frontiers
in Robotics and AI

ORIGINAL RESEARCH
published: 30 September 2020
doi: 10.3389/frobt.2020.475767

A Semantics-Assisted Video Captioning Model Trained With Scheduled Sampling

Haoran Chen[1], *Ke Lin*[2], *Alexander Maye*[3], *Jianmin Li*[1] and *Xiaolin Hu*[1*]

[1] The State Key Laboratory of Intelligent Technology and Systems, Department of Computer Science and Technology, Beijing National Research Center for Information Science and Technology, Institute for Artificial Intelligence, Tsinghua University, Beijing, China, [2] Samsung Research China, Beijing, China, [3] Department of Neurophysiology and Pathophysiology, University Medical Center, Hamburg, Germany

Given the features of a video, recurrent neural networks can be used to automatically generate a caption for the video. Existing methods for video captioning have at least three limitations. First, semantic information has been widely applied to boost the performance of video captioning models, but existing networks often fail to provide meaningful semantic features. Second, the Teacher Forcing algorithm is often utilized to optimize video captioning models, but during training and inference, different strategies are applied to guide word generation, leading to poor performance. Third, current video captioning models are prone to generate relatively short captions that express video contents inappropriately. Toward resolving these three problems, we suggest three corresponding improvements. First of all, we propose a metric to compare the quality of semantic features, and utilize appropriate features as input for a semantic detection network (SDN) with adequate complexity in order to generate meaningful semantic features for videos. Then, we apply a scheduled sampling strategy that gradually transfers the training phase from a teacher-guided manner toward a more self-teaching manner. Finally, the ordinary logarithm probability loss function is leveraged by sentence length so that the inclination of generating short sentences is alleviated. Our model achieves better results than previous models on the YouTube2Text dataset and is competitive with the previous best model on the MSR-VTT dataset.

Keywords: video captioning, sentence-length-leveraged loss, semantic assistance, RNN, scheduled sampling

OPEN ACCESS

Edited by:
Xiaolan Fu,
Institute of Psychology (CAS), China

Reviewed by:
Lei Zhang,
Microsoft, United States
Cuixia Ma,
Chinese Academy of Sciences (CAS), China

****Correspondence:***
Xiaolin Hu
xlhu@tsinghua.edu.cn

Specialty section:
This article was submitted to Sensor Fusion and Machine Perception, a section of the journal Frontiers in Robotics and AI

Received: *31 May 2019*
Accepted: *17 August 2020*
Published: *30 September 2020*

Citation:
Chen H, Lin K, Maye A, Li J and Hu X (2020) A Semantics-Assisted Video Captioning Model Trained With Scheduled Sampling. Front. Robot. AI 7:475767. doi: 10.3389/frobt.2020.475767

1. INTRODUCTION

Video captioning aims to automatically generate a concise and accurate description for a video. It requires techniques both from computer vision (CV) and natural language processing (NLP). Deep learning (DL) methods for sequence-to-sequence learning are able to learn the map from discrete color arrays to dense vectors, which is utilized to generate natural language sequences without the interference of humans. These methods produced impressive results on this task compared with the results yielded by manually crafted features.

It has gained increasing attention in video captioning that the semantic meaning of a video is critical and beneficial for an RNN to generate annotations (Pan et al., 2016; Gan et al., 2017). Keeping semantic consistency between video content and video description helps to refine a generated sentence in semantic richness (Gao et al., 2017). But few researches have explored methods to obtain video semantic features, metrics to measure their quality and the relation between video captioning performance and meaningfulness of semantic features.

Several training strategies have been used to optimize video captioning models, such as the Teacher Forcing algorithm and CIDEnt-RL (Pasunuru and Bansal, 2017b). The Teacher Forcing algorithm is a simple and intuitive way to train RNNs. But it suffers from the discrepancy between training, which utilizes ground truth to guide word generation at each step, and inference, which samples from the model itself at each step. Reinforcement learning (RL) techniques have also been adopted to improve the training process of video captioning. CIDEnt-RL is one of the best RL algorithms, but it is extremely time-consuming to calculate metrics for every batch. In addition, the improvement on different metrics is unbalanced. In other words, the improvements on other metrics are not as large as that on the specific metrics optimized directly.

The commonly used loss function for video captioning is comprised of the logarithm of probabilities of target correct words (Donahue et al., 2015; Venugopalan et al., 2015). A long sentence tends to bring high loss to the model, as each additional word reduces the joint probability by roughly at least one order of magnitude. In contrast, a short sentence with few words has a relatively low loss. Thus, a video captioning model is prone to generate short sentences after being optimized by a log likelihood loss function. Excessively short annotations may neither be able to describe a video accurately nor express the content of a video in a rich language.

We propose to improve solutions to the video captioning task in three aspects. Firstly, we use mean average precision (mAP) as the metric to evaluate the quality of semantic information. By virtue of the evaluation metric, we build our semantic detection network (SDN) with a proper scale and the best inputs that brings the best performance, and, consequently, SDN is able to produce meaningful and accurate semantic features for a video. Secondly, we take advantage of a scheduled sampling method to train our video captioning model, which searches extreme points in the RNN state space more extensively as well as bridges the gap between training process and inference (Bengio et al., 2015). Thirdly, we optimize our model by a sentence-length-modulated loss function, which encourages the model to generate longer captions with more detail.

Our implementation, available on GitHub[1], is based on the TensorFlow deep learning framework.

2. RELATED WORKS

2.1. Image Captioning

The encoder-decoder paradigm has been widely applied by researchers in image captioning since it was introduced to machine translation (Cho et al., 2014). It has become a mainstream method in both image captioning and machine translation (Mao et al., 2014; Vinyals et al., 2015). Inspired by successful attempts to employ attention in machine translation (Bahdanau et al., 2015) and object detection (Ba et al., 2015), models that are able to attend to key elements in an image are investigated for the purpose of generating high-quality image annotations. Semantic features (You et al., 2016) and object features (Anderson et al., 2018) are incorporated into attention mechanisms as heuristic information to guide selective and dynamic attendance of salient segments in images. RL techniques, which optimize specific metrics of a model directly, are also adopted to enhance the performance of image captioning models (Rennie et al., 2017). Graph Convolutional Networks (GCNs) have been introduced to cooperate with RNN to integrate both semantic and spatial information into image encoders in order to generate efficient representations of an image (Yao et al., 2018). Stimulated by the success of the Transformer model in machine translation, researchers extend it to a multimodal model for image captioning (Yu et al., 2019), which utilizes multi-view visual features to further improve the performance. Multi-level relationships between image regions are learnt and both low- and high-level features are exploited at the decoding stage in the Meshed Transformer with memory for image captioning (Cornia et al., 2019).

2.2. Video Captioning

Though both image captioning and video captioning are multi-modal tasks, video captioning is probably harder than the former one, as videos show not only spatial features but also temporal correlations.

Following the successful adoption of the encoder-decoder paradigm in image captioning, multimodal features of videos are fed into a sequence-to-sequence model to generate video descriptions with the assistance of pretrained models in image classification (Donahue et al., 2015; Venugopalan et al., 2015). In order to alleviate the semantic inconsistency between the video content and the generated caption, visual features and semantic features of a video are mapped to a common embedding space so that semantic consistency may be achieved by minimizing the Euclidean distance between these two embedded features (Pan et al., 2016). A model named POS generates video captions with Part-of-Speech (POS) information and multiple representations of video clips (Wang et al., 2019a). MARN exploits a memory structure to explore the relation between a word and its various visual contexts across the training data (Pei et al., 2019). JSRL-VCT manages to generate video descriptions by corporating visual representations and syntax representations (Hou et al., 2019). GRU-EVE captures rich temporal dynamics in video features by Short Fourier Transform, and extracts semantic information from an object detector (Aafaq et al., 2019). Zheng et al. (2020) propose a Syntax-Aware Action Targeting (SAAT) component to learn an action and its subjects that exist in a video for better semantic consistency in captioning.

RNN, especially LSTM, can be extended by integrating high-level tags or attributes of video with visual features of the video through embedding and element-wise addition/multiplication operations (Gan et al., 2017). Yu et al. (2016) exploit a sentence generator that is built upon an RNN module to model language, a multimodal layer to integrate different modal information, and an attention module to dynamically select salient features

[1] https://github.com/WingsBrokenAngel/Semantics- AssistedVideoCaptioning/tree/master

from the input. The output of a sentence generator is fed into a paragraph generator for describing a relatively long video with several sentences.

