GL-RG: Global-Local Representation Granularity for Video Captioning

Liqi Yan[1,2,8*], **Qifan Wang**[3*], **Yiming Cui**[4], **Fuli Feng**[5], **Xiaojun Quan**[6], **Xiangyu Zhang**[7] and **Dongfang Liu**[8†]
[1]Fudan University
[2]Westlake University
[3]Meta AI
[4]University of Florida
[5]University of Science and Technology of China
[6]Sun Yat-sen University
[7]Purdue University
[8]Rochester Institute of Technology
yanliqi@westlake.edu.cn, wqfcr@fb.com, dongfang.liu@rit.edu

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 representation across video frames for caption generation, leaving plenty of room for improvement. In this work, we approach the video captioning task from a new perspective and propose a GL-RG framework for video captioning, namely a **G**lobal-**L**ocal **R**epresentation **G**ranularity. Our GL-RG demonstrates three advantages over the prior efforts: 1) we explicitly exploit extensive visual representations from different video ranges to improve linguistic expression; 2) we devise a novel global-local encoder to produce rich semantic vocabulary to obtain a descriptive granularity of video contents across frames; 3) we develop an incremental training strategy which organizes model learning in an incremental fashion to incur an optimal captioning behavior. Experimental results on the challenging MSR-VTT and MSVD datasets show that our DL-RG outperforms recent state-of-the-art methods by a significant margin. Code is available at https://github.com/ylqi/GL-RG.

1 Introduction

Video captioning is of great societal relevance, holding values for many real-world applications, including subtitle generation, blind assistance and autopilot narration. However, isolated video frames may suffer from motion blur or occlusion, which introduces great confusion in visual understanding for the captioning task. Therefore, there is an urgent need to answer a principal problem: how to leverage the rich global-local features regarding to cross-frame coherence and single frame information in video contents to close the gap from visual understanding to language expression?

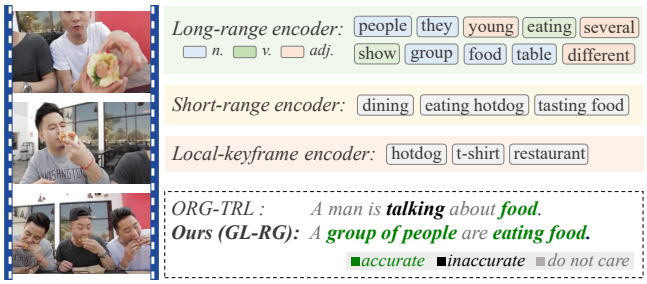

Figure 1: Qualitative examples of drastic scene variations across frames. By using global-local representations in videos, our method achieves fine-grained description of the video contents, in comparison with state-of-the-art methods ORG-TRL.

Despite making significant progress, existing methods for video captioning inadequately capture the local and global representation. Various works apply the deep neural network on raw pixels to build higher-level connections [Wang *et al.*, 2019; Zhang and Peng, 2019]. These methods focus on local object features but neglecting object transformation or interaction. The effort of modeling local object features is a primitive solution for video captioning because the temporal connections across frames are not explored delicately and thus sensitive to the spurious association.

To study the problem of the global-local correlation, other related vision tasks leverage the graph representation using graph neural networks (GNNs). For instance, [Tsai *et al.*, 2019] models object relations by using video spatio-temporal graphs and explicitly builds links between high-level entities. Inspired by the above success, recent video captioning studies extend the graph-based approach and use GNNs to model global-local reasoning [Zhang *et al.*, 2020b; Pan *et al.*, 2020]. Among these works, [Zhang and Peng, 2019] merges local features with the global feature using concatenation; [Ghosh *et al.*, 2020] adds spatio-temporal features

*Equal contributions.
†Corresponding author.

as a separate node in the graph. However, empirical results indicate that using graphs to represent global-local correlation is suboptimal as it often encounters the over-smoothing problem in training which leads to weak results during inference. Alternatively, many video captioning methods intuitively exploit multi-modal fusion ($i.e.$, visual or audio features) to enrich the feature representation in prediction [Rahman *et al.*, 2019]. However, these simple "lumping" approaches inefficiently exploit multi-modal features and struggle to perform joint optimization cross-modality, leaving large room for improvement.

