Augmented Partial Mutual Learning with Frame Masking for Video Captioning

Ke Lin [1,2]**, Zhuoxin Gan** [2]**, Liwei Wang** [1]

[1]Peking University, China
[2] Samsung Research China - Beijing (SRC-B), China
{ke17.lin, zhuoxin1.gan}@samsung.com, wanglw@pku.edu.cn,

Abstract

Recent video captioning work improves greatly due to the invention of various elaborate model architectures. If multiple captioning models are combined into a unified framework not only by simple more ensemble, and each model can benefit from each other, the final captioning might be boosted further. Jointly training of multiple model have not been explored in previous works. In this paper, we propose a novel Augmented Partial Mutual Learning (APML) training method where multiple decoders are trained jointly with mimicry losses between different decoders and different input variations. Another problem of training captioning model is the ”one-to-many” mapping problem which means that one identical video input is mapped to multiple caption annotations. To address this problem, we propose an annotation-wise frame masking approach to convert the ”one-to-many” mapping to ”one-to-one” mapping. The experiments performed on MSR-VTT and MSVD datasets demonstrate our proposed algorithm achieves the state-of-the-art performance.

Introduction

Video captioning refers to the task that generating a description of a given video automatically and it combines computer vision and natural language processing (NLP) in a unified framework (Venugopalan et al. 2015). It can be widely used in video retrieval, video recommendation, disabled supporting and scene understanding. Following Venugopalan et al. (Venugopalan et al. 2015), recent video captioning works are almost based on encoder-decoder framework in which a 2D or 3D CNN with other transformation modules are regarded as the encoder and a sequential module (e.g. GRU) serves as the decoder. The improvement of their caption generation performance mainly comes from more elaborate visual features(Aafaq et al. 2019; Zhang and Peng 2019). Despite the great evolution on their encoders, the decoders of them are still either RNN (LSTM/GRU) based or the Transformer based(Vaswani et al. 2017). However, designing a new general sequence-modeling network is not trivial. Therefore, the motivation of taking advantage of these existed sequence-modeling networks simultaneously to enhance the decoding robustness is natural. One way to address this issue is ensemble, which independently trains multiple models with different architectures or identity models with different initializations and integrates them into a powerful model at inference stage. Nevertheless, this method is incapable of improving the performance of each single model in the ensemble group because of its overlooking of mutual interaction during training and its resource exhausting property at inferencing phase also limits its practical application.

In this paper, we propose to use mutual learning (Zhang et al. 2018) to jointly train multiple decoders and make them guide each other during training, thus their independent caption generation performance get improved significantly. Mutual learning is a technique to transfer knowledge among a group of peer models, deriving from Knowledge Distillation (Hinton, Vinyals, and Dean 2015) which trains a teacher model and a student model simultaneously to improve the student model by the guidance of the bigger teacher model. But unlike Knowledge Distillation, mutual learning does not restrict the size or the type of models and makes all the training models as the mutual teachers to guide each other via minimizing a mimicry loss which measures the prediction discrepancy among peer-models with respect to the same inputs. This mimicry loss can be regarded as a kind of regularization which improves the generalization ability of the model.

Since video captioning is a complicated multimodal task, original mimicry loss is not adequate to regularize this complex multimodal interaction. To further improve the capability of this mimicry regularization, we propose a novel Augmented Partial Mutual Learning (APML) in which instead of feeding the same video data to the peer-models as the input, multiple peer-decoders calculates the mimicry loss with different augmented video data via Auto-Augment (Cubuk et al. 2018). Besides, we add an extra intrinsic loss which calculates the KL divergence between the outputs respect to normal input and augmented input for each decoder. In our method, each peer-model is just a part of the whole captioning model and shares the left part (e.g. Embedding module), which forms another regularization over the shared modules, analogous to sharing backbone in multi-task learning. Note that our method using multiple decoders is different from the ensemble method. Because, instead of seeking to derive an more powerful integrated ensemble from a group of single models, we explore the improving of each single model with the help of other peer models in the group. Therefore, with our train-

