As Table V shows, the results from the boosting phase increase steadily from the seeding phase. The boosting phase starting from the seeding phase and using DXE gets higher scores than its counterpart using XE. This demonstrates that using DXE as the supervision in the seeding phase can yield more optimal model parameters and further improve the performance of the boosting phase than using XE in training. The CIDEr score increasing curve of boosting phase from seeding phase (shown in Figure 6) also proves the performance improvements in our progressive training. Our boosting phase can improve all of the method no matter which single feature, or combined features, is used. As Figure 8 shows, after boosting phase training, our model can output more detailed and fine-tuned information (such as "fashion show" and "SpongeBob") when RL+DR is used.

5) Using b_1 vs. b_2 in the Boosting Phase: The results of our GLR using different discrepant (b_1 and b_2) rewards compared with self-critical baseline (b_{scst}) in [60] are shown in Table VI. Using the reward (b_2) based on top Q sentences sampled by the model can help supervise models to achieve a better performance than using the reward (b_1) based on the ground-truth sentences. Both b_1 and b_2 supervised models outperform their counterpart without using the discrepant reward. They also outperform their counterpart using the self-critical baseline.

6) RNN vs. GRU vs. LSTM as captioning decoder: The results in Table VII show that LSTM performs better than RNN and GRU. This is not surprising, since LSTM has more parameters to explore long-term and short-term memory and has stronger generative ability to produce sentences.

E. Comparison with State-of-the-Arts

1) Quantitative Results: We compare our GLR against some of the recent leading methods on the MSR-VTT dataset. The results are shown in Table VIII. One notable improvement of our method is that we can achieve an on-par performance with other state-of-the-art methods with a much shorter training schedule. Our fully trained model also surpasses all the compared methods on all metrics.

TABLE X
COMPARISON WITH STATE-OF-THE-ART MULTI-MODAL METHODS.

Method	Vision	Motion	Audio	Category	B@4	M	R	C
MA-LSTM [88]	✓	✓	✓	×	36.5	26.5	59.8	41.0
VideoLab [89]	✓	✓	✓	✓	40.7	28.6	61.0	46.5
v2t_navigator [90]	×	✓	✓	✓	42.6	28.8	61.7	46.7
HACA [91]	✓	×	✓	×	43.4	29.5	61.8	49.7
Ours	✓	✓	×	×	**46.9**	**31.2**	**65.7**	**60.6**

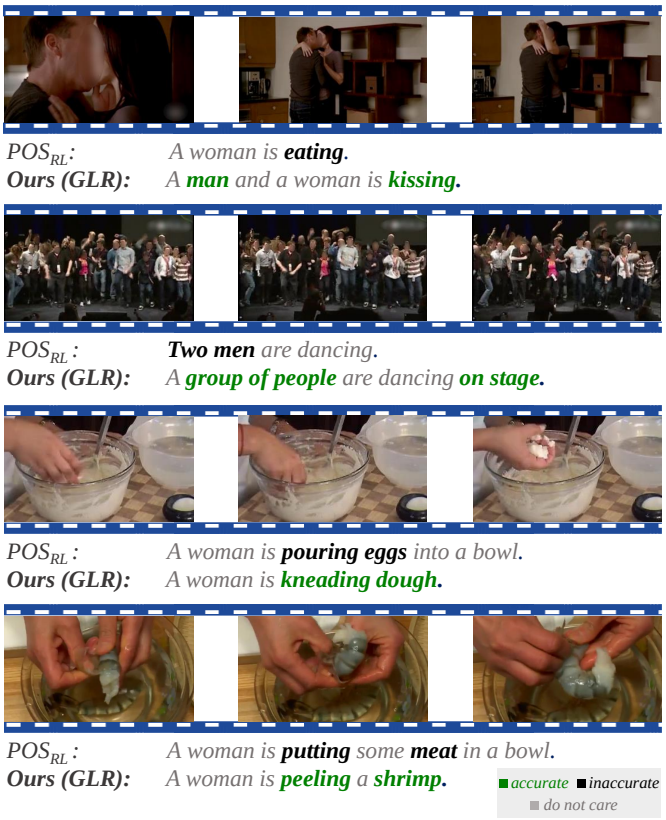


Fig. 10. Qualitative results on MSVD [27]. We present comparisons with state-of-the-art method POS$_{RL}$ [7].