Following the attention mechanism introduced by Xu et al. (2015), Gao et al. (2017) capture the salient structure of video with the help of visual features of the video and context information provided by LSTM. Although bottom-up (Anderson et al., 2018) and top-down attention (Ramanishka et al., 2017) have been proposed for image captioning, selectively focusing on salient regions in an image is, to some extent, similar to picking key frames in a video (Chen et al., 2018). Wang et al. (2018) explore crossmodal attention at different granularity levels and capture global temporal structures as well as local temporal structures implied in multimodal features to assist the generation of video captions.

Due to the lack of labeled video data and the abundance of unlabeled video data, Pasunuru and Bansal (2017a) and Sun et al. (2019) propose to improve video captioning with self-supervised learning tasks or unsupervised learning tasks, such as unsupervised video prediction, entailment generation and text-to-video generation. Pasunuru and Bansal (2017a) demonstrate that multi-task training contributes to sharing knowledge across different domains, and each task, including video captioning, benefits from the training of other irrelevant tasks. Sun et al. (2019) take advantage of the abundance of unlabeled videos on YouTube and train the BERT model introduced in Devlin et al. (2018) on comparably large-scale videos, which is then used as a feature extractor for video captioning. A large amount of pre-training data is critical to BERT models both in video captioning and machine translation (Devlin et al., 2018; Sun et al., 2019). By aggregating different experts on different known activities, Wang et al. (2019b) take advantage of external textual corpora and transfer knowledge to unseen data for zero-shot video captioning. A spatio-temporal graph model is built to find object interactions and knowledge distillation mechanism is proposed to increase stability of performance (Pan et al., 2020).

2.3. RNN Training Strategy

The traditional method to train an RNN is the Teacher Forcing algorithm (Williams and Zipser, 1989), which feeds human annotations to the RNN as input at each step to guide the token generation during training and samples a token from the model itself as input during inference. The different sources of input tokens during training and inference lead to the inability of the model to generate high-quality tokens in inference, as errors may accumulate along the sequence generation.

Bengio et al. (2015) propose to switch gradually from guiding generation by true tokens to feeding sampled tokens during training, which helps RNN models adapt to the inference scheme in advance. It has been applied to image captioning and speech recognition. Inspired by Huszar (2015), who mathematically proves that both the Teacher Forcing algorithm and Curriculum Learning have a tendency to learn a biased model, Goyal et al. (2016) solve the problem by adopting an adversarial domain method to align the dynamics of the RNN during training and inference. Zhang et al. (2020) propose an object relational graph (ORG) to encode interaction features and design a teacher-recommended learning (TRL) method to utilize linguistic knowledge.

Inspired by the successful application of RL methods in image captioning (Rennie et al., 2017; Pasunuru and Bansal, 2017b) propose a modified reward that compensates for the logical contradiction in phrase-matching metrics as the direct optimization target in video captioning. The gradient of the non-differentiable RL loss function is computed and back-propagated by the REINFORCEMENT algorithm (Williams, 1992). But calculation of the reward for each training batch adds a non-negligible computation cost to the training process and slows down the optimization progress. In addition, the improvements of RL methods on various metrics are not comparable with the improvement on the specific metric used as RL reward.

3. THE PROPOSED APPROACHES

We consider the video captioning task as a supervised task. The training set is annotated as N pairs of $\{\mathbf{X}_i, \hat{\mathbf{Y}}_i\}$, where $\mathbf{X}_i$ denotes a video and $\hat{\mathbf{Y}}_i$ represents the corresponding target caption. Suppose there are M frames from a video and a caption consisting of L_i words, then we have:

$$\begin{aligned}\mathbf{X}_i &= \{\boldsymbol{x}_{i,0}, \boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,M-1}\},\\ \hat{\mathbf{Y}}_i &= \{\hat{\boldsymbol{y}}_{i,0}, \hat{\boldsymbol{y}}_{i,1}, \ldots, \hat{\boldsymbol{y}}_{i,L_i-1}\},\end{aligned} \tag{1}$$

where each $\boldsymbol{x}$ denotes a single frame and each $\boldsymbol{y}$ denotes a word belonging to a fixed known dictionary.

A pretrained model is used to produce word embeddings, and we obtain a low-dimension embedding of the caption $\hat{\mathbf{Y}}_i \in \mathbb{R}^{L_i \times D_w}$:

$$\hat{\mathbf{Y}}_i = (\boldsymbol{w}_{i,0}, \boldsymbol{w}_{i,1}, \ldots, \boldsymbol{w}_{i,L_i-1})^T, \quad \boldsymbol{w}_{i,j} \in \mathbb{R}^{D_w}, \tag{2}$$

where D_w is the dimension of the word embedding space.

3.1. Encoder-Decoder Paradigm

3.1.1. Encoder

Our encoder is composed of a 3D ConvNet, a 2D ConvNet and a semantic detection network (SDN). The 3D ConvNet is utilized to produce the spatio-temporal feature $\boldsymbol{e}_i \in \mathbb{R}^{D_e}$ for the ith video. The 2D ConvNet is supposed to find the static visual feature $\boldsymbol{r}_i \in \mathbb{R}^{D_r}$ for the ith video. The visual spatio-temporal representation of the ith video can then be obtained by concatenating both features together as follows:

$$\boldsymbol{v}_i = \begin{pmatrix}\boldsymbol{r}_i\\ \boldsymbol{e}_i\end{pmatrix} \in \mathbb{R}^{D_v}, \tag{3}$$

where $D_v = D_e + D_r$.

For semantic detection, we manually select the K most common and meaningful words, which consists of the most frequent nouns, verbs or adjectives, from both the training set and the validation set as candidate tags for all videos (Gan et al., 2017). The semantic detection task is treated as a multi-label classification task with $\boldsymbol{v}_i$ as the representation of the ith video and $\hat{\boldsymbol{s}}_i = \{\hat{s}_{i,0}, \hat{s}_{i,1}, \ldots, \hat{s}_{i,K-1}\} \in \{0,1\}^K$ as the ground truth. If the jth tag exists in the annotations of the ith video, then $\hat{s}_{i,j} = 1$; otherwise, $\hat{s}_{i,j} = 0$. Suppose $\boldsymbol{s}_i$ is the semantic feature of the ith video. Then, we have $\boldsymbol{s}_i = \sigma(f(\boldsymbol{v}_i)) \in (0,1)^K$, where $f(\cdot)$ is a non-linear mapping and $\sigma(\cdot)$ a sigmoid activation function. Mean average precision is applied to evaluate the quality of semantic features. A multi-layer perceptron (MLP) of adequate scale is exploited to learn semantic representations from the samples. The set of input features is determined by the experimental results for each dataset. The SDN is trained by minimizing the loss function:

$$L(\boldsymbol{s}_i, \hat{\boldsymbol{s}}_i) = \frac{1}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{K-1} \hat{s}_{i,j}\log s_{i,j} + (1-\hat{s}_{i,j})\log(1-s_{i,j}). \tag{4}$$

A probability distribution of tags $\boldsymbol{s}_i$ is produced by the SDN to represent the semantic content of the ith video in the training set, the validation set or the test set.

3.1.2. Decoder

Standard RNNs (Elman, 1990) are capable of learning temporal patterns from input sequences. But they suffer from the gradient vanishing/explosion problem, which results in their inability to generalize to long sequences. LSTM (Hochreiter and Schmidhuber, 1997) is a prevailing variant of RNN that alleviates the long-term dependency problem by using gates to update the cell state, but it ignores the semantic information of the input sequence. We use SCN(Semantic Compositional Network) (Gan et al., 2017), a variant of LSTM, as our decoder, because it not only avoids the long-term dependency problem but also takes advantage of semantic information of the input video. Suppose we have a video feature $\boldsymbol{v}$, a semantic feature $\boldsymbol{s}$, an input vector $\boldsymbol{x}_t$ at time step t and a hidden state $\boldsymbol{h}_{t-1}$ at time step $t-1$. The SCN integrates semantic information $\boldsymbol{s}$ into $\boldsymbol{v}$, $\boldsymbol{x}_t$, and $\boldsymbol{h}_{t-1}$, respectively, and obtains the semantics-related video feature $\hat{\boldsymbol{v}}$, the semantics-related input $\hat{\boldsymbol{x}}_t$ and the semantics-related hidden state $\hat{\boldsymbol{h}}_{t-1}$ as follows:

$$\begin{aligned} \hat{\boldsymbol{x}}_{z,t} &= \mathbf{W}_{z,c}\cdot((\mathbf{W}_{z,a}\cdot\boldsymbol{x}_t)\odot(\mathbf{W}_{z,b}\cdot\boldsymbol{s})), \quad z\in\{c,i,f,o\},\\ \hat{\boldsymbol{v}}_z &= \mathbf{C}_{z,c}\cdot((\mathbf{C}_{z,a}\cdot\boldsymbol{v})\odot(\mathbf{C}_{z,b}\cdot\boldsymbol{s})), \quad z\in\{c,i,f,o\},\\ \hat{\boldsymbol{h}}_{z,t-1} &= \mathbf{U}_{z,c}\cdot((\mathbf{U}_{z,a}\cdot\boldsymbol{h}_{t-1})\odot(\mathbf{U}_{z,b}\cdot\boldsymbol{s})), \quad z\in\{c,i,f,o\}, \end{aligned}\tag{5}$$

where c, i, f and o denote the cell state, the input gate, the forget gate and the output gate, respectively.