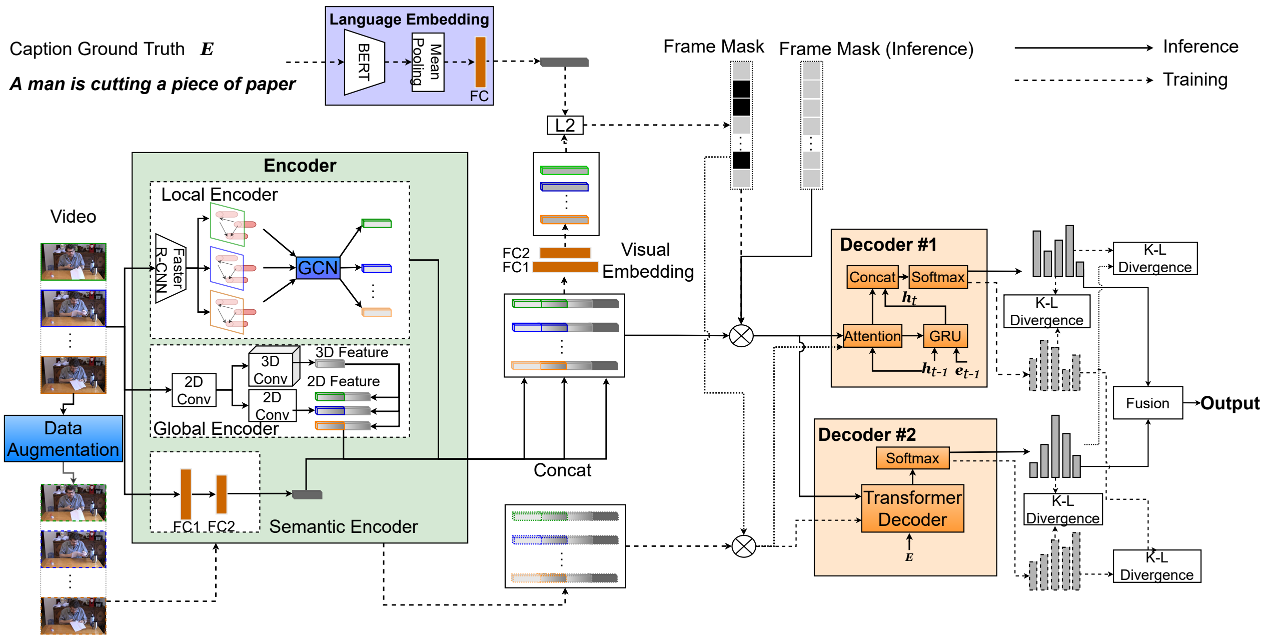

Figure 1: The proposed video captioning framework. The encoder contains a global encoder, a local encoder and a semantic encoder. Multiple decoders are trained by augmented partial mutual learning. Besides, frame masking is applied to address the ”one-to-many” mapping problem.

ing scheme, every single model in the group gets improved contrast to ensemble in which the performance of each single model remains weak. Moreover, the model trained with APML can be further boosted via ensemble.

The commonly used datasets for video captioning usually contain several caption annotations for one video. The most frequently used method regards each video-caption pair as a distinct training sample. This method might confuse the video captioning model when trained with cross-entropy loss, because one identical input is mapped to several different targets. This is the so-called ”one-to-many” mapping problem (Duan et al. 2018). To the best of our knowledge, this problem has not been discussed in previous video captioning papers. To address this problem, we propose a novel annotation-wise frame masking approach which embeds the video features and caption annotations into a mutual hidden space and mask the frames which have lower similarities with the language embedding, thus converting the one-to-many mapping to one-to-one mapping. Our frame masking is only used on the training stage, on the inference stage, we feed all the frame features to the model since there is no annotation guiding. Experiment shows that even though the gap between training and inference exists, our frame masking method indeed improves the captioning model on both MSVD and MSR-VTT datasets.

In summary, the main contributions of the proposed approach are three-folds:

- We propose a novel Augmented Partial Mutual Learning(APML) strategy to train multiple decoders with shared encoder simultaneously for video captioning task. Experiments show that this training strategy makes obviously improvement for every intact model with one of these peer decoders and these models can be further boosted by ensemble method.
- A novel annotation-wise frame masking approach is proposed to alleviate the influence of the one-to-many mapping problem.
- The proposed approach achieves the state-of-the-art performances on MSR-VTT and MSVD datasets.

Methods

Given the training video $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ and its caption annotations $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_H\}, \mathbf{y}_i = [y_1, \ldots, y_T], y_t \in \mathcal{Y}$, video captioning model learns the mapping from $\mathbf{X}$ to $\mathbf{Y}$, where N denotes the number of selected frames of a video, H denotes the number of caption annotations for one video, T is the maximal length of a caption, $\mathcal{Y}$ is the vocabulary set. Next, we describe our framework in detail. We illustrate our algorithm in Figure 1 which is best viewed in color.

Multiple Encoders

Global Encoder. We use an Efficient Convolutional Network (ECO) (Zolfaghari, Singh, and Brox 2018) pre-trained on Kinetics-400 dataset (Kay et al. 2017) as the global encoder to extract global appearance and motion feature. ECO is a convolutional network framework for video action recognition, which is comprised of a 3D network to extract the temporal visual feature and a 2D network focusing on

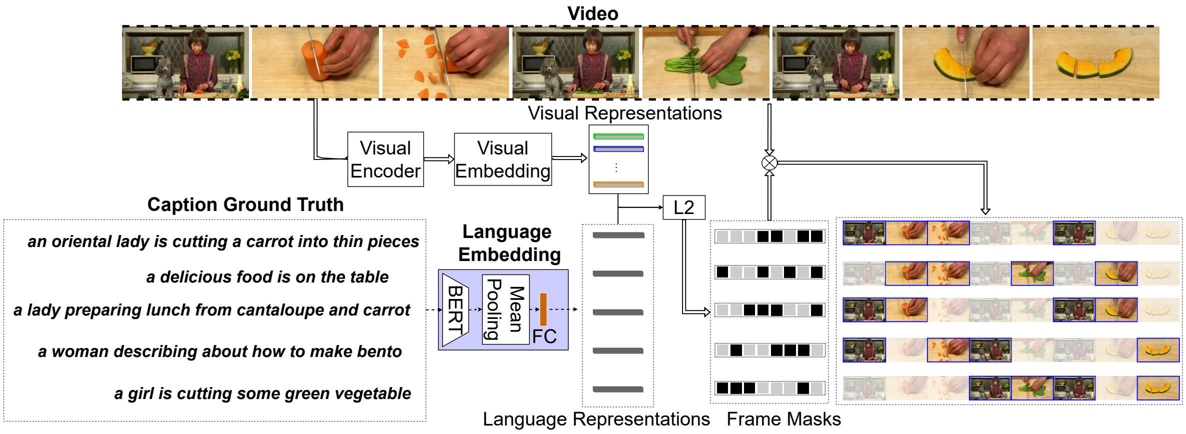
Video
Visual Representations
Visual Encoder
Visual Embedding
Caption Ground Truth
an oriental lady is cutting a carrot into thin pieces
a delicious food is on the table
a lady preparing lunch from cantaloupe and carrot
a woman describing about how to make bento
a girl is cutting some green vegetable
Language Embedding
BERT
Mean Pooling
FC
L2
Language Representations
Frame Masks

Figure 2: The detailed illustration of frame masking.

short-term or static action feature in parallel. We directly use the output of parallel 2D network in ECO as the 2D feature $\mathbf{F} = \{\mathbf{f}_1, \dots, \mathbf{f}_N\}$ where $\mathbf{f}_i \in \mathcal{R}^{D_{2D}}$. Besides, we use the max-pooled outputs of 3D network as the 3D feature $\mathbf{g} \in \mathcal{R}^{D_{3D}}$. Then, we concatenate the 3D feature to each frame 2D feature to get final visual feature which can be represented by $\mathbf{V}^{(g)} = \{[\mathbf{f}_1; \mathbf{g}], \dots, [\mathbf{f}_N; \mathbf{g}]\}, \mathbf{v}_i^{(g)} \in \mathcal{R}^{D_{2D}+D_{3D}}$ where $[;]$ is the concatenation operator.