When using the same level of supervision, our margins (model trained by XE) over the next best method (ORG-TRL [22]) are 1.9% on B@4, 1.9% on M, 0.5% on R, and 0.3% on C respectively. We can achieve further performance gain by using DXE on M, R, and C metrics. Note that our results of XE and DXE, reported from the 30th epoch (just used as seeding phase), outperform the state-of-the-art systems with shorter training schedules. If supervised by reinforcement learning (RL), our method also outperforms its counterparts by a significant margin. Our margins (model trained by RL+DR over the next best method (POS$_{RL}$ [7]) are 5.6% on B@4, 2.5% on M, 3.6% on R, and 7.2% on C respectively. All of those previous RL method are trained with self-critical baseline b_{scst}.

We also compare our GLR against some of the recent leading methods on the MSVD dataset. The results are shown in Table IX. When trained by DXE, our margins over the next best method (ORG-TRL [22]) are 3.4% on B@4, 2.2% on M, 1.0% on R, and 0.7% on C respectively. Using reinforcement learning, our margins (model trained by RL+DR) over the next best method (POS$_{RL}$ [7]) are 6.6% on B@4, 4.0% on M, 1.5% on R, and 10.0% on C respectively.

TABLE XI
IMPROVEMENTS OF STATE-OF-THE-ART METHODS USING OUR TRAINING STRATEGY. ↑ INDICATES THE INCREASE FROM THEIR CORRESPONDING ORIGINAL XE OR RL RESULTS.

	Method	B@4	M	R	C
XE	SAAT$_{XE}$	40.5	28.2	60.9	49.1
	ORG-TRL	43.6	29.7	62.1	50.9
	STGraph	40.5	28.3	60.9	47.1
	SBAT	42.9	28.9	61.5	51.6
DXE	SAAT$_{XE}$	41.5$_{\uparrow 1.0}$	29.4$_{\uparrow 1.2}$	61.8$_{\uparrow 0.9}$	50.6$_{\uparrow 1.5}$
	ORG-TRL	44.9$_{\uparrow 1.3}$	30.9$_{\uparrow 1.2}$	63.6$_{\uparrow 1.8}$	53.5$_{\uparrow 2.6}$
	STGraph	41.7$_{\uparrow 1.2}$	29.7$_{\uparrow 1.4}$	62.1$_{\uparrow 1.2}$	48.8$_{\uparrow 1.7}$
	SBAT	44.6$_{\uparrow 1.7}$	30.5$_{\uparrow 1.6}$	63.2$_{\uparrow 1.7}$	53.6$_{\uparrow 2.0}$
RL	HRL	41.3	28.7	61.7	48.0
	PickNet	38.9	27.2	59.5	42.1
	POS$_{RL}$	41.3	28.7	62.1	53.4
	SBAT	41.5	28.4	61.8	53.9
RL+DR	HRL	42.8$_{\uparrow 1.5}$	30.4$_{\uparrow 1.7}$	63.4$_{\uparrow 1.7}$	49.5$_{\uparrow 1.5}$
	PickNet	40.6$_{\uparrow 1.7}$	29.0$_{\uparrow 1.8}$	61.0$_{\uparrow 1.5}$	43.2$_{\uparrow 1.1}$
	POS$_{RL}$	42.5$_{\uparrow 1.2}$	30.4$_{\uparrow 1.7}$	63.9$_{\uparrow 1.8}$	54.9$_{\uparrow 1.5}$
	SBAT	42.7$_{\uparrow 1.2}$	30.5$_{\uparrow 2.1}$	63.4$_{\uparrow 1.6}$	54.8$_{\uparrow 0.9}$

We argue that our improvements are stemmed from the employment of global-local features for strong visual representation for video captioning. In addition, the results indicate that the progressive training, which includes the seeding phase and the boosting phase, is effective to supervise the feature learning for our captioning task. The results from the boosting phase also resonate with our assumption for this proposed training strategy: leveraging CIDEr score to compute the reward with our proposed baseline can further boost the captioning performance.