Then input gate $\boldsymbol{i}_t$, forget gate $\boldsymbol{f}_t$ and output gate $\boldsymbol{o}_t$ at time step t are calculated, respectively, in a way similar to the standard LSTM:

$$\begin{aligned} \boldsymbol{i}_t &= \sigma(\hat{\boldsymbol{x}}_{i,t} + \hat{\boldsymbol{h}}_{i,t-1} + \hat{\boldsymbol{v}}_i + \boldsymbol{b}_i),\\ \boldsymbol{f}_t &= \sigma(\hat{\boldsymbol{x}}_{f,t} + \hat{\boldsymbol{h}}_{f,t-1} + \hat{\boldsymbol{v}}_f + \boldsymbol{b}_f),\\ \boldsymbol{o}_t &= \sigma(\hat{\boldsymbol{x}}_{o,t} + \hat{\boldsymbol{h}}_{o,t-1} + \hat{\boldsymbol{v}}_o + \boldsymbol{b}_o), \end{aligned}\tag{6}$$

where σ denotes the logic sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}} \in (0,1)$ and $\boldsymbol{b}$ is a bias term for each gate.

The raw cell state at the current step t can be computed as follows:

$$\hat{\boldsymbol{c}}_t = \tanh(\hat{\boldsymbol{x}}_{c,t} + \hat{\boldsymbol{h}}_{c,t-1} + \hat{\boldsymbol{v}}_c + \boldsymbol{b}_c), \tag{7}$$

where tanh denotes the hyperbolic function $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \in (-1,1)$ and $\boldsymbol{b}_c$ is the bias term for the cell state. The input gate $\boldsymbol{i}_t$ is supposed to control the throughput of the semantic-related input $\hat{\boldsymbol{x}}_t$, and the forget gate $\boldsymbol{f}_t$ is designed to determine the preservation of the previous cell state $\boldsymbol{c}_{t-1}$. Thus, we have the final cell state $\boldsymbol{c}_t$ at time step:

$$\boldsymbol{c}_t = \boldsymbol{f}_t * \boldsymbol{c}_{t-1} + \boldsymbol{i}_t * \hat{\boldsymbol{c}}_t. \tag{8}$$

The output gate controls the throughput ratio of the cell state $\boldsymbol{c}_t$ so that the cell output $\boldsymbol{h}_t$ can be determined by:

$$\boldsymbol{h}_t = \boldsymbol{o}_t * \tanh(\boldsymbol{c}_t). \tag{9}$$

The semantics-related variables $\hat{\boldsymbol{x}}_t$, $\hat{\boldsymbol{v}}$, $\hat{\boldsymbol{h}}_{t-1}$, and $\hat{\boldsymbol{c}}_t$ are dependent on semantic feature $\boldsymbol{s}$ so that the SCN takes semantic information of the video into account implicitly. The forget gate $\boldsymbol{f}_t$ is a key component in updating $\boldsymbol{c}_{t-1}$ to $\boldsymbol{c}_t$, which, to some degree, avoids the long-term dependency problem. The overview of the SCN unit is showed in **Figure 1**.

3.2. Training Method

In the context of the RNN trained with the Teacher Forcing algorithm, the logarithmic probability $P(Y_i|X_i;\Theta)$ of a given triplet of input/output/label $(X_i, Y_i, \hat{Y}_i)$ and given model parameters Θ can be calculated as:

$$P(Y_i|X_i;\Theta) = \sum_{t=0}^{L_i-1}\log P(y_{i,t}|\hat{y}_{i,0},\cdots,\hat{y}_{i,t-1},X_i;\Theta), \tag{10}$$

where L_i is the length of output.

In the case of SCN, the joint logarithmic probability can be computed as:

$$\begin{aligned} P(Y_i|X_i;\Theta) &= \sum_{t=0}^{L_i-1}\log P(y_{i,t}|\hat{y}_{i,0},\cdots,\hat{y}_{i,t-1},s_i,X_i;\Theta),\\ &= \sum_{t=0}^{L_i-1}\log P(y_{i,t}|h_{i,t-1},c_{i,t-1},\hat{y}_{i,t-1},s_i,X_i;\Theta), \end{aligned}\tag{11}$$

where $h_{i,t}$, $c_{i,t}$, and s_i are the output state, the cell state and the semantic feature of the ith video, respectively.

To some extent, $h_{i,t}$ and $c_{i,t}$ can be viewed as the aggregation of all the previous information. We can compute them using the recurrence relation:

$$\begin{aligned} h_{i,t} &= \begin{cases} f(X_i, h_{i,t-1}, c_{i,t-1}, s_i, X_i;\Theta) & \text{if } t=0,\\ f(\hat{y}_{i,t-1}, h_{i,t-1}, c_{i,t-1}, s_i, X_i;\Theta) & \text{if } t>0, \end{cases}\\ c_{i,t} &= \begin{cases} g(X_i, h_{i,t-1}, c_{i,t-1}, s_i, X_i;\Theta) & \text{if } t=0,\\ g(\hat{y}_{i,t-1}, h_{i,t-1}, c_{i,t-1}, s_i, X_i;\Theta) & \text{if } t>0, \end{cases} \end{aligned}\tag{12}$$

where $h_{i,-1} = \mathbf{0}$, $c_{i,-1} = \mathbf{0}$. In inference, we need to replace $\hat{y}_{i,t}$ with $y_{i,t}$, which may lead to the accumulation of prediction errors.

In order to bridge the gap between training and testing in the Teacher Forcing algorithm, we train our video captioning model with scheduled sampling. Scheduled sampling transfers the training process gradually from using ground truth words

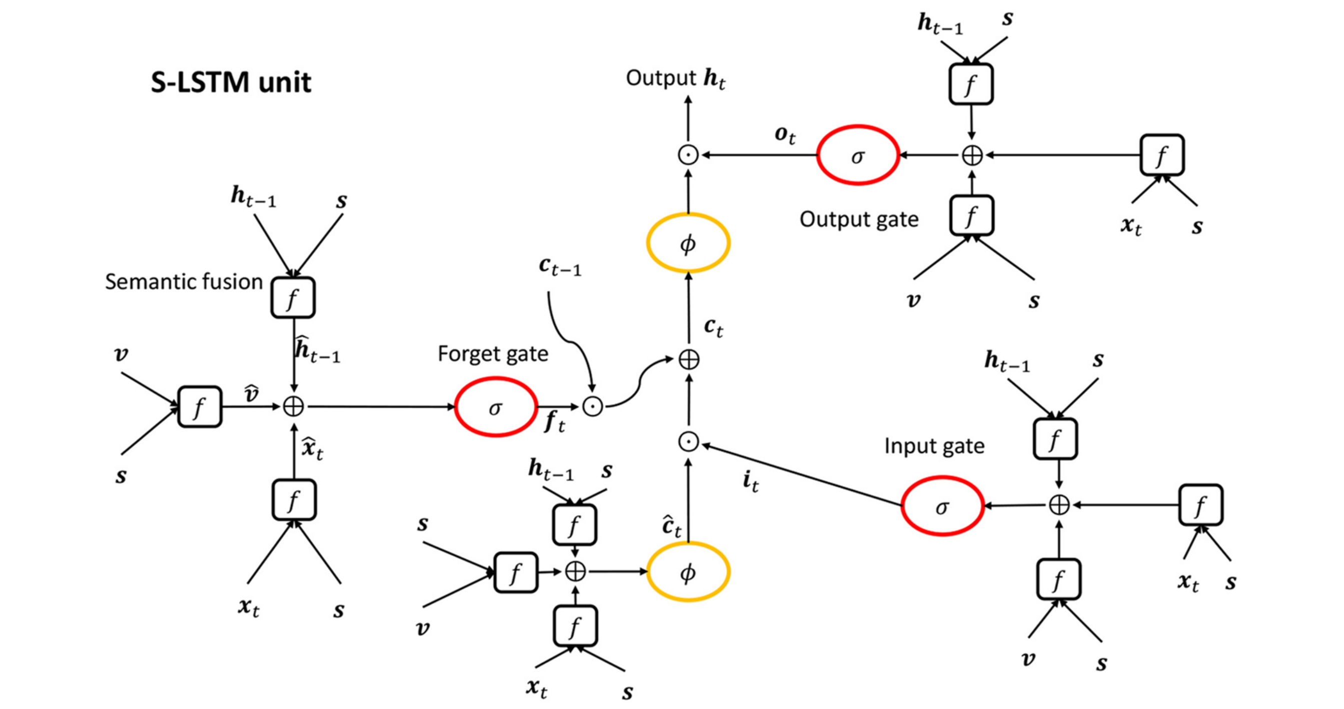


FIGURE 1 | The figure of SCN unit. σ, ϕ, semantic fusion (f) denotes a sigmoid function, a tanh function and Equation (5), respectively. $\odot$ and $\oplus$ in a circle denote element-wise product and element-wise addition, respectively.