To address the aforementioned problems, we attempt to solve video captioning in a more flexible approach, which exploits the global-local vision representation granularity. Concretely, we make the following contributions:

- We devise a simple framework called GL-RG, namely the global-local representation granularity, which models extensive vision representations and generates rich vocabulary features based on video contents of different ranges.
- We propose a novel global-local encoder, which exploits rich temporal representation for video captioning. The encoder jointly encodes the long-range frames to describe spatio-temporal correspondence, the short-range frames to capture object motion and tendency, and the local keyframe to preserve finer object appearance and location details (Figure 1).
- We introduce an incremental two-phase training strategy. In the first seeding phase, we design a discriminative cross-entropy for non-reinforcement learning, which addresses the problem of human annotation discrepancy. In the second boosting phase, we adapt a discrepant reward for reinforcement learning, which stably estimates a bias of the expected reward for each individual video.
- We evaluate our approach on the MSR-VTT [Xu *et al.*, 2016] and MSVD [Chen and Dolan, 2011] datasets. Extensive experimental results indicate that our method outperforms the latest best systems and uses shorter training schedules.

2 Related Work

Video captioning. Inspired by the success of other vision tasks, the seminal work [Venugopalan *et al.*, 2015] extends the encoder-decoder architecture for the video captioning task. Following the same paradigm, [Chen *et al.*, 2018; Venugopalan *et al.*, 2015] explore the temporal patterns on video using attention mechanisms to depict object movements. [Pei *et al.*, 2019b] devises a MARN method, which generalizes descriptions from a single video to other videos with high semantic similarity. [Hou *et al.*, 2019] develops an idea of feature fusion to guide sentence generation for video content. Different from the existing efforts, we explicitly explore global-local representations for sentence generation.

Global-local representation. To model the global-local vision representation, many methods [Liu *et al.*, 2021a; Zhang *et al.*, 2021; Liu *et al.*, 2020] resort to the sequence learning strategy. [Yao *et al.*, 2015] uses a temporal attention method to depict the global-local connections. [Wang *et al.*, 2019] leverages the decoding hidden states to increase the temporal feature representation. More recently, [Hu *et al.*, 2019; Yang *et al.*, 2017; Zhang and Peng, 2019] exploit the object features to model the object movement across frames. For instance, [Zhang and Peng, 2019] employs a bidirectional temporal graph to capture detailed movements for the salient objects in the video; [Hu *et al.*, 2019] devises a stacked LSTM to encode both the frame-level and object-level temporal information. However, the above work primarily focuses on feature salience from the global contents with less consideration of the global-local representation reasoning. In contrast, we model global-local representations to achieve the lexical granularity, using long-range temporal correspondence, short-range object motion, and local spatial appearances on video contents.

Training strategies. A popular strategy for training video captioning models is "Teacher Forcing" [Williams and Zipser, 1989], which has been widely used in training video captioning tasks [Zhang *et al.*, 2020a]. More recently, many research efforts attempt to explore different training methods to boost captioning performance [Wang *et al.*, 2018b; Rennie *et al.*, 2017; Hou *et al.*, 2019; Ryu *et al.*, 2021]. For instance, [Pasunuru and Bansal, 2017] uses a mixed loss function to optimize the video captioning algorithm, which leverages the weighted combination of cross-entropy and reinforcement learning. Similarly, [Rennie *et al.*, 2017] adopts the paradigm of reinforcement learning and devises a self-critical baseline to reward the model learning to train the video captioning network. Although demonstrating appealing supervision performance [Deng *et al.*, 2021; Liu *et al.*, 2021b], the above methods generally require a complicated pipeline to train with a computation overhead for optimization. Building on the lessons learned from the concurrent approaches, we propose an incremental training strategy, which can easily operate training on our proposed GL-RG. Empirical results indicate that our training strategy can serve as a good addition to the existing training scheme to boost a further training gain.

3 GL-RG

3.1 Overview

The framework of GL-RG is demonstrated in Figure 2. Following [Pan *et al.*, 2020], GL-RG also adopts an encoder-decoder architecture. More specifically, we include a global-local encoder and a captioning decoder. The global-local encoder selects frames of different ranges 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 incremental training strategy translates the vocabulary feature into natural language sentences. We elaborate on the proposed GL-RG below.

