# SibNet: Sibling Convolutional Encoder for Video Captioning

Sheng Liu, Zhou Ren, and Junsong Yuan

**Abstract**—Visual captioning, the task of describing an image or a video using one or few sentences, is a challenging task owing to the complexity of understanding the copious visual information and describing it using natural language. Motivated by the success of applying neural networks for machine translation, previous work applies sequence to sequence learning to translate videos into sentences. In this work, different from previous work that encodes visual information using a single flow, we introduce a novel Sibling Convolutional Encoder (SibNet) for visual captioning, which employs a dual-branch architecture to collaboratively encode videos. The first content branch encodes visual content information of the video with an autoencoder, capturing the visual appearance information of the video as other networks often do. While the second semantic branch encodes semantic information of the video via visual-semantic joint embedding, which brings complementary representation by considering the semantics when extracting features from videos. Then both branches are effectively combined with soft-attention mechanism and finally fed into a RNN decoder to generate captions. With our SibNet explicitly capturing both content and semantic information, the proposed model can better represent the rich information in videos. To validate the advantages of the proposed model, we conduct experiments on two benchmarks for video captioning, YouTube2Text and MSR-VTT. Our results demonstrates that the proposed SibNet consistently outperforms existing methods across different evaluation metrics.

**Index Terms**—SibNet, video captioning, autoencoder, visual-semantic joint embedding, convolutional encoder.

## 1 Introduction

Visual captioning [1]–[10], the task of automatically describing visual contents, *i.e.*, the contents of images or videos, with natural language, has drawn increasingly more attention from computer vision researchers, owing to its wide range of applications including human computer interaction, video retrieval and video surveillance. With the rapid growth of the amount of visual data, it is critical to endow machines with the ability to caption images and videos, *i.e.*, the ability to "translate" copious visual data into concise summarization in the form of natural language, because it could expedite indexing, querying and searching in large visual data corpus.

The recent success of encoder-decoder pipeline in neural machine translation [11], [12] has motivated researchers to adopt such a pipeline for the task of visual captioning. More specifically, they employ an encoder, *e.g.*, a convolutional neural networks (CNN) for image captioning or a recurrent neural network (RNN) for video captioning, to compress the original visual contents into a vectorial representation, and then employ a decoder, *i.e.*, a RNN, to decode the representation into a sentence comprised of a sequence of words following specific syntax. The simple encoder-decoder pipeline has achieved exceptional performance in describing visual contents with natural language [5]–[8], [10], [13]–[15]. However, in order to generate high-quality captions, a visual captioning model based on encoder-decoder pipeline has to make sure that its encoder captures crucial visual information. Decoder *only* takes the output of encoder as its input, hence information that the encoder fails to capture is doomed to be missing in the generated descriptions.

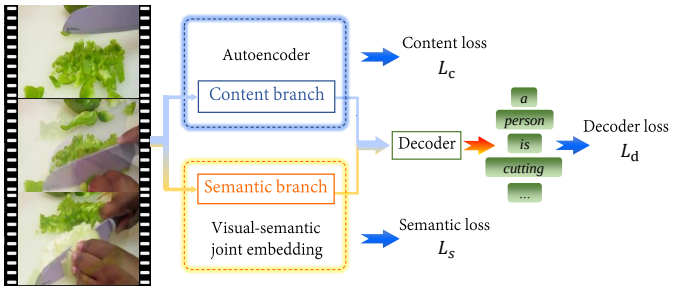


Fig. 1: Overview of the proposed SibNet, which employs a dual-branch architecture to collaboratively encode videos. The proposed loss function contains three components: content loss $L_c$, semantic loss $L_s$, and decoder loss $L_d$. We leverage autoencoder and visual-semantic joint embedding to impose fine-grained regularization that pushes content branch to capture visual contents and pushes semantic branch to encode video semantics.

The complexity and diversity of videos make it challenging to encode their contents. Different from a single image, a video, which is composed of a sequence of images, conveys much richer information. Therefore, existing single-branch video encoders, which encode video contents with a simple


- *S. Liu is with the Department of Computer Science and Engineering, University at Buffalo.*
  *E-mail: sliu66@buffalo.edu*
- *Z. Ren is with Wormpex AI Research.*
  *E-mail: renzhou200622@gmail.com*
- *J. Yuan is with the Department of Computer Science and Engineering, University at Buffalo.*
  *E-mail: jsyuan@buffalo.edu*

RNN or by averaging CNN features extracted from video frames, might lack the ability to represent these critical visual information needed for caption generation.