$\hat{Y}_i$ for guiding to using sampled words Y_i for guiding at each recurrent step. The commonly used strategy to sample a word from the output distribution is arg max. But the search scope is limited to a relatively small part of the search space, since it always selects the word with the largest probability. For the sake of enlarging the search scope, we draw a word randomly from the output distribution as a part of the input for the next recurrent step. In this way, words with higher probabilities are more likely to be chosen. The randomness of the sampling procedure will enable the recurrent network to explore a relatively large range of the network state space. In addition, the network is less likely to get stuck in a local minimum. In the perspective of training machine learning models, the multinomial sampling strategy reduces overfitting of the network; in other words, it acts like a regularizer.

Our method to optimize the language model consists of two parts: the outer loop schedule the sampling probability at each recurrent step (Algorithm 1), while the algorithm inside the RNN (Algorithm 2) specifies the procedure to sample from the output of a model with a given possibility as a part of the input for the next step of the RNN.

3.3. Sentence-Length-Related Loss Function

What is a good description for a video? A good description should be both accurate and concise. In order to achieve this goal, we design a sentence-length-modulated loss function for our model as follows:

Algorithm 1: Scheduling Algorithm: schedule the ϵ across epochs.

```
Require: EPOCH: max epoch number, STEPS_PER_EPOCH: steps per epoch, feature: necessary features
$\epsilon list \leftarrow generate\_epsilon()$ {Generate epsilon for each epoch by a predeterminate strategy.}
output ← 0
for $i = 0$ to EPOCH do
  for $j = 0$ to STEPS_PER_EPOCH do
    $\mathbf{output}_{i,j} \leftarrow function(\mathbf{feature}_{i,j}, \epsilon list[i])$ {Run RNN}
    optimize the network with an optimizer
    extend output with $\mathbf{output}_{i,j}$
  end for
end for
return output
```

$$\mathbf{Loss}(\hat{y}_i, s_i, X_i; \Theta) = -\sum_{i=0}^{b_s-1} \frac{1}{L_i^{\beta}} \sum_{t=0}^{L_i-1} \log p(\hat{y}_{i,t} | h_{i,t-1}, c_{i,t-1}, s_i, X_i; \Theta), \tag{13}$$

where b_s is the batch size and $\beta >= 0$ is a hyper-parameter that is used to keep a balance between the conciseness and the accuracy of the generated captions. If $\beta = 0$, it is a loss function commonly used in video captioning tasks:

$$\mathbf{Loss}(\hat{y}_i, s_i, X_i; \Theta) = -\sum_{i=0}^{b_s-1} \sum_{t=0}^{L_i-1} \log p(\hat{y}_{i,t} | h_{i,t-1}, c_{i,t-1}, s_i, X_i; \Theta). \tag{14}$$

Algorithm 2: Random Sampling Algorithm: specific procedures in RNN.

```
Require: $\mathbf{v}_i$: video feature, $\mathbf{s}_i$: semantic feature, $\mathbf{x}_i$: input array, $\epsilon$: sampling probability, $STEP$: max time step
Ensure: $\mathbf{h}_i$: output state, $\mathbf{c}_i$: cell state
$\mathbf{h}_{i,0} \leftarrow \mathbf{0}$
$\mathbf{c}_{i,0} \leftarrow \mathbf{0}$
$\mathbf{h}_i \leftarrow \mathbf{0}$
$\mathbf{c}_i \leftarrow \mathbf{0}$
$\mathbf{embed} \leftarrow \mathbf{x}_{i,0}$
for $t = 1$ to $STEP$ do
    $\mathbf{h}_{i,t}, \mathbf{c}_{i,t} \leftarrow recurrent\_step(\mathbf{h}_{i,t-1}, \mathbf{c}_{i,t-1}, \mathbf{v}_i, \mathbf{s}_i, \mathbf{embed})$
    extend $\mathbf{h}_i$ with $\mathbf{h}_{i,t}$
    extend $\mathbf{c}_i$ with $\mathbf{c}_{i,t}$
    $prob \leftarrow random(0, 1)$
    if $prob < \epsilon$ then
        $\mathbf{prob\_dist}_{i,t} \leftarrow word\_dist\_map(\mathbf{h}_{i,t})$ {Map output state to word probability.}
        $\mathbf{word\_index} \leftarrow multinomial(\mathbf{prob\_dist}_{i,t})$ {Sample from the word distribution.}
        $\mathbf{embed} \leftarrow lookup\_embed(\mathbf{word\_index})$ {Use an embedding vector to represent the word.}
    else
        $\mathbf{embed} \leftarrow \mathbf{x}_{i,t}$
    end if
    $t \leftarrow t + 1$
end for
return $\mathbf{h}_i, \mathbf{c}_i$
```

In this loss function, a long sentence has greater loss than a short sentence. Thus, after minimizing the loss, the RNN is inclined to generate relatively short annotations that may be incomplete in semantics or sentence structure. If $\beta = 1$, all words in the generated captions are treated equally in the loss function as well as in the process of optimization, which may lead to redundancy or duplicate words in the process of generating captions.

Thus, we have the following optimization problem:

$$\Theta = \arg\min_{\Theta} - \sum_{i=0}^{N-1} \frac{1}{L_i^{\beta}} \sum_{t=0}^{L_i-1} \log p(\hat{y}_{i,t} | h_{i,t-1}, c_{i,t-1}, s_i, X_i; \Theta), \tag{15}$$

where N is the size of the training data and Θ is the parameter of our model.

GNMT, Google's Neural Machine Translation system, employs a similar length-normalization technique in the beam search during test, but not during training (Wu et al., 2016). In contrast, our model abandons beam search in the decoder, and the model parameters are optimized by the sentence-length-modulated loss function (13). Note that beam search makes the decoding process slower.

The overall structure of our model is visualized in **Figure 2**. Our SDN and visual feature extractors in the encoder component share the same 2D ConvNet and 3D ConvNet in practice.

4. EXPERIMENTS

We evaluate our model on two popular video captioning datasets to show the performance of our approach. We compare our results to other existing methods.

4.1. Datasets

4.1.1. YouTube2Text

The YouTube2Text or MSVD (Chen and Dolan, 2011; Guadarrama et al., 2013) dataset, published in 2013, contains 1970 short YouTube video clips. The average length of them is about 10 seconds. We get roughly 40 descriptions for each video. We follow the dataset split setting used in prior studies (Pan et al., 2016; Yu et al., 2016; Gan et al., 2017), in which the training dataset contains 1200 clips, the validation dataset contains 100 clips, and the rest of them belong to the test dataset. We tokenize the captions from the training and validation datasets and obtain approximately 14,000 unique words. Twelve thousand five hundred and ninety-two of them are utilized for prediction, and the remaining words are replaced by $< unk >$. We add the token $< eos >$ to signal the end of a sentence.

4.1.2. MSR-VTT

MSR-Video to Text (MSR-VTT) (Pan et al., 2016; Xu et al., 2016) is a large-scale video benchmark, first presented in 2016. In its first version, MSR-VTT provided 10k short video segments with 200k descriptions in total. Each video segment was described by about 20 independent English sentences. In its second version, which was published in 2017, MSR-VTT provides additional 3k short clips as a testing set, and video clips in the first version can be used as training and validation sets. Because of lacking human annotations for the test set in the second version, we perform experiments on the first version. We tokenize and obtain 14,071 unique words that appear in the training set and validation set of MSR-VTT 1.0 more than once. Thirteen thousand seven hundred and ninety-four of them are indexed with integer numbers starting at 0, and the rest are substituted by $< unk >$. $< eos >$, which signifies the end of a sentence, is added to the vocabulary of MSR-VTT.