3.2 Global-Local Encoder

Our global-local encoder includes three essential parts: long-range encoder, short-range encoder, and keyframe encoder

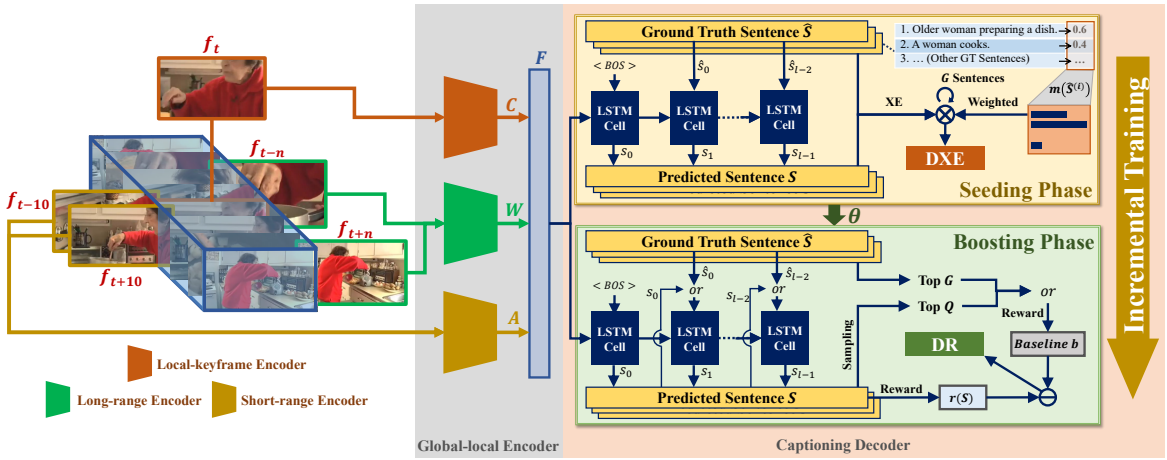

Figure 2: The architectural framework of GL-RG. Our global-local representation encoder includes: 1). the long-range encoder captures temporal correspondence among distant frames ($t-n$ to $t+n$ frames) and makes the cross-frame representations robust to appearance variations and shape deformations; 2). the short-range encoder focuses on motion and tendency, which depicts the local consistency of object movement within a short moment ($t \pm 10$ 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 an incremental 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.

(see Figure 2). Collectively, our encoder can enrich global-local vision representations for video captioning tasks.

Long-range encoder. We encode random global video frames to produce the global vocabulary based on a random keyframe f_t in training. Note that our training iterations will fully saturate the whole video clips, since each iteration will randomly choose different frames (the total number is fixed) from the videos. 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 are 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-classification task. Outputs of the dense layers are defined as:

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

where W is the collection of predicted long-range vocabulary. Since W includes all possibilities of word choices from the ground truths (*i.e.*, MSR-VTT [Xu *et al.*, 2016] and MSVD [Chen and Dolan, 2011]), it can offer the description of temporal contents in video. It is consisted of top K frequently used words (such as verbs, nouns and adjectives, excluding "is", "be" and "do", ext.) from all annotated GT sentences of all videos in the dataseta, *i.e.*, MSR-VTT and MSVD.

Short-range encoder. Our short-range encoder is to capture object motion and tendency. Specifically, simultaneously taking two close neighbours (a.k.a. f_{t-10} and f_{t+10}) of the keyframe, 2D CNN and 3D-Resnet18 [Tran *et al.*, 2017] yield the semantic and movement representations respectively. Afterwards, these representations are stacked and fed into dense layers for action classifications. Given the highest frequency of J actions (in Kinetics-400, UCF101 or HMDB datasets), our output of the short-range encoder is the following.

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

where A is the collection of a_j, which is the predicted confidence of the j_{th} action from the short-range action dataset.

Local-keyframe encoder. The lexical knowledge for the local semantics is learned by a residual network [Xie *et al.*, 2017], which extracts salient object features from the keyframe f_t. Given the number of image classes in the image classification dataset (e.g. ImageNet) is M, the output of our local encoder is:

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

where C is the collection of c_m, which is the predicted confidence of the m_{th} class for this local 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:

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

3.3 Captioning Decoder

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

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

[1] where n is a random range larger than 25 frames.