Local Encoder. In order to detect visual semantic information of the video frame, inspired by (Yao et al. 2018), we utilize Faster R-CNN and two simple classifier to get the features and locations of objects, the attributes of objects and relationships between objects. Following (Xu et al. 2017), we set up 150 object labels, 300 attribute labels and 50 relation labels. With the object label, attribute label and relationship label, we can construct the spatial scene graph for every video frame. Inspired by (Kipf and Welling 2017), in our framework, we use GCN to encode each spatial scene graph. To avoid redundancy propagation, we embed the attributes node and object node jointly into a feature vector. Therefore, for each object, we concatenate its mean pooled convolution feature of Faster R-CNN $\mathbf{v}_i^{(lm)}$ and its embedded class label $\mathbf{v}_i^{(lc)}$ and its attribute class $\mathbf{v}_i^{(la)}$ to form its final input node feature $\mathbf{v}_i^{(l)} = [\mathbf{v}_i^{(lm)}; \mathbf{v}_i^{(lc)}; \mathbf{v}_i^{(la)}] \in \mathcal{R}^{D_l}$.

In our framework, for each object node, instead of using the same transformation weights to transform all its neighboring nodes' feature, we set up different transformation weight matrixes for the object node itself and its object-neighbor nodes and subject-neighbor nodes respectively. And for the relationship edge, we use different weight matrix based on the role the object node plays. Consequently, our spatial GCN module can be formulated by:

$$\mathbf{v'}_i^{(l)} = \sigma(\mathbf{W}_s \mathbf{v}_i^{(l)} + \textstyle\sum_{\mathbf{v}_j^{(l)} \in \mathcal{N}\left(\mathbf{v}_i^{(l)}\right)} [A(\mathbf{v}_i^{(l)}, \mathbf{v}_j^{(l)}) \times \\ (\mathbf{W}_{(sub,obj)} \mathbf{v}_j^{(l)} + \mathbf{W}_{(in,out)} e^r_{(\mathbf{v}_i^{(l)}, \mathbf{v}_j^{(l)})})]) \tag{1}$$

$$A(\mathbf{v}_i^{(l)}, \mathbf{v}_j^{(l)}) = softmax\left(\mathbf{v}_i^{(l)T} \mathbf{W}_\mathbf{a} \mathbf{v}_j^{(l)}\right) \tag{2}$$

where $\mathbf{v'}_i^{(l)}$ is the output of the GCN layer, $\mathcal{N}\left(\mathbf{v}_i^{(l)}\right)$ is the set of features of the neighbors of node i, σ is a non-linear operation, $\mathbf{W}_{(sub,obj)}$ represents the transformation matrix for the neighbors node based on their role, if the neighbor node j plays the subject role in the relationship between i and j, the transformation matrix for j is $\mathbf{W}_{sub}$, otherwise $\mathbf{W}_{obj}$. $\mathbf{W}_{(in,out)}$ is the edge transformation matrix, based on the edge's role, the model chooses different matrix. e^r is the embedded relationship label. In order to weigh different neighbors and relationships of node i, we add a product attention layer into GCN. $A(\mathbf{v}_i^{(l)}, \mathbf{v}_j^{(l)})$ is the formulated attention layer, $\mathbf{W}_\mathbf{a}$ is the learning attention layer weights.

Semantic Encoder. Inspired by several prior works (Chen et al. 2019; Yao et al. 2017; Wang et al. 2019b), we also extract semantic information using a semantic encoder. Following Chen's work (Chen et al. 2019), we manually select the Q most frequent words from training and validation set as attributes. The semantic encoder consists of a multi-layer-perceptron on top of the ECO framework. Attribute detection is treated as a multi-label classification task. Following (Chen et al. 2019), we concatenate the predicted probability distribution of attributes and the probability distribution of ECO as the semantic feature. We denote the semantic feature as $\mathbf{v}_i^{(s)} \in \mathcal{R}^{D_s}$

Feature Concatenation. The final feature vector of frame i is the concatenation of global feature, local feature and semantic feature $\mathbf{v}_i = [\mathbf{v}_i^{(g)}; \mathbf{v}_i^{(l)}; \mathbf{v}^{(s)}] \in \mathcal{R}^{D_{2D}+D_{3D}+D_l+D_s}$

Multiple Decoders

We apply three types of decoders in this study, which are GRU based, LSTM based and Transformer (Vaswani et al. 2017) based.

Mutual Learning. Suppose we have m decoders, each one can be GRU based, LSTM based or Transformer based. We denote these decoders as $\Theta_1, \dots, \Theta_m$. For each decoder Θ_i, we apply a cross-entropy loss,

$$L_c(\Theta_i) = -\sum_{h=1}^{H}\sum_{t=1}^{T} log(p_{\Theta_i}(y_{h,t}|y_{h,1:t-1}, \mathbf{V})) \quad (3)$$

where $y_{h,t}$ is the t th word of the h th caption annotation of the video, p_{Θ_i} is the posterior probability of decoder Θ_i given previous words $y_{h,1:t-1}$.