2) Qualitative Results: Figure 9 demonstrates some qualitative examples on MSR-VTT [1]. Our method leverages the global-local features to achieve a fine-grained description of video contents across frames. Compared to other methods, our method demonstrates improved captioning behavior. For example, our GLR can recognize "group of children" and "soldier", instead of using "man" or "person" to make up sentences. Figure 10 also demonstrates some qualitative examples on MSVD [27]. Those results show that our GLR captures more accurate details (*e.g.*, "kissing", "kneading dough" and "peeling shrimp") for videos when generate captions, while the previous state-of-the-art method POS$_{RL}$ [7] generates wrong

descriptions. More qualitative examples from our method can be found in the attached video.

F. Comparison with Multi-Modal Methods

Comparison with state-of-the-art multi-modal methods is shown in Table X. For example, MA-LSTM [88] uses GoogLeNet [84] to extract visual feature, C3D [92] to extract motion feature and MFCC [93] to extract audio feature. VideoLab [89] and v2t_navigator [90] uses the category feature from the dataset content. But our method outperforms all of those methods without using audio and category multi-modal feature.

V. General Improvements

To evaluate the generalization of our method, we craft our global-local encoding into some of the flexible methods in Table XI and leverage our DXE and DR to optimize their feature learning for video captioning prediction. Table XI demonstrates the improvements of each of the selected methods to their original implementations. For instance, the average performance gains by using DXE range from 0.9% to 2.6%; the margins for methods using RL+DR over the original implementations range from 1.1% to 1.8%. This empirical evidence verifies the power of our systemic design and efficacy. We also plug our progressive training strategy into SBAT [44], a Transformer-based video captioning model, and the results show that our training strategy also works well for Transformer-based encoder-decoders.

VI. Conclusion

In this paper, we approach the video captioning task from a new perspective and propose a GLR framework, namely a global-local representation granularity. We successfully leverage the global-local vision representation to achieve fine-grained captioning expression on video frames. In supervised the proposed GLR, we propose a progressive training strategy, which demonstrates a powerful capacity to boost the captioning performance. Extensive experimental results indicate the effectiveness of our method in comparison with the recent leading methods. For its simplicity and efficacy, we hope that our GLR could serve as a strong baseline for the video captioning task.

References

[1] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *CVPR*, 06 2016.

[2] D. Liu, Y. Cui, Y. Chen, J. Zhang, and B. Fan, "Video object detection for autonomous driving: Motion-aid feature calibration," *Neurocomputing*, vol. 409, pp. 1–11, 2020.

[3] Y. Cui, Z. Cao, Y. Xie, X. Jiang, F. Tao, Y. V. Chen, L. Li, and D. Liu, "Dg-labeler and dgl-mots dataset: Boost the autonomous driving perception," in *WACV*, 2022.

[4] Y. Cui, L. Yan, Z. Cao, and D. Liu, "Tf-blender: Temporal feature blender for video object detection," in *CVPR*, 2021.

[5] D. Liu, Y. Cui, W. Tan, and Y. Chen, "Sg-net: Spatial granularity network for one-stage video instance segmentation," in *CVPR*, 2021.

[6] D. Liu, Z.-J. Zha, H. Zhang, Y. Zhang, and F. Wu, "Context-aware visual policy network for sequence-level image captioning," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1416–1424.

[7] B. Wang, L. Ma, W. Zhang, W. Jiang, J. Wang, and W. Liu, "Controllable video captioning with pos sequence guidance based on gated fusion network," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2641–2650, 2019.