[2] l (*i.e.*, 30 in our experiment) denotes the maximum length of a sentence.

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 $\hat{s}_{i-1}$ using a random variable $\xi \in \{0, 1\}$:

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

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 $\xi = 1$.

Accordingly, the probability of a predicted word is:

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

where h_i is the hidden state from Eq. (5) and W_o is the weight matrix which maps the hidden state h_i to vocabulary-sized embedding, in order to find a context-matching word in the sentence.

3.4 Incremental Training

Our incremental training includes a seeding phase followed by a boosting phase (see Figure 2). The two training phases fulfill different learning objectives. The seeding phase aims to produce an entrance model to facilitate smooth training in the second phase, while the boosting phase leverages reinforcement learning (RL) to boost the performance gain.

Seeding phase. The existing models [Pei *et al.*, 2019a; Zhang and Peng, 2019; Zhang *et al.*, 2020b] 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 cannot yield the optimal training outcomes*. We thus 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 truths.

Understandably, manually annotated ground truths have severe bias, that is, some ground truth sentences are well-written, 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, METEOR, ROUGE_L [Sharma *et al.*, 2017], and CIDEr [Vedantam *et al.*, 2015]. 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:

$$L_{DXE}(\theta) = -\frac{1}{G}\sum_{j=1}^{G} m(\hat{S}^{(j)}) \log p(\hat{S}^{(j)}|F;\theta), \quad (8)$$

where $m()$ can be deemed as a constant, computed with every ground truth of each sentence. Our DXE loss increases the probability of generating captions with a high metric score by assigning higher weights to well-written ground-truth sentences. The gradient of DXE is calculated by the weighted difference between the prediction and all the target descriptions. Consequently, we DXE 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. Intuitively, we want to encourage predictions with higher scores. Empirical results resonate with our assumption.

Different from the weighted loss entropy (which manually assign weights to all categories 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 captions closer to them.

Boosting phase. After the seeding phase, we employ reinforcement learning with a *discrepant reward (DR)* to further boost the performance of our GL-RD model. To optimize the model parameters θ, conventional methods using reinforcement learning performs a non differentiable reward in training:

$$\begin{aligned} \nabla_\theta L_{DR}(\theta) &= -\sum_S r(S)\nabla_\theta p(S|F;\theta) \\ &= -\mathbb{E}_{S\sim p}[r(S)\nabla_\theta \log p(S|F;\theta)], \end{aligned} \quad (9)$$

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 can be defined as:

$$\nabla_\theta L_{DR}(\theta) = -\mathbb{E}_{S\sim p}[(r(S) - b)\nabla_\theta \log p(S|F;\theta)], \quad (10)$$

where $b \approx E[r(S)]$. Previous self-critical method SCST [Rennie *et al.*, 2017] utilizes the reward of the greedy output at the test time as the baseline b_{scst}, at the cost to run inference again in every training iteration. 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 forward step. Note that either baseline (b_1 or b_2) can reduce the variance of the gradient without changing the expected value of gradient because of $\sum_S b\nabla_\theta p(S|F;\theta) = 0$. When updating gradients, this gradient ∇_θ can be approximated by Monte-Carlo sampling through a single training example. So the final gradient of our discrepant reward is:

$$\nabla_\theta L_{DR}(\theta) \approx -(r(S) - r(S^{b_j}))\nabla_\theta \log p(S|F;\theta), \quad (11)$$