4.2. Overall Score

Based on the widely used BLEU, METEOR, ROUGE-L, and CIDEr metrics, we propose an overall score to evaluate the performance of a language model:

$$\mathbf{S}_{overall} = \frac{\text{B-4}}{top1(\text{B-4})} + \frac{\text{C}}{top1(\text{C})} + \frac{\text{M}}{top1(\text{M})} + \frac{\text{R}}{top1(\text{R})} \in [0, 1], \tag{16}$$

where B-4 denotes BLEU-4, C denotes CIDEr, M denotes METEOR, R represents ROUGE-L and $top1(\cdot)$ denotes the best numeric value of the specific metric. We presume that BLEU-4, CIDEr, METEOR, and ROUGE-L reflect one particular aspect of the performance of a model respectively. First, we normalize each metric value of a model, and then we take the mean value of them as an overall measurement for that model (16). If the result of a model on each metric is closer to the best result of all models, the overall score will be close to 1. If and only if a model has the state-of-the-art performance on all metrics, the overall score is 1.

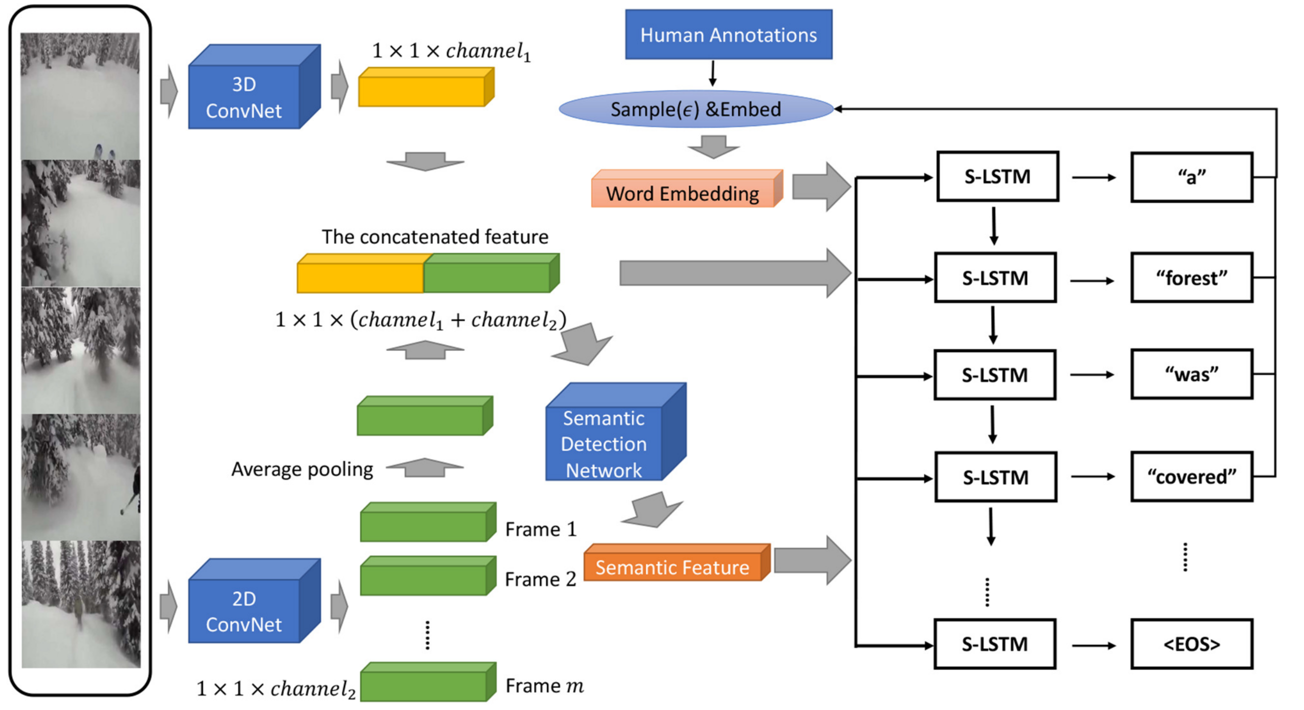


FIGURE 2 | Overall framework of our model. A 3D ConvNet, a 2D ConvNet and a Semantic Detection Net (SDN) constitute the encoder component of our model. S-LSTM stands for a semantics-assisted variant of LSTM which takes a semantic feature, a visual feature and a word embedding as inputs at each step. The word fed as the input to the decoder is sampled from human annotations or the model itself randomly, and then is embeded with the pretrained weights.

If a model is much lower than the state-of-the-art result on each metric, the overall score of the model will be close to 0.

4.3. Training Details

Our visual feature consists of two parts: a static visual feature and a dynamic visual feature. ResNeXt (Xie et al., 2017), which is pretrained on the ImageNet ILSVRC2012 dataset, is utilized as the static visual feature extractor in the encoder of our model. The ECO (Zolfaghari et al., 2018), which is pretrained on the Kinetics-400 dataset, is utilized as the dynamic visual feature extractor for the encoder in our model. More specifically, 32 frames are extracted from each video clip evenly. For each video, we feed 32 frames as input to ResNeXt, take the conv5/block3 output, and apply average pooling to these outputs along the time axis. The newly obtained 2048-dim feature vector is taken as the 2D representation of that video. What's more, we take the 1536-way feature of the global pool in ECO as the 3D representation of each video. Global Vectors for Word Representations (GloVe) (Pennington et al., 2014) is used as the pretrained word embedding model in our experiments. And it is fixed during our training processes.

We set the initial learning rate to 2×10^{-4} for the YouTube2Text dataset and 4×10^{-4} for the MSR-VTT dataset. In addition, we drop the learning rate by 0.316 every 20,350 steps for the MSR-VTT dataset. Batch size is set to 64, and the Adam algorithm is applied to optimize the model for both datasets. The hyper-parameter β_1 is set to 0.9, β_2 is set to 0.999, and ϵ is set to 1×10^{-8} for the Adam algorithm. Each model is trained for 50 epochs, in which the hyper parameter sample probability ϵ is set as $ep \times 0.008$ for the epth epoch. We fine-tune the hyper-parameters of our model on the validation sets and select the best checkpoint for testing according to the overall score of the evaluation on the validation set.

4.4. Comparison With Existing Models

Empirically, we evaluate our method on the YouTube2Text/MSVD (Guadarrama et al., 2013) and MSR-VTT (Xu et al., 2016) datasets. We report the results of our model along with a number of existing models in **Tables 1**, **2**.

4.4.1. Comparison on the YouTube2Text Dataset

Table 1 displays the performance of several models on YouTube2Text. We compare our model with existing methods, including LSTM-E (Pan et al., 2016), h-RNN (Yu et al., 2016), aLSTMs (Gao et al., 2017), SCN (Gan et al., 2017), MTVC (Pasunuru and Bansal, 2017a), ECO (Zolfaghari et al., 2018), SibNet (Liu et al., 2018), POS (Wang et al., 2019a), MARN (Pei et al., 2019), JSRL-VCT (Hou et al., 2019), GRU-EVE (Aafaq et al., 2019), STG-KD (Pan et al., 2020), SAAT (Zheng et al., 2020), and ORG-TRL (Zhang et al., 2020). Our method outperforms all the other methods on all the metrics by a large margin. Note that many of them were published after our initial submission of the present work in the end of May in 2019. Specifically, compared with ORG-TRL (Zhang et al., 2020), the previous state-of-the-art model on this dataset, BLEU-4, CIDEr, METEOR, and ROUGE-L are improved relatively by 14.9, 15.2,

TABLE 1 | Result comparison with existing models on the YouTube2Text dataset.

Model	B-4	C	M	R	Overall (16)
LSTM-E (V+C3D) (Pan et al., 2016)	45.3		31.0		
h-RNN (V+C3D) (Yu et al., 2016)	49.9	65.8	32.6		
aLSTMs (I-3) (Gao et al., 2017)	50.8	74.8	33.3		
SCN (R-152+C3D) (Gan et al., 2017)	51.1	77.7	33.5		
MTVC (I-4) (Pasunuru and Bansal, 2017a)	54.5	92.4	36.0	72.8	0.8961
ECO (R-152+E) (Zolfaghari et al., 2018)	53.5	85.8	35.0		
SibNet (I-1) (Liu et al., 2018)	54.2	88.2	34.8	71.7	0.8740
POS (IR+I3D) (Wang et al., 2019a)	53.9	91.0	34.9	72.1	0.8811
MARN (R-101+R3D) (Pei et al., 2019)	48.6	92.2	35.1	71.9	0.8633
JSRL-VCT (IR+C3D) (Hou et al., 2019)	52.8	87.8	36.1	71.8	0.8762
GRU-EVE (IR+C3D) (Aafaq et al., 2019)	47.9	78.1	35.0	71.5	0.8264
STG-KD (R-101+I3D) (Pan et al., 2020)	52.2	93.0	36.9	73.9	0.8975
SAAT (IR+C3D) (Zheng et al., 2020)	46.5	81.0	33.5	69.4	0.8110
ORG-TRL (IR+C3D) (Zhang et al., 2020)	54.3	95.2	36.4	73.9	0.9078
Our model	**62.4**	**109.7**	**39.0**	**77.0**	**1.0000**

V, C3D, I-n, R-n, E, IR, I3D and R3D denote VGG19, C3D, n-version Inception, n-layer ResNet, ECO, Inception-ResNet-v2, I3D and 3D-ResNeXt features, respectively. The boldness denotes the best value in the corresponding column.