[8] J. Wang, W. Jiang, L. Ma, W. Liu, and Y. Xu, "Bidirectional attentive fusion with context gating for dense video captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7190–7198.

[9] Z.-J. Zha, D. Liu, H. Zhang, Y. Zhang, and F. Wu, "Context-aware visual policy network for fine-grained image captioning," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[10] J. Hou, X. Wu, W. Zhao, J. Luo, and Y. Jia, "Joint syntax representation learning and visual cue translation for video captioning," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8917–8926, 2019.

[11] J. Zhang and Y. Peng, "Object-aware aggregation with bidirectional temporal graph for video captioning," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8319–8328, 2019.

[12] C.-Y. Ma, A. Kadav, I. Melvin, Z. Kira, G. Al-Regib, and H. Graf, "Attend and interact: Higher-order object interactions for video understanding," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6790–6800, 2018.

[13] L. Zhou, Y. Kalantidis, X. Chen, J. J. Corso, and M. Rohrbach, "Grounded video description," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[14] L. Li, Y. Zhang, S. Tang, L. Xie, X. Li, and Q. Tian, "Adaptive spatial location with balanced loss for video captioning," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2020.

[15] J. Zhang and Y. Peng, "Video captioning with object-aware spatio-temporal correlation and aggregation," *IEEE Transactions on Image Processing (TIP)*, vol. 29, pp. 6209–6222, 2020.

[16] T. Jin, Y. Li, and Z. Zhang, "Recurrent convolutional video captioning with global and local attention," *Neurocomputing*, vol. 370, pp. 118–127, 2019.

[17] Y.-H. H. Tsai, S. Divvala, L.-P. Morency, R. Salakhutdinov, and A. Farhadi, "Video relationship reasoning using gated spatio-temporal energy graph," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10 416–10 425, 2019.

[18] X. Wang and A. Gupta, "Videos as space-time region graphs," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, p. 399–417.

[19] L. Fan, W. Wang, S. Huang, X. Tang, and S. Zhu, "Understanding human gaze communication by spatio-temporal graph reasoning," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5723–5732, 2019.

[20] J. Wu, L. Wang, J. Guo, and G. Wu, "Learning actor relation graphs for group activity recognition," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9956–9966, 2019.

[21] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *AAAI*, 2018.

[22] Z. Zhang, Y. Shi, C. Yuan, B. Li, P. Wang, W. Hu, and Z. Zha, "Object relational graph with teacher-recommended learning for video captioning," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13 275–13 285, 2020.

[23] B. Pan, H. Cai, D. Huang, K.-H. Lee, A. Gaidon, E. Adeli, and J. C. Niebles, "Spatio-temporal graph for video captioning with knowledge distillation," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10 867–10 876, 2020.

[24] P. Ghosh, Y. Yao, L. Davis, and A. Divakaran, "Stacked spatio-temporal graph convolutional networks for action segmentation," *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 565–574, 2020.

[25] V. Iashin and E. Rahtu, "Multi-modal dense video captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 958–959.

[26] T. Rahman, B. Xu, and L. Sigal, "Watch, listen and tell: Multi-modal weakly supervised dense event captioning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8908–8917.

[27] D. L. Chen and W. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *ACL*, 2011.

[28] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko, "Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2712–2719.

[29] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele, "Coherent multi-sentence video description with variable level of detail," in *German conference on pattern recognition*. Springer, 2014, pp. 184–195.

[30] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele, "Translating video content to natural language descriptions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 433–440.

[31] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4534–4542.

[32] Y. Chen, S. Wang, W. Zhang, and Q. Huang, "Less is more: Picking informative frames for video captioning," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 358–373.

[33] J. Deng, L. Li, B. Zhang, S. Wang, Z. Zha, and Q. Huang, "Syntax-guided hierarchical attention network for video captioning," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2021.

[34] W. Pei, J. Zhang, X. Wang, L. Ke, X. Shen, and Y.-W. Tai, "Memory-attended recurrent network for video captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8347–8356.