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

Training	Method	Epoch	Feature			MSR-VTT				MSVD			
			Local	Short	Long	B@4	M	R	C	B@4	M	R	C
XE	SA-LSTM [Xu *et al.*, 2016]	100	✓	✓	×	36.3	25.5	58.3	39.9	45.3	31.9	64.2	76.2
	RecNet [Wang *et al.*, 2018a]	-	✓	×	×	39.1	26.6	59.3	42.7	52.3	34.1	69.8	80.3
	ORG-TRL [Zhang *et al.*, 2020b]	-	✓	✓	×	43.6	29.7	62.1	50.9	54.3	36.4	73.9	95.2
	STGraph [Pan *et al.*, 2020]	50	✓	×	×	40.5	28.3	60.9	47.1	52.2	36.9	73.9	93.0
	SGN [Ryu *et al.*, 2021]	-	✓	×	×	40.8	28.3	60.8	49.5	52.8	35.5	72.9	94.3
	O2NA [Liu *et al.*, 2021b]	50	✓	✓	×	41.6	28.5	62.4	51.1	55.4	37.4	74.5	96.4
	RCG [Zhang *et al.*, 2021]	-	✓	✓	×	42.8	29.3	61.7	52.9	-	-	-	-
	Ours (GL-RG)	**30**	✓	✓	✓	**45.5**	**30.1**	**62.6**	**51.2**	**55.5**	**37.8**	**74.7**	**94.3**
DXE	**Ours (GL-RG)**	**30**	✓	✓	✓	**46.9**	**30.4**	**63.9**	**55.0**	**57.7**	**38.6**	**74.9**	**95.9**
RL	HRL [Wang *et al.*, 2018b]	-	✓	×	×	41.3	28.7	61.7	48.0	-	-	-	-
	PickNet [Chen *et al.*, 2018]	300	✓	×	×	38.9	27.2	59.5	42.1	46.1	33.1	69.2	76.0
	POS_{RL} [Wang *et al.*, 2019]	-	✓	✓	×	41.3	28.7	62.1	53.4	53.9	34.9	72.1	91.0
	VRE [Shi *et al.*, 2019]	-	✓	×	×	43.2	28.0	62.0	48.3	51.7	34.3	71.9	86.7
	SAAT_{RL} [Zheng *et al.*, 2020]	200	✓	✓	×	39.9	27.7	61.2	51.0	46.5	33.5	69.4	81.0
RL+DR	**Ours (GL-RG + IT)**	**100**	✓	✓	✓	**46.9**	**31.2**	**65.7**	**60.6**	**60.5**	**38.9**	**76.4**	**101.0**

Table 1: Comparisons with state-of-the-art methods on MSR-VTT and MSVD datasets. 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. "L", "S", and "L" in "Feature" indicate the local, short, and long visual representation. "IT" in our method stands for incremental training, which optimizes the CIDEr metric in boosting phase.

4 Experiments

4.1 Implementation Details

Dataset. We evaluate our GL-RG on MSR-VTT dataset [Xu *et al.*, 2016]. Each video is associated with 20 ground-truth captions given by different workers. We follow the data split of 6513 videos for training, 497 videos for validation, and 2990 videos for testing. We also evaluate our GL-RG on the MSVD dataset [Chen and Dolan, 2011]. We split the dataset into a 1,200 training set, 100 validation set, and 670 testing set by the contiguous index number.

Evaluation Metrics. We evaluate our method on four commonly used metrics BLEU_4, METEOR, ROUGE_L [Sharma *et al.*, 2017], and CIDEr [Vedantam *et al.*, 2015], which are denoted as `B@4`, `M`, `R`, and `C` respectively.

Training setup. Our *long-range encoder* is pre-trained on the video-to-words dataset (K=300 words) extracted from MSR-VTT or MSVD. Our *short-range encoder* is pre-trained on Kinetics-400 dataset [Carreira and Zisserman, 2017], which includes J=400 actions. Our *Local-keyframe encoder* is pre-trained on ImageNet, which includes M=1000 objects. Our decoder is trained with the learning rate of 0.0003 in *the seeding phase*, and 0.0001 in *the boosting phase*. For each video, training is operated on 20 or 17 ground-truth captions for MSR-VTT or MSVD respectively.

4.2 Comparison with State-of-the-Arts

The evaluation results on the MSR-VTT dataset are shown in Table 1. With a much shorter training schedule, we can achieve an on-par performance with other state-of-the-art methods. Our fully trained model also surpasses all the compared methods on all metrics. In addition, when using the same level of supervision, our margins (model trained by XE) over the next best method (ORG-TRL [Zhang *et al.*, 2020b]) are 1.9% on `B@4`, 0.4% 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. We further compare our GL-RG against some of the recent leading methods on the MSVD dataset (see Table 1). When trained by DXE, our margins over the next best method (O2NA [Liu *et al.*, 2021b]) are 2.3% on `B@4`, 1.2% on `M` and 0.4% on `R` respectively. It is worth mentioning that our results of XE and DXE are from the 30^{th} epoch (as seeding phase). It can be seen that they not only outperform those latest best systems, but also use shorter training schedules. Note that all the previous RL methods are trained with self-critical baseline b_{scst}. Figure 3 demonstrates some qualitative examples.