TABLE 2 | Result comparison with existing models on the MSR-VTT dataset.

Model	B-4	C	M	R	Overall
MTVC (I-4) (Pasunuru and Bansal, 2017a)	40.8	47.1	28.8	60.2	0.9223
CIDEnt-RL (I-4) (Pasunuru and Bansal, 2017b)	40.5	51.7	28.4	61.4	0.9435
SibNet (I-3) (Liu et al., 2018)	40.9	47.5	27.5	60.2	0.9137
HACA (R-152+A) (Wang et al., 2018)	43.4	49.7	29.5	61.8	0.9608
TAMoE (I3D) (Wang et al., 2019b)	42.2	48.9	29.4	62.0	0.9505
POS (IR+I3D) (Wang et al., 2019a)	41.3	**53.4**	28.7	62.1	0.9611
MARN (R-101+R3D) (Pei et al., 2019)	40.4	47.1	28.1	60.7	0.9162
JSRL-VCT (IR+C3D) (Hou et al., 2019)	42.3	49.1	**29.7**	62.8	0.9576
GRU-EVE (IR+C3D) (Aafaq et al., 2019)	38.3	48.1	28.4	60.7	0.9119
STG-KD (R-101+I3D) (Pan et al., 2020)	40.5	47.1	28.3	60.9	0.9192
SAAT (IR+C3D+Ca) (Zheng et al., 2020)	39.9	51.0	27.7	61.2	0.9303
ORG-TRL (IR+C3D) (Zhang et al., 2020)	43.6	50.9	28.8	62.1	0.9628
Our model	**45.8**	53.2	29.3	**63.6**	**0.9957**

A and Ca denote audio and category features, respectively. The boldness denotes the best value in the corresponding column.

7.1, and 4.2%, respectively. Our model has the highest overall score as defined in (16).

4.4.2. Comparison on the MSR-VTT Dataset

Table 2 displays the evaluation results of several video captioning models on the MSR-VTT. In this table, we compare our model with existing models, including MTVC (Pasunuru and Bansal, 2017a), CIDEnt-RL (Pasunuru and Bansal, 2017b), SibNet (Liu et al., 2018), HACA (Wang et al., 2018), TAMoE (Wang et al., 2019b), POS (Wang et al., 2019a), MARN (Pei et al., 2019), JSRL-VCT (Hou et al., 2019), GRU-EVE (Aafaq et al., 2019), STG-KD (Pan et al., 2020), SAAT (Zheng et al., 2020), ORG-TRL (Zhang et al., 2020). According to the overall score defined in (16), ORG-TRL is the best among existing models. Our model achieves higher values on all metrics than this model. Two models POS and JSRL-VCT achieve slightly higher CIDEr value and METEOR values than our model, respectively, but their other metric values are clearly lower than our results.

Our model achieves better results on both the YouTube2Text dataset and the MSR-VTT dataset. Note that our model is only trained on a single dataset without an attention mechanism, and it is tested without ensemble or beam search.

5. MODEL ANALYSIS

In this section, we discuss the utility of the three improvements on our model.

5.1. Analysis on Semantic Features

Semantic features are the output of a multi-label classification task. Mean average precision (mAP) is often used to evaluate the results of multi-label classification tasks (Tsoumakas and Katakis, 2007). Here, we apply it to evaluate the quality of semantic features.

5.1.1. Semantic Features Predicted With Different Sets of Input Features

Figures 3, **4** demonstrate the quality of semantic features, using different sets of feature maps as inputs, with respect to the training epochs. **Figure 3** shows that, on the YouTube2Text dataset, the mAP values are proportional to training epochs. With the same number of training epochs, the qualities of semantic features are in the order: ECO-ResNeXt > ResNeXt > ECO, where ECO-ResNeXt, ResNeXt, and ECO denote the models trained with visual features from ECO-ResNeXt, ResNeXt, or ECO, respectively. **Figure 4** demonstrates that, on the MSR-VTT dataset, both mAP values of semantic information decline after the models are trained for more than 800 epochs with ResNeXt feature maps or ECO-ResNeXt feature maps as inputs. With ECO feature maps as inputs, the performance of the semantic detection model is still proportional to the training epochs.

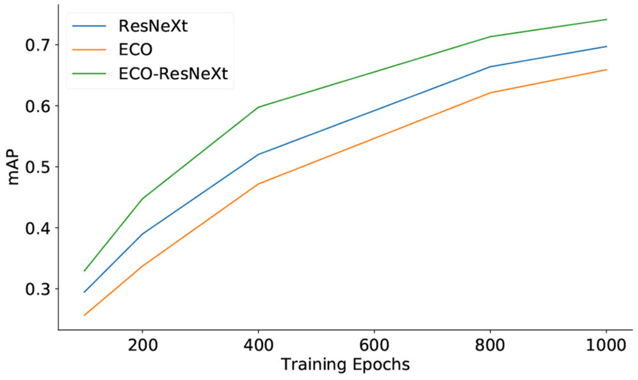


FIGURE 3 | The quality of semantic features predicted with different sets of input features evaluated by mAP on the YouTube2Text. "ResNeXt," "ECO," and "ECO-ResNeXt" denote that the semantic models are trained and the semantic features are predicted with visual features produced by ResNeXt, ECO, both ECO and ResNeXt, respectively.

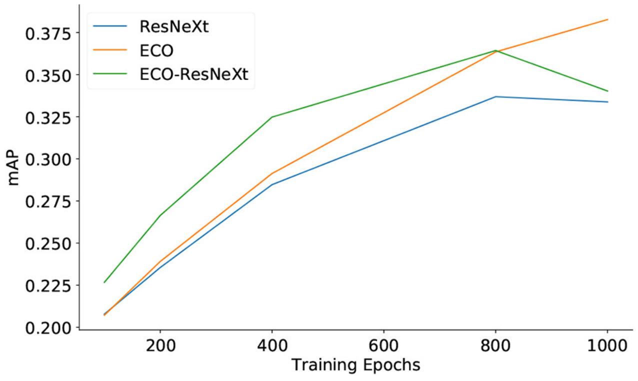


FIGURE 4 | The quality of semantic features predicted with different sets of input features evaluated by mAP on the MSR-VTT dataset.

TABLE 3 | Results of scheduled sampling methods (multinomial sampling) on the YouTube2Text dataset with different sets of semantic features.

Semantic features (mAP)	B-4	C	M	R	Overall
0.3295	53.9	90.5	35.8	73.4	0.8896
0.5977	60.5	102.7	38.0	75.9	0.9663
0.7414	**62.4**	**109.7**	**39.0**	**77.0**	**1.0000**

A larger mAP implies a better representation of semantic meanings. The boldness denotes the best value in the corresponding column.

TABLE 4 | Results of scheduled sampling methods (multinomial sampling) on MSR-VTT data with different sets of semantic features.

Semantic feature (mAP)	B-4	C	M	R	Overall
0.2072	40.5	46.8	27.2	62.7	0.9292
0.2913	44.0	50.7	**28.9**	62.6	0.9878
0.3827	**44.9**	**51.8**	28.8	**63.12**	**0.9996**

The boldness denotes the best value in the corresponding column.

TABLE 5 | Results of different training strategies on YouTube2Text data with the best semantic features.

Training method	B-4	C	M	R	Overall
Teacher Forcing	61.93	108.56	38.96	76.75	0.9942
arg max	62.16	109.31	38.98	76.81	0.9972
Multinomial	**62.35**	**109.71**	**39.04**	**77.04**	**1.0000**

The boldness denotes the best value in the corresponding column.

TABLE 6 | Results of different training strategies on MSR-VTT data with the best semantic features.