Inspired by (Zhang et al. 2018), we use posterior probability of other decoders to improve the generalization of decoder Θ_i, and we use the Kullback Leibler (KL) divergence to quantify the match of network's predictions, thus we get another loss L_e:

$$\begin{aligned} L_e(\Theta_i) = &\sum_{j=1,j\neq i}^{m}\sum_{h=1}^{H}\sum_{t=1}^{T}[D_{KL}(\\ &p_{\Theta_j}(y_{h,t}|y_{h,1:t-1}, \mathbf{V})||p_{\Theta_i}(y_{h,t}|y_{h,1:t-1}, \mathbf{V}))] \end{aligned} \quad (4)$$

Augmented Partial Mutual Learning. To facilitate each decoder, mutual learning leverages the extrinsic guidance from other decoders. In this study, we propose to use intrinsic guidance from other input variations. Specifically, we perform data augmentation on the video and compute the K-L divergence of the probability distributions between original data and augmented data as the intrinsic loss. The posterior entropy of the decoder could be further reduced due to the introduction of this intrinsic guidance. We use AutoAugment (Cubuk et al. 2018) to augment the video data for K times, the augmented video feature is denoted as $\{\tilde{\mathbf{V}}_1, \ldots, \tilde{\mathbf{V}}_K\}$. The intrinsic loss is obtained by the following equation:

$$\begin{aligned} L_a(\Theta_i) = &\textstyle\sum_{j=1}^{K}\sum_{h=1}^{H}\sum_{t=1}^{T}[D_{KL}(\\ &p_{\Theta_i}(y_{h,t}|y_{h,1:t-1}, \mathbf{V})||p_{\Theta_i}(y_{h,t}|y_{h,1:t-1}, \tilde{\mathbf{V}}_j))] \end{aligned} \quad (5)$$

The extrinsic loss is updated to:

$$\begin{aligned} L_e(\Theta_i) = &\sum_{j=1,j\neq i}^{m}\sum_{h=1}^{H}\sum_{t=1}^{T}[\\ &D_{KL}(p_{\Theta_j}(y_{h,t}|y_{h,1:t-1}, \mathbf{V})||p_{\Theta_i}(y_{h,t}|y_{h,1:t-1}, \mathbf{V}))+ \\ &\sum_{c=1}^{K} D_{KL}(p_{\Theta_j}(y_{h,t}|y_{h,1:t-1}, \tilde{\mathbf{V}}_c)||p_{\Theta_i}(y_{h,t}|y_{h,1:t-1}, \tilde{\mathbf{V}}_c))] \end{aligned} \quad (6)$$

Traning with this strategy, multi-decoders benefit from the mutual guidence intrinsicly and extrinsicly, and the shared encoder is guided by more than one decoder, which also makes the encoder more robust.

Frame Masking

To overcome the "one-to-many" mapping problem, we propose a frame masking strategy to convert the "one-to-many" mapping to "one-to-one" mapping. First, we embed each caption ground truth $\{\mathbf{y}_1, \ldots, \mathbf{y}_H\}$ of a video to a high dimensional space through a language embedding module. The language embedding module consists of a BERT (Delvin et al. 2018) model[1] pretrained on BooksCorpus (Yukun Zhu and Fidler 2015) and English Wikipedia, a mean pooling layer and a fully connected layer. The language embeddings are denoted as $\{\mathbf{z}_1^{(l)}, \ldots, \mathbf{z}_H^{(l)}\} \in \mathcal{R}^{D_e}$. Video feature is also embedded into a mutual high dimensional space through a visual embedding module composed of two fully connected layers. The visual embedding is denoted as $[\mathbf{z}_1^{(v)}, \ldots, \mathbf{z}_N^{(v)}] \in \mathcal{R}^{D_e}$. For each language embedding $\mathbf{z}_i^{(l)}$, we compute the Mean Square Error (MSE) between $\mathbf{z}_i^{(l)}$ and $[\mathbf{z}_1^{(v)}, \ldots, \mathbf{z}_N^{(v)}] \in \mathcal{R}^{D_e}$ to get a distance vector $\mathbf{d} \in \mathcal{R}^N$ and a loss L_f:

$$\mathbf{d}_i = [d_{i,1}, \ldots, d_{i,j}, \ldots, d_{i,N}], d_{i,j} = \sum_{D_e} ||z_i^{(l)} - z_j^{(v)}||^2 \quad (7)$$

$$L_f = \sum_{h=1}^{H}\sum_{j=1}^{N} d_{i,j} \quad (8)$$

Lower d_j represents the j th frame has higher correlation with the i th caption ground truth. If we keep the frames with lower d and mask the frames with higher d, we can get a distinct "frame mask" for each caption ground truth. Different caption annotation of one video is mapped to a unique frame mask, in other words, we get an "one-to-one" mapping. The frame mask obtained by the original video data is also applied for the augmented video.

We propose two ways of frame masking: soft masking and hard masking.

Soft Masking. Soft masking means that multiplying the visual feature of each frame with a weight, while the weight is the negative correlated with the MSE loss between visual embedding of that frame and language embedding. We define the weights as follows:

$$\mathbf{w}_s = \frac{1}{\mathbf{d}} \quad (9)$$

And the masked video feature $\hat{\mathbf{V}}_s$ is:

$$\hat{\mathbf{V}}_s = \text{diag}(\mathbf{w}_s) \times \mathbf{V} \quad (10)$$

where $\text{diag}(\mathbf{w}_s)$ is a diagonal matrix.

Hard Masking. Hard masking means we select the top F frames of lower r with caption annotation and mask the rest frames. That is to say, $\mathbf{w}$ in equation (9) is:

$$\mathbf{w}_h = [w_j], w_j = \begin{cases} 0, & if\ r_j\ is\ not\ top\ F \\ 1, & if\ r_j\ is\ top\ F \end{cases} \quad (11)$$

Masked video feature $\hat{\mathbf{V}}_h$ is obtained by Equation (10) with $\mathbf{w}_h$.

During inference, because caption annotation is not available and no frame should be masked, the frame mask is an all one vector.

We illustrate the hard frame masking approach as Fig 2. Each caption ground truth is feed to the language embedding module, the predicted language representations are compared with visual representations to get the frame masks. For example, the caption annotation "a delicious food is on the table" only contains the objects of "food" and "table", while major objects in frame #1, frame #4 and frame #6 are the "woman" and the "dog", so these frames are masked by our frame masking approach. Only the feature of frames with higher correlation with the caption annotations are feed to the following decoders.