Features			B@4	M	R	C
Local	Short	Long				
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						
✓	✓	×	$36.7\uparrow_{5.0}$	$26.0\uparrow_{2.0}$	$57.9\uparrow_{3.4}$	$42.3\uparrow_{7.0}$
✓	×	✓	$45.1\uparrow_{13.4}$	$29.3\uparrow_{5.3}$	$62.0\uparrow_{7.5}$	$53.0\uparrow_{17.7}$
×	✓	✓	$45.6\uparrow_{13.9}$	$29.3\uparrow_{5.3}$	$62.9\uparrow_{8.4}$	$53.9\uparrow_{18.6}$
✓	✓	✓	$\mathbf{46.9}\uparrow_{15.2}$	$\mathbf{30.4}\uparrow_{6.4}$	$\mathbf{63.9}\uparrow_{9.4}$	$\mathbf{55.0}\uparrow_{19.7}$

Table 2: Comparison of using different features (local, short, and long-range features) in the seeding phase. DXE loss is used. ↑ is the increase from the baselines which use single feature.

4.3 Ablation Study

In this section, we conduct extensive ablation studies to analyze the effects of configurable components in GL-RG.

Global-local features. We measure the performances of our model using different global-local features (see in Table 2). At the higher level of Table 2, we 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. At the lower level of Table 2, we examine the impact of progressively combin-

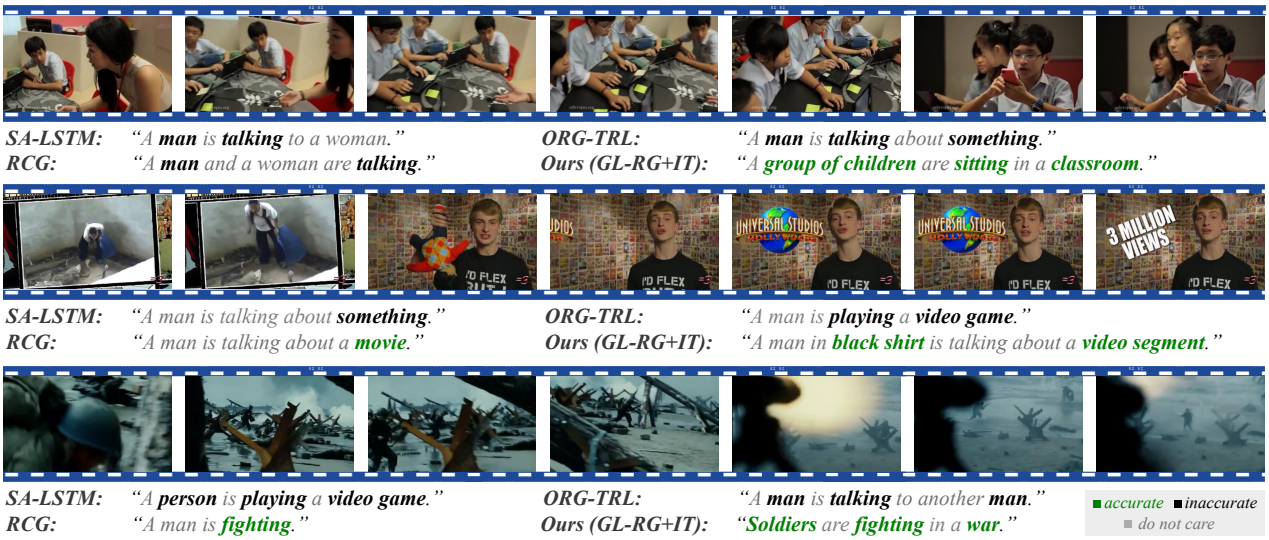

Figure 3: Qualitative results on MSR-VTT. We present comparisons with state-of-the-art methods SA-LSTM, ORG-TRL, and RCG.

	$m(\hat{S}^{(i)})$	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**$\uparrow_{3.8}$

Table 3: Comparison of using different weighting metric $m(\hat{S}^{(i)})$ for DXE in the seeding phase. $\uparrow$ and $\downarrow$ indicates the performance change from the methods trained by XE.