[35] L. Wang, H. Li, H. Qiu, Q. Wu, F. Meng, and K. N. Ngan, "Pos-trends dynamic-aware model for video caption," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2021.

[36] Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, and L. Deng, "Semantic compositional networks for visual captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5630–5639.

[37] X. Long, C. Gan, and G. De Melo, "Video captioning with multi-faceted attention," *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 173–184, 2018.

[38] X. Duan, W. Huang, C. Gan, J. Wang, W. Zhu, and J. Huang, "Weakly supervised dense event captioning in videos," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[39] C. Gan, Z. Gan, X. He, J. Gao, and L. Deng, "Stylenet: Generating attractive visual captions with styles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3137–3146.

[40] K. Yi, C. Gan, Y. Li, P. Kohli, J. Wu, A. Torralba, and J. B. Tenenbaum, "Clevrer: Collision events for video representation and reasoning," *arXiv preprint arXiv:1910.01442*, 2019.

[41] Z. Chen, J. Mao, J. Wu, K.-Y. K. Wong, J. B. Tenenbaum, and C. Gan, "Grounding physical concepts of objects and events through dynamic visual reasoning," *arXiv preprint arXiv:2103.16564*, 2021.

[42] B. Wu, S. Yu, Z. Chen, J. B. Tenenbaum, and C. Gan, "Star: A benchmark for situated reasoning in real-world videos," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[43] L. Baraldi, C. Grana, and R. Cucchiara, "Hierarchical boundary-aware neural encoder for video captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1657–1666.

[44] T. Jin, S. Huang, M. Chen, Y. Li, and Z. Zhang, "SBAT: video captioning with sparse boundary-aware transformer," in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence (IJCAI)*, 2021, pp. 630–636.

[45] Y. Pan, T. Yao, H. Li, and T. Mei, "Video captioning with transferred semantic attributes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6504–6512.

[46] J. Chen, Y. Pan, Y. Li, T. Yao, H. Chao, and T. Mei, "Temporal deformable convolutional encoder-decoder networks for video captioning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 8167–8174.

[47] J. Zhang and Y. Peng, "Hierarchical vision-language alignment for video captioning," in *International Conference on Multimedia Modeling*. Springer, 2019, pp. 42–54.

[48] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, "Jointly modeling embedding and translation to bridge video and language," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4594–4602.

[49] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4507–4515.

[50] Y. Hu, Z. Chen, Z.-J. Zha, and F. Wu, "Hierarchical global-local temporal modeling for video captioning," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 774–783.

[51] Z. Yang, Y. Han, and Z. Wang, "Catching the temporal regions-of-interest for video captioning," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 146–153.

[52] Y. Zheng, Y. Zhang, R. Feng, T. Zhang, and W. Fan, "Stacked multimodal attention network for context-aware video captioning," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2021.

[53] G. Nan, R. Qiao, Y. Xiao, J. Liu, S. Leng, H. Zhang, and W. Lu, "Interventional video grounding with dual contrastive learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2765–2775.

[54] H. Fan and Y. Yang, "Person tube retrieval via language description," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 10 754–10 761.

[55] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

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

[57] H. Chen, K. Lin, A. Maye, J. Li, and X. Hu, "A semantics-assisted video captioning model trained with scheduled sampling," 08 2019.

[58] W. Zhang, Y. Feng, F. Meng, D. You, and Q. Liu, "Bridging the gap between training and inference for neural machine translation," 06 2019.

[59] I. Laina, C. Rupprecht, and N. Navab, "Towards unsupervised image captioning with shared multimodal embeddings," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7414–7424.

[60] S. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *CVPR*, 07 2017.

[61] X. Wang, W. Chen, J. Wu, Y.-F. Wang, and W. Y. Wang, "Video captioning via hierarchical reinforcement learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4213–4222.

[62] R. Pasunuru and M. Bansal, "Reinforced video captioning with entailment rewards," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017, pp. 979–985.