[1] https://github.com/huggingface/transformers

Dataset	Decoder Type	GE	SE	LE	FM	BLEU-4	ROUGE-L	METEOR	CIDEr
MSVD	GRU	√	×	×	×	52.4	72.3	35.8	88.6
		√	√	×	×	53.4	73.2	36.3	96.0
		√	√	√	×	54.9	74.1	37.3	97.1
		√	√	√	√	**54.9**	**75.0**	**38.7**	**101.2**
	LSTM	√	×	×	×	51.3	71.8	35.7	88.0
		√	√	×	×	53.0	73.1	35.9	95.3
		√	√	√	×	54.3	73.5	37.0	96.3
		√	√	√	√	**54.7**	**74.8**	**38.4**	**100.6**
	Transformer	√	×	×	×	50.4	71.2	35.4	88.1
		√	√	×	×	52.7	72.8	35.6	95.5
		√	√	√	×	53.3	74.2	38.1	97.7
		√	√	√	√	**54.7**	**74.4**	**38.2**	**101.0**
MSR-VTT	GRU	√	×	×	×	37.4	58.9	26.6	40.3
		√	√	×	×	40.8	61.3	28.4	48.3
		√	√	√	×	41.3	61.5	28.8	49.2
		√	√	√	√	**42.6**	**61.9**	**29.1**	**49.5**
	LSTM	√	×	×	×	34.0	56.6	25.3	37.1
		√	√	×	×	40.2	60.8	28.7	48.0
		√	√	√	×	40.8	61.1	29.0	48.6
		√	√	√	√	**41.4**	**62.2**	**29.5**	**49.6**
	Transformer	√	×	×	×	34.3	57.0	25.7	37.5
		√	√	×	×	40.8	61.2	28.9	48.7
		√	√	√	×	41.3	61.5	29.0	48.9
		√	√	√	√	**41.9**	**62.6**	**29.9**	**49.8**

Table 1: Results of an ablation study on multiple encoders on MSVD and MSR-VTT dataset. ”GE”, ”SE”, ”LE” and ”FM” are the abbreviations for global encoder, semantic encoder, local encoder and frame masking.

Metric	Network Types		Net 1			Net 2			Ensemble		
	Net 1	Net 2	Ind	ML	APML	Ind	ML	APML	Ind	ML	APML
BLEU-4	GRU	GRU	54.9	55.7	**58.0**	54.9	56.5	**57.9**	55.3	57.5	**58.8**
	Transformer	Transformer	54.7	54.2	**55.4**	54.6	54.5	**55.8**	54.8	55.2	**56.9**
	LSTM	LSTM	54.3	55.1	**56.0**	54.0	54.5	**55.7**	55.1	55.3	**56.8**
	GRU	LSTM	55.0	55.5	**56.7**	54.6	55.5	**57.0**	55.9	56.5	**58.1**
	LSTM	Transformer	54.4	55.1	**56.2**	54.2	54.5	**55.5**	55.2	56.0	**57.2**
	GRU	Transformer	54.9	56.3	**56.7**	54.7	55.0	**56.0**	55.5	55.9	**57.9**
ROUGE-L	GRU	GRU	75.0	75.4	**76.2**	75.2	75.4	**76.1**	75.5	76.0	**76.9**
	Transformer	Transformer	74.4	74.9	**75.9**	74.4	75.0	**75.2**	74.6	75.4	**76.0**
	LSTM	LSTM	75.2	75.4	**76.3**	74.8	75.1	**76.0**	76.2	76.5	**76.8**
	GRU	LSTM	75.3	75.3	**76.1**	75.2	75.4	**76.2**	75.8	76.0	**77.0**
	LSTM	Transformer	74.7	75.2	**75.8**	75.2	75.6	**75.9**	75.4	75.7	**76.4**
	GRU	Transformer	75.0	75.6	**76.1**	74.4	75.4	**75.5**	75.2	75.5	**76.5**
METEOR	GRU	GRU	38.7	38.9	**39.2**	38.6	38.8	**39.3**	38.9	39.2	**39.7**
	Transformer	Transformer	38.2	38.4	**38.7**	38.1	38.2	**38.7**	38.5	38.8	**39.6**
	LSTM	LSTM	38.0	38.6	**39.4**	38.1	38.6	**39.3**	38.6	39.3	**39.9**
	GRU	LSTM	38.4	38.8	**39.6**	38.7	38.9	**39.5**	39.2	39.5	**39.8**
	LSTM	Transformer	38.0	38.3	**38.9**	37.8	38.1	**38.7**	38.4	38.9	**39.5**
	GRU	Transformer	38.7	38.8	**39.3**	38.2	38.6	**39.1**	38.8	39.0	**39.8**
CIDEr	GRU	GRU	101.2	103.8	**108.3**	101.0	102.4	**107.5**	102.0	105.8	**109.5**
	Transformer	Transformer	101.0	102.8	**107.6**	101.1	102.8	**105.0**	102.3	104.6	**108.0**
	LSTM	LSTM	100.8	101.6	**107.1**	101.3	102.8	**104.5**	103.4	105.4	**108.2**
	GRU	LSTM	102.2	104.8	**108.2**	103.0	103.4	**107.9**	104.0	106.2	**109.1**
	LSTM	Transformer	101.7	102.7	**106.4**	102.4	103.6	**107.1**	102.5	104.2	**108.1**
	GRU	Transformer	101.2	103.0	**108.1**	101.1	102.3	**104.0**	103.2	105.0	**108.7**

Table 2: Comparison of the CIDEr of independent training, Mutual Learning and Augmented Partial Mutual Learning on MSVD dataset. ML denotes mutual learning, APML denotes augmented partial mutual learning, Ind denotes independent training.

Objective

The overall loss of decoder Θ_i is the combination of L_c, L_e, L_a and L_f:

$$L(\Theta_i) = L_c(\Theta_i) + \lambda_1 L_e(\Theta_i) + \lambda_2 L_a(\Theta_i) + \lambda_3 L_f(\Theta_i) \quad (12)$$

Evaluation and Results

Datasets and Implementation Details

We conduct experiments on two benchmark datasets which are Microsoft Research Video Description Corpus (MSVD) and Microsoft Research video to text (MSR-VTT).