	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 4: Comparison of using XE or DXE-trained entrance model in the boosting phase. $\uparrow$ indicates the increase from the seeding phase.

ing different features together. Our full model using all three features outperforms all other counterparts.

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 (see Table 3). On `R` and `C`, models trained by different DXE loss all outperform the counterpart trained by XE. Meanwhile, using CIDEr as the metric weight in DXE training obtain the best results on all metrics.

Incremental training analysis. We investigate if the incremental training could effectively improve our method performance. Table 4 shows the results from the boosting phase increase steadily from the seeding phase. The boosting phase starting from the seeding phase using DXE gets higher scores than its counterpart using the XE. This proves 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.

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$\uparrow_{43.7}$
Ours(b_1)	46.4$\uparrow_{32.9}$	30.5$\uparrow_{14.4}$	65.0$\uparrow_{18.9}$	58.1$\uparrow_{45.4}$
Ours(b_2)	**46.9**$\uparrow_{33.4}$	**31.2**$\uparrow_{15.1}$	**65.7**$\uparrow_{19.6}$	**60.6**$\uparrow_{47.9}$

Table 5: The performance of using different baselines as discrepant reward. $\uparrow$ indicates the increase from "-" (without baseline). We use DXE in the seeding phase.

Using b_1 vs. b_2 in the boosting phase. Table 5 shows the results of models using different discrepant (b_1 and b_2) rewards compared with self-critical baseline (b_{scst}). Using the reward with b_2 baseline based on top Q sentences sampled by the model can help our GL-RG to achieve a better performance than using the reward with b_1 baseline which is based on G ground-truth sentences. Both models with b_1 or b_2 outperform methods without using the discrepant reward or with the self-critical reward (b_{scst}).

5 Conclusion

Video captioning is an important research topic, which has various downstream applications. In this paper, we propose a GL-RG framework for video captioning, which leverages the global-local vision representation to achieve fine-grained captioning on video contents with an incremental training strategy. Experimental results on two benchmarks demonstrate the effectiveness of our approach. In future, we plan to explore dynamic weighting scheme to capture the preferences on different granularities. We also plan to investigate integrating more multi-modal information.

References

[Carreira and Zisserman, 2017] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, 2017.

[Chen and Dolan, 2011] David L. Chen and W. Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011.

[Chen *et al.*, 2018] Yangyu Chen, Shuhui Wang, Weigang Zhang, and Qingming Huang. Less is more: Picking informative frames for video captioning. In *ECCV*, 2018.

[Deng *et al.*, 2021] Chaorui Deng, Shizhe Chen, Da Chen, Yuan He, and Qi Wu. Sketch, ground, and refine: Top-down dense video captioning. In *CVPR*, 2021.

[Ghosh *et al.*, 2020] Pallabi Ghosh, Yi Yao, L. Davis, and Ajay Divakaran. Stacked spatio-temporal graph convolutional networks for action segmentation. In *WACV*, 2020.

[Hou *et al.*, 2019] Jingyi Hou, X. Wu, Wentian Zhao, Jiebo Luo, and Y. Jia. Joint syntax representation learning and visual cue translation for video captioning. In *ICCV*, 2019.

[Hu *et al.*, 2019] Yaosi Hu, Zhenzhong Chen, Zheng-Jun Zha, and Feng Wu. Hierarchical global-local temporal modeling for video captioning. In *ACM MM*, 2019.

[Liu *et al.*, 2020] Dongfang Liu, Yiming Cui, Yingjie Chen, Jiyong Zhang, and Bin Fan. Video object detection for autonomous driving: Motion-aid feature calibration. *Neurocomputing*, 409:1–11, 2020.

[Liu *et al.*, 2021a] Dongfang Liu, Yiming Cui, Wenbo Tan, and Yingjie Chen. Sg-net: Spatial granularity network for one-stage video instance segmentation. In *CVPR*, 2021.

[Liu *et al.*, 2021b] Fenglin Liu, Xuancheng Ren, Xian Wu, Bang Yang, Shen Ge, and Xu Sun. O2NA: an object-oriented non-autoregressive approach for controllable video captioning. In *ACL/IJCNLP*, 2021.