[63] H. Fan, Z. Xu, L. Zhu, C. Yan, J. Ge, and Y. Yang, "Watching a small portion could be as good as watching all: Towards efficient video classification," in *IJCAI International Joint Conference on Artificial Intelligence*, 2018.

[64] H. Fan, L. Zhu, Y. Yang, and F. Wu, "Recurrent attention network with reinforced generator for visual dialog," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 3, pp. 1–16, 2020.

[65] Z. Ren, X. Wang, N. Zhang, X. Lv, and L.-J. Li, "Deep reinforcement learning-based image captioning with embedding reward," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 290–298.

[66] T. Wang, H. Zheng, M. Yu, Q. Tian, and H. Hu, "Event-centric hierarchical representation for dense video captioning," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 31, no. 5, pp. 1890–1900, 2020.

[67] J. Shijie, W. Ping, J. Peiyi, and H. Siping, "Research on data augmentation for image classification based on convolution neural networks," in *2017 Chinese automation congress (CAC)*. IEEE, 2017, pp. 4165–4170.

[68] D. Tran, J. Ray, Z. Shou, S. Chang, and M. Paluri, "Convnet architecture search for spatiotemporal feature learning," *ArXiv*, vol. abs/1708.05038, 2017.

[69] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," 07 2017, pp. 5987–5995.

[70] S. Wiseman and A. Rush, "Sequence-to-sequence learning as beam-search optimization," 06 2016.

[71] W. Pei, J. Zhang, X. Wang, L. Ke, X. Shen, and Y.-W. Tai, "Memory-attended recurrent network for video captioning," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8339–8348, 2019.

[72] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "Bleu: a method for automatic evaluation of machine translation," 10 2002.

[73] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," 01 2005.

[74] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," 01 2004, p. 10.

[75] R. Vedantam, C. Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *CVPR*, 06 2015, pp. 4566–4575.
[76] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, pp. 229–256, 2004.
[77] X. Shi, J. Cai, S. R. Joty, and J. Gu, "Watch it twice: Video captioning with a refocused video encoder," *Proceedings of the 27th ACM International Conference on Multimedia*, 2019.
[78] M. Zolfaghari, K. Singh, and T. Brox, "Eco: Efficient convolutional network for online video understanding," 04 2018.
[79] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4733, 2017.
[80] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *ArXiv*, vol. abs/1502.03167, 2015.
[81] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Gool, "Temporal segment networks: Towards good practices for deep action recognition," *ArXiv*, vol. abs/1608.00859, 2016.
[82] L. Wang, W. Li, and L. Gool, "Appearance-and-relation networks for video classification," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1430–1439, 2018.
[83] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015.
[84] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.
[85] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034, 2015.
[86] B. Wang, L. Ma, W. Zhang, and W. Liu, "Reconstruction network for video captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7622–7631.
[87] Q. Zheng, C. Wang, and D. Tao, "Syntax-aware action targeting for video captioning," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13 093–13 102, 2020.
[88] J. Xu, T. Yao, Y. Zhang, and T. Mei, "Learning multimodal attention lstm networks for video captioning," *Proceedings of the 25th ACM international conference on Multimedia*, 2017.
[89] V. Ramanishka, A. Das, D. H. Park, S. Venugopalan, L. A. Hendricks, M. Rohrbach, and K. Saenko, "Multimodal video description," *Proceedings of the 24th ACM international conference on Multimedia*, 2016.
[90] Q. Jin, J. Chen, S. Chen, Y. Xiong, and A. Hauptmann, "Describing videos using multi-modal fusion," *Proceedings of the 24th ACM international conference on Multimedia*, 2016.
[91] X. Wang, Y. Wang, and W. Y. Wang, "Watch, listen, and describe: Globally and locally aligned cross-modal attentions for video captioning," in *NAACL-HLT*, 2018.
[92] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4489–4497, 2015.
[93] S. W. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken se," 1980.