MSVD. It contains 1970 YouTube short video clips in 10 seconds to 25 seconds and each video clip depicts a single activity. Each video clip has about 40 English descriptions. We use the public splits which take 1200 video clips for training, 100 clips for validation and 670 clips for testing.

MSR-VTT. We use the initial version of MSR-VTT, referred as MSR-VTT-10K which has 10k video clips and each video clip has 20 descriptions annotated by 1327 workers from Amazon Mechanical Turk. MSR-VTT has 200k video-caption pairs and 29316 unique words. Similar as (Wang et al. 2018a), we split 6513 video clips for training, 497 clips for validation and 2990 clips for testing.

We follow the standard caption pre-processing procedure including converting all words to lower case, tokenizing on white space, clipping sentences over 30 words and filterring words which occur at least five times. The final vocabulary sizes are 5663 for MSVD dataset and 8781 for MSR-VTT dataset. We use standard automatic evaluation metrics including BLEU (Papineni et al. 2002), METEOR (Denkowski and Lavie 2014), ROUGE-L (Lin 2004) and CIDEr (Vedantam, Zitnick, and Parikh 2014).

We uniformly sample $N = 16$ frames for each video. We select top 10 proposals with higher output probabilities for each frame. We pre-train our scene graph construction model with an Adam optimizer and the learning rate is 5×10^{-4}. The batch size is 64 and the dropout rate is 0.3, the word embedding dimension $e = 512$. For GRU and LSTM decoder, the model size and all hidden size are 512. For transformer decoder, the layer number is 6, the number of head is 8 and the model dimension is 512. We train the captioning model using an Adam optimizer. We set hyper-parameters by $Q = 300$, $K = 10$, $\lambda_1 = \lambda_2 = 1 \times 10^3$, $\lambda_3 = 1$.

Quantitative Results of Multiple Encoders and Frame Masking

Table 1. shows the ablation results for multiple encoders and frame masking. The models are trained independently without mutual learning. Fusing the features extracted by global encoder, local encoder and semantic encoder has better performance than removing any one or two encoders for different datasets and different decoder types. This result demonstrates that the proposed multiple encoders can obtain more concrete video representations. For GRU, LSTM and Transformer decoders and both two datasets, frame masking can enhance all metrics consistently. This result verifies that converting ”one-to-many” mapping to ”one-to-one” mapping by frame masking can alleviate the noise and confusion brought by ”one-to-many” mapping.

Quantitative Results of APML

Table 2. shows the result of CIDEr of training multiple decoders by mutual learning, augmented partial mutual learning or independently on MSVD dataset. Regardless of the type of each decoder, the performances of decoders are boosted by mutual learning compared with independently training, and this result is consistent with prior work (Zhang et al. 2018). Augmented partial mutual learning can further increase the single model performance by the introduction of consistency loss between different input variations. It is natural to ensemble the outputs of multiple decoders. Augmented partial mutual learning also achieves best performance among ensemble results. We have also performed the experiment of training a model with AutoAugment without mutual learning loss on MSVD dataset. The results are [GRU: CIDEr (101.7), METEOR (39.0); Transformer: CIDEr (101.5), METEOR (38.6)]. Comparing these results with the results in Table 2, these results are only slightly better than training a model independently without mutual learning or AutoAugment [GRU: CIDEr (101.2), METEOR (38.7); Transformer: CIDEr (101.0), METEOR (38.2)]. The results of APML are much better than others which demonstrate that the improvement is because of the augmented mutual learning loss instead of AutoAugment itself.

Benchmarking Results

Table 3. and Table 4. show the benchmarking results on MSVD and MSR-VTT datasets. We compare our single model and ensemble model results trained by augmented partial mutual learning with several latest state-of-the-art methods. For MSR-VTT dataset, transformer and ”GRU&transformer” are the best single model and ensemble models respectively. All metrics obtained by our method outperform other state-of-the-art methods. For MSVD dataset, GRU and ”GRU&GRU” are the best single model and ensemble models. Most metrics of our approach achieve the best performances.

Ablation Study for Frame Masking

Table 5. shows the result of the ablation study on frame masking using GRU decoder on MSVD dataset. The performance is slightly enhanced by soft masking and attention mechanism compared with no masking. Hard masking has better performance than soft masking and masking 4 frames achieves the best performance. Note the different distribution during training and testing might cause performance degradation. Table 5. shows that if too many frames are masked (8 frames and 16 frames), the difference of the distribution during training and testing are higher and the performances drop significantly. So we only masking 4 frames. We have also tried generating a caption first and using this caption to generate frame masking, then generate the final caption using the frame masking during testing, thus forcing the data distribution during testing to be identical with that during training. The result of this approach is lower than the proposed way. So we simply use the all-one frame masking vector during testing.

Note that even though frame masking looks a bit like the attention mechanism, but they are totally different in the following three aspects. First, the goal of attention is to attend

	BLEU-1	**BLEU-2**	**BLEU-3**	**BLEU-4**	**METEOR**	**ROUGE-L**	**CIDEr**
OA-BTG (Zhang and Peng 2019)	-	-	-	56.9	36.2	-	90.6
JSRL+VCT (Hou et al. 2019)	-	-	-	52.8	36.1	71.8	87.8
L-HOCA-UBT (Jin et al. 2019)	-	-	-	52.9	35.5	72.0	86.1
SAVC (Chen et al. 2019)	-	-	-	**61.8**	37.8	76.8	103.0
ORG-TRL (Zhang et al. 2020)	-	-	-	54.3	36.4	73.9	95.2
ST (Pan et al. 2020)	-	-	-	52.2	36.9	73.9	93.0
SAAT (Zheng, Wang, and Tao 2020)	-	-	-	46.5	33.5	69.4	81.0
PMI-CAP (Chen et al. 2020)	-	-	-	54.6	36.4	-	95.1
Ours: Single Model	**86.4**	**76.8**	**67.5**	58.0	**39.2**	76.2	**108.3**
Ours: Ensemble	**86.8**	**77.3**	**68.2**	58.8	**39.7**	**76.9**	**109.5**

Table 3: Benchmarking result on MSVD dataset.