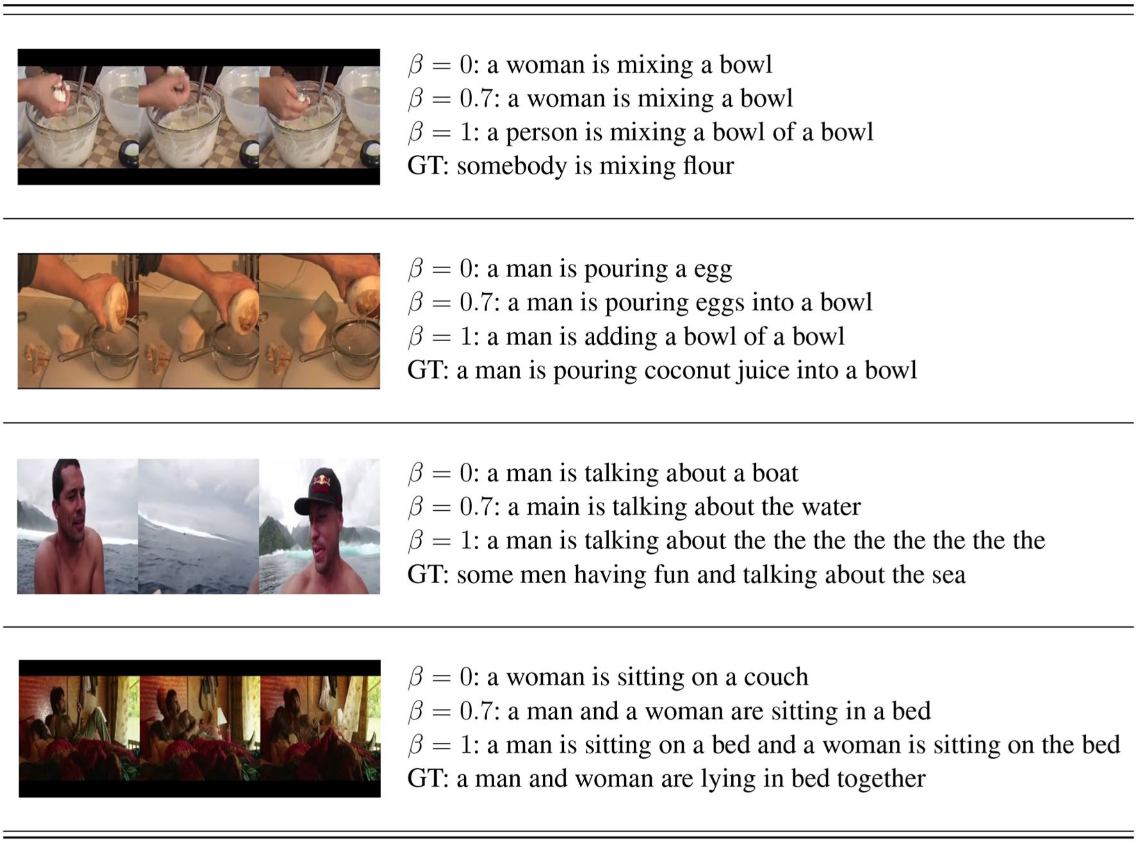
Training method	B-4	C	M	R	Overall
Teacher Forcing	45.05	50.25	29.12	62.72	0.9771
arg max	**45.83**	**53.16**	**29.28**	**63.64**	**1.0000**
Multinomial	44.94	51.77	28.82	63.12	0.9826

The boldness denotes the best value in the corresponding column.

TABLE 7 | Average length of the captions in the test set.

Model	$\beta = 0$	$\beta = 0.7$	$\beta = 1$	Ground truth
mLen1	5.12	5.18	5.80	7.01
mLen2	6.27	6.69	6.99	9.32

mLen1 stands for the mean length of YouTube2Text, and mLen2 stands for the mean length of MSR-VTT. Ground Truth denotes the human annotations for the test set.



$\beta = 0$: a woman is mixing a bowl
$\beta = 0.7$: a woman is mixing a bowl
$\beta = 1$: a person is mixing a bowl of a bowl
GT: somebody is mixing flour

$\beta = 0$: a man is pouring a egg
$\beta = 0.7$: a man is pouring eggs into a bowl
$\beta = 1$: a man is adding a bowl of a bowl
GT: a man is pouring coconut juice into a bowl

$\beta = 0$: a man is talking about a boat
$\beta = 0.7$: a main is talking about the water
$\beta = 1$: a man is talking about the the the the the the the the
GT: some men having fun and talking about the sea

$\beta = 0$: a woman is sitting on a couch
$\beta = 0.7$: a man and a woman are sitting in a bed
$\beta = 1$: a man is sitting on a bed and a woman is sitting on the bed
GT: a man and woman are lying in bed together

FIGURE 5 | Examples of machine-generated captions and human annotations (GT).

5.1.2. Models Trained With Different Semantic Features

Tables 3, **4** list the performance of our model trained by scheduled multinomial sampling with different semantic features on the YouTube2Text and MSR-VTT datasets, respectively. The results clearly show that a better multi-label classification enables a better video captioning model. Semantic features with higher mAP provide more appropriate potential attributes of a video for the model. Thus, the model is able to generate better video annotations by comprehensively considering semantic features, spatio-temporal features, and contextual information.

5.2. Analysis on the Scheduled Sampling

Tables 5, **6** show the comparison among the Teacher Forcing algorithm, scheduled sampling with the arg max strategy and scheduled sampling with the multinomial strategy on YouTube2Text and MSR-VTT datasets, respectively. Teacher Forcing utilizes human annotations to guide the generation of words during training and samples from the word distribution of the output of the model to direct the generation during inference. The arg max strategy switches gradually from the Teacher Forcing way to sample words with the largest possibility from the model itself during training. The Multinomial strategy is similar to the arg max strategy but samples words randomly from the distribution of the model at each step. As we can infer from **Tables 3**, **4**, the scheduled sampling with the multinomial strategy yields a better performance than the other two methods on the YouTube2Text dataset and the one with the arg max strategy yields the best performance on the MSR-VTT dataset. Our method explores a larger range of RNN state space and thus is likely to find a better solution during training.

5.3. Analysis on the Length Normalization of the Loss Function

As demonstrated in **Table 7**, the average length of human annotations is larger than those generated by models with $\beta = \{0, 0.7, 1\}$ (13), respectively. But **Figure 5** displays the tendency of redundancy in captions generated by the $\beta = 1$ model, which deteriorates the overall quality of model-generated sentences. The average caption length of the model with $\beta = 0.7$ is greater than that of the model with $\beta = 0$, whereas it is smaller than that from the model with $\beta = 1$. The model with $\beta = 0.7$ generates relatively long annotations for videos without suffering from redundancy or duplication of words, and we therefore consider it the optimal choice.

6. CONCLUSION

We suggest three improvements for solving the video captioning task. First, mAP is applied to evaluate the quality of semantic information, and a SDN with adequate computation complexity and input features is used to extract high-quality semantic features from videos, which contributes to the success of our semantics-assisted model. Second, we employ a scheduled sampling training strategy. Third, a sentence-length-modulated loss function is proposed to keep the model in a balance between language redundancy and conciseness. Our method achieves results that are superior to the state-of-the-art on the YouTube2Text dataset. The performance of our model is comparable to the state-of-the-art on the MSR-VTT dataset. In the future, we may obtain further improvements in video captioning by integrating spatio-temporal attention mechanisms with visual-semantics features.

DATA AVAILABILITY STATEMENT

The YouTube2Text/MSVD dataset could be obtained from http://www.cs.utexas.edu/users/ml/clamp/videoDescription/ and the MSR-VTT dataset could be obtained from http://ms-multimedia-challenge.com/2016/dataset.

ETHICS STATEMENT

Written informed consent was obtained from the individual(s) for the publication of any potentially identifiable images or data included in this article.

AUTHOR CONTRIBUTIONS

HC designed and performed the experiments. HC, JL and XH analyzed the experimental results and wrote the article. KL and AM analyzed the data and polished the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700904, in part by the National Natural Science Foundation of China under Grant 61621136008, in part by the German Research Council (DFG) under Grant TRR-169, and in part by Samsung under contract NO. 20183000089.

ACKNOWLEDGMENTS

The authors thank Han Liu, Hallbjorn Thor Gudmunsson, and Jing Wen for valuable and insightful discussions.

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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA). doi: 10.1109/CVPR.2019.01277

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., et al. (2018). "Bottom-up and top-down attention for image captioning and visual question answering," in *IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT), 6077–6086. doi: 10.1109/CVPR.2018.00636

Ba, J., Mnih, V., and Kavukcuoglu, K. (2015). "Multiple object recognition with visual attention," in *3rd International Conference on Learning Representations* (San Diego, CA).

Bahdanau, D., Cho, K., and Bengio, Y. (2015). "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations* (San Diego, CA).

Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. (2015). "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems* (Montreal, QC), 1171–1179.

Chen, D. L., and Dolan, W. B. (2011). "Collecting highly parallel data for paraphrase evaluation," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-2011)* (Portland, OR).