[Pan *et al.*, 2020] Boxiao Pan, Haoye Cai, Dean Huang, Kuan-Hui Lee, Adrien Gaidon, E. Adeli, and Juan Carlos Niebles. Spatio-temporal graph for video captioning with knowledge distillation. In *CVPR*, 2020.

[Pasunuru and Bansal, 2017] Ramakanth Pasunuru and Mohit Bansal. Reinforced video captioning with entailment rewards. In *EMNLP*, 2017.

[Pei *et al.*, 2019a] W. Pei, Jiyuan Zhang, Xiangrong Wang, Lei Ke, Xiaoyong Shen, and Yu-Wing Tai. Memory-attended recurrent network for video captioning. In *CVPR*, 2019.

[Pei *et al.*, 2019b] Wenjie Pei, Jiyuan Zhang, Xiangrong Wang, Lei Ke, Xiaoyong Shen, and Yu-Wing Tai. Memory-attended recurrent network for video captioning. In *CVPR*, 2019.

[Rahman *et al.*, 2019] Tanzila Rahman, Bicheng Xu, and Leonid Sigal. Watch, listen and tell: Multi-modal weakly supervised dense event captioning. In *ICCV*, 2019.

[Rennie *et al.*, 2017] Steven Rennie, E. Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017.

[Ryu *et al.*, 2021] Hobin Ryu, Sunghun Kang, Haeyong Kang, and C. Yoo. Semantic grouping network for video captioning. In *AAAI*, 2021.

[Sharma *et al.*, 2017] Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR*, abs/1706.09799, 2017.

[Shi *et al.*, 2019] Xiangxi Shi, Jianfei Cai, Shafiq R. Joty, and Jiuxiang Gu. Watch it twice: Video captioning with a refocused video encoder. In *ACM MM*, 2019.

[Tran *et al.*, 2017] Du Tran, Jamie Ray, Zheng Shou, S. Chang, and Manohar Paluri. Convnet architecture search for spatiotemporal feature learning. *ArXiv*, abs/1708.05038, 2017.

[Tsai *et al.*, 2019] Yao-Hung Hubert Tsai, Santosh Kumar Divvala, Louis-Philippe Morency, Ruslan Salakhutdinov, and Ali Farhadi. Video relationship reasoning using gated spatio-temporal energy graph. In *CVPR*, 2019.

[Vedantam *et al.*, 2015] Ramakrishna Vedantam, C. Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015.

[Venugopalan *et al.*, 2015] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *ICCV*, 2015.

[Wang *et al.*, 2018a] Bairui Wang, Lin Ma, Wei Zhang, and Wei Liu. Reconstruction network for video captioning. In *CVPR*, 2018.

[Wang *et al.*, 2018b] Xin Wang, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. Video captioning via hierarchical reinforcement learning. In *CVPR*, 2018.

[Wang *et al.*, 2019] Bairui Wang, L. Ma, W. Zhang, Wenhao Jiang, Junling Wang, and W. Liu. Controllable video captioning with pos sequence guidance based on gated fusion network. In *ICCV*, 2019.

[Williams and Zipser, 1989] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[Xie *et al.*, 2017] Saining Xie, Ross Girshick, Piotr Dollar, Z. Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[Xu *et al.*, 2016] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016.

[Yang *et al.*, 2017] Ziwei Yang, Yahong Han, and Zheng Wang. Catching the temporal regions-of-interest for video captioning. In *ACM MM*, 2017.

[Yao *et al.*, 2015] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *CVPR*, 2015.

[Zhang and Peng, 2019] Junchao Zhang and Y. Peng. Object-aware aggregation with bidirectional temporal graph for video captioning. In *CVPR*, 2019.

[Zhang *et al.*, 2020a] Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. Bridging the gap between training and inference for neural machine translation. In *IJCAI*, 2020.

[Zhang *et al.*, 2020b] Z. Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Z. Zha. Object relational graph with teacher-recommended learning for video captioning. In *CVPR*, 2020.

[Zhang *et al.*, 2021] Ziqi Zhang, Zhongang Qi, Chunfeng Yuan, Ying Shan, Bing Li, Ying Deng, and Weiming Hu. Open-book video captioning with retrieve-copy-generate network. In *CVPR*, 2021.

[Zheng *et al.*, 2020] Qi Zheng, Chaoyue Wang, and D. Tao. Syntax-aware action targeting for video captioning. In *CVPR*, 2020.