	BLEU-1	**BLEU-2**	**BLEU-3**	**BLEU-4**	**METEOR**	**ROUGE-L**	**CIDEr**
OA-BTG (Zhang and Peng 2019)	-	-	-	41.4	28.2	-	46.9
JSRL+VCT (Hou et al. 2019)	-	-	-	42.3	29.7	62.8	49.1
L-HOCA-UBT (Jin et al. 2019)	-	-	-	44.6	29.5	62.6	49.8
SAVC (Chen et al. 2019)	-	-	-	43.8	28.9	62.4	51.4
ORG-TRL (Zhang et al. 2020)	-	-	-	43.6	28.8	62.1	50.9
ST (Pan et al. 2020)	-	-	-	40.5	28.3	60.9	47.1
SAAT (Zheng, Wang, and Tao 2020)	79.6	65.9	52.1	39.9	27.7	61.2	51.0
PMI-CAP (Chen et al. 2020)	-	-	-	42.1	28.8	-	49.4
Ours: Single Model	**82.9**	**67.5**	**53.6**	**43.8**	**30.3**	**63.6**	**52.2**
Ours: Ensemble	**83.4**	**68.3**	**54.4**	**45.7**	**31.4**	**64.4**	**52.7**

Table 4: Benchmarking results on MSR-VTT dataset.

on different frames when generating a new word given previous words. While frame masking is proposed to address the ”one-to-many” mapping problem. Second, attention weights are obtained by the visual feature and previous word embedding, but frame masking taking visual feature and the whole sentence as input. Third, attention weights vary with the position of generated words, but frame masking is fixed for one training sample. Frame masking is also different with PickNet (Chen et al. 2018), because the picked frames of PickNet are identical for each caption annotation.

Ablation Study for Number of Decoders

Table 6. shows the result of an ablation study for the number of decoders trained using augmented partial mutual learning on MSVD dataset. By enlarging the number of decoders, the performance only increases slightly, but the time complexity and memory cost of training are multiplied. Thus, we use two decoders for all experiments in this study.

Conclusion

We propose an APML training method where multiple decoders are trained jointly with a shared encoder for video captioning. Multi-decoders benefit from the mutual guidence intrinsicly and extrinsicly. Furthermore, we propose an annotation-wise frame masking approach to address the ”one-to-many” mapping problem. The experiments performed on MSVD and MSR-VTT datasets demonstrate that the proposed framework achieves the state-of-the-art performance.

Masking Type	**CIDEr**	**METEOR**
No Masking	96.4	37.1
No Masking + Attention	97.1	37.3
Randomly Masking 2 frames	97.5	37.2
Randomly Masking 4 frames	96.8	37.0
Soft Masking	97.7	37.5
Hard Masking 2 frames	100.1	38.2
Hard Masking 4 frames	**101.2**	**38.7**
Hard Masking 8 frames	100.8	38.6
Hard Masking 12 frames	98.6	38.0

Table 5: Ablation study on frame masking.

Number of Decoders	**CIDEr**	**METEOR**
2 Decoders	108.3	39.3
3 Decoders	108.8	39.2
4 Decoders	**109.0**	39.3
5 Decoders	107.9	**39.4**

Table 6: Ablation study on number of decoders

Acknowledgements

This work was supported by National Key R&D Program of China (2018YFB1402600).

References

Aafaq, N.; Akhtar, N.; Liu, W.; Gilani, S. Z.; and Mian, A. 2019. Spatio-Temporal Dynamics and Semantic Attribute Enriched Visual Encoding for Video Captioning. In *CVPR*.

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*.

Barbu, A.; Bridge, A.; Burchill, Z.; Coroian, D.; Dickinson, S.; Fidler, S.; Michaux, A.; Mussman, S.; Narayanaswamy, S.; Salvi, D.; Schmidt, L.; Shangguan, J.; Siskind, J. M.; Waggoner, J.; Wang, S.; Wei, J.; Yin, Y.; and Zhang, Z. 2012. Video in sentences out. *arXiv preprint arXiv:1204.2742* .

Chen, H.; Lin, K.; Maye, A.; Li, J.; and Hu, X. 2019. A Semantics-Assisted Video Captioning Model Trained with Scheduled Sampling. *arXiv preprint arXiv:1909.00121* .

Chen, S.; Chen, J.; and Jin, Q. 2017. Generating Video Descriptions with Topic Guidance. In *ICMR*.

Chen, S.; Jiang, W.; Liu, W.; and Jiang, Y.-G. 2020. Learning Modality Interaction for Temporal Sentence Localization and Event Captioning in Videos. In *ECCV*.

Chen, S.; and Jiang, Y.-G. 2019. Motion Guided Spatial Attention for Video Captioning. In *AAAI*.

Chen, Y.; Wang, S.; Zhang, W.; and Huang, Q. 2018. Less Is More: Picking Informative Frames for Video Captioning. In *ECCV*.

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2018. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805:09501* .

Delvin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810:04805* .

Denkowski, M.; and Lavie, A. 2014. METEOR universal: Language specific translation evaluation for any target language. In *EACL 2014 Workshop on Statistical Machine Translation*.

Duan, X.; Huang, W.; Gan, C.; and Wang, J. 2018. Weakly Supervised Dense Event Captioning in Videos. In *NeurIPS*.

Fang, K.; Zhou, L.; Jin, C.; Zhang, Y.; Weng, K.; Zhang, T.; and Fan, W. 2019. Fully Convolutional Video Captioning with Coarse-to-Fine and Inherited Attention. In *AAAI*.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503:02531* .

Hou, J.; Wu, X.; Zhao, W.; Luo, J.; and Jia, Y. 2019. Joint Syntax Representation Learning and Visual Cue Translation for Video Captioning. In *ICCV*.