Chen, Y., Wang, S., Zhang, W., and Huang, Q. (2018). "Less is more: picking informative frames for video captioning," in *Computer Vision - 15th European Conference* (Munich), 367–384. doi: 10.1007/978-3-030-01261-8_22

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in EMNLP, (Doha), 1724–1734. doi: 10.3115/v1/D14-1179

Cornia, M., Stefanini, M., Baraldi, L., and Cucchiara, R. (2019). Meshed-memory transformer for image captioning. *arXiv preprint arXiv:1912.08226*. doi: 10.1109/CVPR42600.2020.01059

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR abs/1810.04805*. doi: 10.18653/v1/n19-1423

Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Darrell, T., et al. (2015). "Long-term recurrent convolutional networks for visual recognition and description," in *IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA), 2625–2634. doi: 10.1109/CVPR.2015.7298878

Elman, J. L. (1990). Finding structure in time. *Cogn. Sci.* 14, 179–211. doi: 10.1207/s15516709cog1402_1

Gan, Z., Gan, C., He, X., Pu, Y., Tran, K., Gao, J., et al. (2017). "Semantic compositional networks for visual captioning," in *IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI), 1141–1150. doi: 10.1109/CVPR.2017.127

Gao, L., Guo, Z., Zhang, H., Xu, X., and Shen, H. T. (2017). Video captioning with attention-based LSTM and semantic consistency. *IEEE Trans. Multimedia* 19, 2045–2055. doi: 10.1109/TMM.2017.2729019

Goyal, A., Lamb, A., Zhang, Y., Zhang, S., Courville, A. C., and Bengio, Y. (2016). "Professor forcing: a new algorithm for training recurrent networks," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems* (Barcelona), 4601–4609.

Guadarrama, S., Krishnamoorthy, N., Malkarnenkar, G., Venugopalan, S., Mooney, R. J., Darrell, T., et al. (2013). "Youtube2text: recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition," in *IEEE International Conference on Computer Vision, ICCV* (Sydney, NSW), 2712–2719. doi: 10.1109/ICCV.2013.337

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Hou, J., Wu, X., Zhao, W., Luo, J., and Jia, Y. (2019). "Joint syntax representation learning and visual cue translation for video captioning," in *The IEEE International Conference on Computer Vision (ICCV)* (Seoul). doi: 10.1109/ICCV.2019.00901

Huszar, F. (2015). How (not) to train your generative model: scheduled sampling, likelihood, adversary? *CoRR abs/1511.05101*.

Liu, S., Ren, Z., and Yuan, J. (2018). "SIBNet: sibling convolutional encoder for video captioning," in *ACM Multimedia Conference on Multimedia Conference* (Seoul), 1425–1434. doi: 10.1145/3240508.3240667

Mao, J., Xu, W., Yang, Y., Wang, J., and Yuille, A. L. (2014). Explain images with multimodal recurrent neural networks. *CoRR abs/1410.1090*.

Pan, B., Cai, H., Huang, D.-A., Lee, K.-H., Gaidon, A., Adeli, E., et al. (2020). "Spatio-temporal graph for video captioning with knowledge distillation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Virtual Conference). doi: 10.1109/CVPR42600.2020.01088

Pan, Y., Mei, T., Yao, T., Li, H., and Rui, Y. (2016). "Jointly modeling embedding and translation to bridge video and language," in *IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV), 4594–4602. doi: 10.1109/CVPR.2016.497

Pasunuru, R., and Bansal, M. (2017a). "Multi-task video captioning with video and entailment generation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (Vancouver, BC), 1273-1283. doi: 10.18653/v1/P17-1117

Pasunuru, R., and Bansal, M. (2017b). "Reinforced video captioning with entailment rewards," in *EMNLP* (Copenhagen), 979–985. doi: 10.18653/v1/D17-1103

Pei, W., Zhang, J., Wang, X., Ke, L., Shen, X., and Tai, Y.-W. (2019). "Memory-attended recurrent network for video captioning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA). doi: 10.1109/CVPR.2019.00854

Pennington, J., Socher, R., and Manning, C. D. (2014). "Glove: global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)* (Doha: Association for Computational Linguistics (ACL)), 1532–1543. doi: 10.3115/v1/D14-1162

Ramanishka, V., Das, A., Zhang, J., and Saenko, K. (2017). "Top-down visual saliency guided by captions," in *IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI), 3135–3144. doi: 10.1109/CVPR.2017.334

Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. (2017). "Self-critical sequence training for image captioning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* (Honolulu, HI), 1179–1195. doi: 10.1109/CVPR.2017.131

Sun, C., Myers, A., Vondrick, C., Murphy, K., and Schmid, C. (2019). "Videobert: a joint model for video and language representation learning," in *IEEE/CVF International Conference on Computer Vision* (Seoul: IEEE), 7463–7472. doi: 10.1109/ICCV.2019.00756

Tsoumakas, G., and Katakis, I. (2007). Multi-label classification: an overview. *Int. J. Data Warehous. Mining* 3, 1–13. doi: 10.4018/jdwm.2007070101

Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R. J., Darrell, T., and Saenko, K. (2015). "Sequence to sequence - video to text," in *ICCV* (Santiago), 4534–4542. doi: 10.1109/ICCV.2015.515

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). "Show and tell: a neural image caption generator," in *IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA), 3156–3164. doi: 10.1109/CVPR.2015.7298935

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 *The IEEE International Conference on Computer Vision (ICCV)* (Seoul). doi: 10.1109/ICCV.2019.00273

Wang, X., Wang, Y., and Wang, W. Y. (2018). "Watch, listen, and describe: globally and locally aligned cross-modal attentions for video captioning," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT* (New Orleans, LA), 795–801. doi: 10.18653/v1/N18-2125

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 *The Thirty-Third AAAI Conference on Artificial Intelligence, The Thirty-First Innovative Applications of Artificial Intelligence Conference, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence* (Honolulu, HI: AAAI Press), 8965–8972. doi: 10.1609/aaai.v33i01.33018965

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn*. 8, 229–256. doi: 10.1007/BF00992696

Williams, R. J., and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Comput*. 1, 270–280. doi: 10.1162/neco.1989.1.2.270

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., et al. (2016). Google's neural machine translation system: bridging the gap between human and machine translation. Available online at: https://arxiv.org/abs/1609.08144

Xie, S., Girshick, R. B., Dollár, P., Tu, Z., and He, K. (2017). "Aggregated residual transformations for deep neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI), 5987–5995. doi: 10.1109/CVPR.2017.634

Xu, J., Mei, T., Yao, T., and Rui, Y. (2016). "MSR-VTT: a large video description dataset for bridging video and language," in *IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV), 5288–5296. doi: 10.1109/CVPR.2016.571

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., et al. (2015). "Show, attend and tell: neural image caption generation with visual attention," in *Proceedings of the 32nd International Conference on Machine Learning* (Lille), 2048–2057.

Yao, T., Pan, Y., Li, Y., and Mei, T. (2018). "Exploring visual relationship for image captioning," in *Computer Vision - 15th European Conference* (Munich), 711–727. doi: 10.1007/978-3-030-01264-9_42

You, Q., Jin, H., Wang, Z., Fang, C., and Luo, J. (2016). "Image captioning with semantic attention," in *IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV), 4651–4659. doi: 10.1109/CVPR.2016.503

Yu, H., Wang, J., Huang, Z., Yang, Y., and Xu, W. (2016). "Video paragraph captioning using hierarchical recurrent neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV), 4584–4593. doi: 10.1109/CVPR.2016.496

Yu, J., Li, J., Yu, Z., and Huang, Q. (2019). Multimodal transformer with multi-view visual representation for image captioning. *IEEE Trans. Circ. Syst. Video Technol*. doi: 10.1109/TCSVT.2019.2947482. Available online at: https://ieeexplore.ieee.org/document/8869845/

Zhang, Z., Shi, Y., Yuan, C., Li, B., Wang, P., Hu, W., et al. (2020). "Object relational graph with teacher-recommended learning for video captioning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Virtual Conference). doi: 10.1109/CVPR42600.2020.01329

Zheng, Q., Wang, C., and Tao, D. (2020). "Syntax-aware action targeting for video captioning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Virtual Conference). doi: 10.1109/CVPR42600.2020.01311

Zolfaghari, M., Singh, K., and Brox, T. (2018). "ECO: efficient convolutional network for online video understanding," in *Computer Vision - 15th European Conference* (Munich), 713–730. doi: 10.1007/978-3-030-01216-8_43

Conflict of Interest: KL is employed by Samsung Research China, Beijing (SRC-B), China. The authors declare that this study received funding from Samsung Research China, Beijing (SRC-B). The funder had the following involvement in the study: study design, data analysis and writing of the article.