Iashin, V.; and Rahtu, E. 2020. A Better Use of Audio-Visual Cues: Dense Video Captioning with Bi-modal Transformer. In *BMVC*.

Jin, T.; Huang, S.; Li, Y.; and Zhang, Z. 2019. Low-Rank HO-CA: Efficient High-Order Cross-Modal Attention for Video Captioning. In *EMNLP*.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; Suleyman, M.; and Zisserman, A. 2017. The Kinetics Human Action Video Dataset. *arXiv preprint arXiv:1705.06950* .

Kipf, T. N.; and Welling, M. 2017. Semi-supervised Classification with Graph Convolutional Networks. In *ICLR*.

Kojima, A.; Tamura, T.; and Fukunaga, K. 2002. Natural Language Description of Human Activites from Video Images Based on Concept Hierarchy of Actions. *International Journal of Computer Vision* .

Lin, C.-Y. 2004. ROUGE: a package for automatic evaluation of summaries. In *Proceedings of Workshop on Text Summarization Branches Out*.

Lin, K.; Gan, Z.; and Wang, L. 2020a. Multi-modal Feature Fusion with Feature Attention for VATEX Captioning Challenge 2020. *arXiv preprint arXiv:2006.03315* .

Lin, K.; Gan, Z.; and Wang, L. 2020b. Semi-supervised Learning for Video Captioning. In *Findings of EMNLP*.

Ma, C.-Y.; Kadav, A.; Melvin, I.; Kira, Z.; AlRegib, G.; and Graf, H. P. 2018. Grounded Objects and Interactions for Video Captioning. *arXiv preprint arXiv:1711.06354* .

Pan, B.; Cai, H.; Huang, D.-A.; Lee, K.-H.; Gaidon, A.; Adeli, E.; and Niebles, J. C. 2020. Spatio-Temporal Graph for Video Captioning with Knowledge Distillation. In *CVPR*.

Pan, Y.; Yao, T.; Li, H.; and Mei, T. 2017. Video Captioning with Transferred Semantic Attributes. In *CVPR*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*.

Pei, W.; Zhang, J.; Wang, X.; Ke, L.; Shen, X.; and Tai, Y.-W. 2019. Memory-Attended Recurrent Network for Video Captioning. In *CVPR*.

Rohrbach, M.; Qiu, W.; Titov, I.; Thater, S.; Pinkal, M.; and Schiele, B. 2013. Translating video content to natural language descriptions. In *ICCV*.

Song, J.; Gao, L.; Guo, Z.; Liu, W.; Zhang, D.; and Shen, H. T. 2017. Hierarchical LSTM with Adjusted Temporal Attention for Video Captioning. In *IJCAI*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. In *NeurIPS*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All You Need. In *NeurIPS*.

Vedantam, R.; Zitnick, L. C.; and Parikh, D. 2014. CIDEr: Consensus-based image description evaluation. In *CVPR*.

Venugopalan, S.; Rohrbach, M.; Donahue, J.; Mooney, R.; Darrell, T.; and Saenko, K. 2015. Sequence to Sequence - Video to Text. In *ICCV*.

Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *CVPR*.

Wang, B.; Ma, L.; Zhang, W.; Jiang, W.; Wang, J.; and Liu, W. 2019a. Controllable Video Captioning with POS Sequence Guidance Based on Gated Fusion Network. In *ICCV*.

Wang, B.; Ma, L.; Zhang, W.; and Liu, W. 2018a. Reconstruction Network for Video Captioning. In *CVPR*.

Wang, J.; Wang, W.; Huang, Y.; Wang, L.; and Tan, T. 2018b. Multimodal Memory Modelling for Video Captioning. In *CVPR*.

Wang, X.; Wu, J.; Zhang, D.; Su, Y.; and Wang, W. Y. 2019b. Learning to compose topic-aware mixture of experts for zero-shot video captioning. In *AAAI*.

Xu, D.; Zhu, Y.; Choy, C. B.; and Fei-Fei, L. 2017. Scene graph generation by iterative message passing. In *CVPR*.

Yao, T.; Pan, Y.; Li, Y.; and Mei, T. 2018. Exploring Visual Relationship for Image Captioning. In *ECCV*.

Yao, T.; Pan, Y.; Li, Y.; Qiu, Z.; and Mei, T. 2017. Boosting image captioning with attributes. In *ICCV*.

Yukun Zhu, Ryan Kiros, R. Z. R. S. R. U. A. T.; and Fidler, S. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*.

Zhang, J.; and Peng, Y. 2019. Object-aware Aggregation with Bidirectional Temporal Graph for Video Captioning. In *CVPR*.

Zhang, Y.; Xiang, T.; Hospedales, T. M.; and Lu, H. 2018. Deep Mutual Learning. In *CVPR*.

Zhang, Z.; Shi, Y.; Yuan, C.; Li, B.; Wang, P.; Hu, W.; and Zha, Z. 2020. Object Relational Graph with Teacher-Recommended Learning for Video Captioning. In *CVPR*.

Zheng, Q.; Wang, C.; and Tao, D. 2020. Syntax-Aware Action Targeting for Video Captioning. In *CVPR*.

Zhou, L.; Kalantidis, Y.; Chen, X.; Corso, J. J.; and Rohrbach, M. 2018a. Grounded Video Description. *arXiv preprint arXiv:1812.06587* .

Zhou, L.; Zhou, Y.; Corso, J. J.; Socher, R.; and Xiong, C. 2018b. End-to-End Dense Video Captioning with Masked Transformer. In *CVPR*.

Zhu, Z.; Xue, Z.; and Yuan, Z. 2018. Topic-guided attention for image captioning. *arXiv preprint arXiv:1807.03514* .

Zolfaghari, M.; Singh, K.; and Brox, T. 2018. Eco: Efficient convolutional network for online video understanding. In *Proceedings of the European misc on Computer Vision (ECCV)*, 695–712.