In this paper, we introduce a novel Sibling Convolutional Encoder (SibNet) which is able to provide a holistic representation of video information. Composed of a content branch and a semantic branch, SibNet encodes videos using a novel dual-branch architecture, which endows it greater representation power than single-branch encoders used by previous state-of-the-art methods. The content branch explicitly learns to capture important visual content information with an autoencoder; the semantic branch leverages visual-semantic joint embedding with ground truth captions in the training data so that it could produce semantic-specific representation. Then, soft-attention mechanism is used to efficiently combine video representations generated by both branches and finally, a RNN decoder is employed to decode the resulting representation into captions. Our SibNet is specifically designed for video captioning task and it brings the following two advantages: (1) the content branch is able to faithfully capture the visual contents of the video. As it is a pure visual encoder, it can better capture the video details to provide more precise captions; (2) the semantic branch leverages visual-semantic joint embedding to produce semantic-specific representation, which has been shown to be essential for our visual captioning task. Such a representation can capture how important certain frame is semantically, thus providing complementary information of the content branch. To better model temporal structures of videos, we propose to use a novel temporal convolutional block (TCB) rather than a RNN. Based on temporal convolution, TCB provides more efficient video temporal encoding than RNN with much less number of parameters.

We jointly train all the components of our model, the encoder, *i.e.*, SibNet, and the decoder by minimizing a novel objective function composed of three loss terms: (1) content loss from the content branch, (2) semantic loss from the semantic branch, and (3) decoder loss from the RNN decoder. The content loss is used to ensure that the content branch captures critical visual content information needed to reconstruct its original input, while the semantic loss is used to make sure the semantic branch encodes video semantic information consisted in the ground-truth captions. The decoder loss, which is also used by previous state-of-the-art methods, is used to ensure that the decoder generates coherent (syntactically correct) sentences. In our joint optimization framework, these three loss terms regularize each other, hence the video representation learned by our SibNet contains the information necessary to generate high quality captions for unseen videos. Figure 1 illustrates an overview of the proposed SibNet.

To showcase the effectiveness of our SibNet, we evaluate its performance on two standard video captioning benchmarks, *i.e.*, YouTube2Text [16] and MSR-VTT [17]. With the proposed dual-branch architecture capturing crucial and complementary information of videos, our model noticeably outperforms previous state-of-the-art methods across different evaluation metrics, including those methods that rely on additional external training data. We also offer insights into our SibNet with comprehensive ablation studies regarding contributions of the core components, *e.g.*, TCB, the content branch, the semantic branch, to the overall performance. Our experiments verify the unique merits of the proposed dual-branch architecture for video captioning as well as the superiority of the proposed TCB for temporal structure modeling.

## 2 Related Work

### 2.1 Sequence to Sequence Learning

Sequence to sequence learning [11], [18], [19], which endows neural networks with the ability to deal with variable-length input and output sequences in an end-to-end manner, achieves remarkable success in neural constituency parsing [20], [21], neural machine translation [11], [12], [19], neural text summarization [22], *etc*.

Originally proposed by Cho et al. [19] and Sutskever et al. [11], sequence to sequence models are composed of an encoder and a decoder (encoder-decoder pipeline), both of which are RNNs. The encoder encodes a variable-length input sequence into a fixed-length vectorial representation; the decoder decodes the vectorial representation into another variable-length output sequence. Bahdanau et al. [12] improved sequence to sequence models' ability to handle long sequences by proposing attention mechanism, which makes it possible for the decoder to selectively focus on parts of the input sequence. Luong et al. [23] examined a combination of multi-task learning and sequence to sequence learning by proposing models which shared encoders or decoders between several related tasks. Recently, there emerges a growing number of sequence to sequence models which are composed of non-recurrent encoders or decoders [24], [25]. Gehring et al. [24] proposed a convolutional sequence to sequence model which performs sequence modeling with a CNN. Vaswani et al. [25] proposed Transformer, *i.e.*, a sequence to sequence model solely relies on attention mechanism.

### 2.2 Image Captioning

Recently, encoder-decoder pipeline becomes a novel paradigm for neural image captioning [5], [7], [26]–[29]. Unlike previous models, which are composed of several separately tuned sub-modules, *e.g.*, attribute prediction module, object detection module, models based on encoder-decoder pipeline first encodes images into feature vectors using CNNs, *e.g.*, ResNet [30], Inception [31], GoogLeNet [32], and then decodes the feature vectors into captions with RNNs.

Following this paradigm, various neural image captioning models have been proposed in literature. Xu et al. [27] and You et al [5] introduced models with attention mechanism, whose decoders are able to focus on salient objects or semantic concepts, respectively. Johnson et al. [33] presented the task of dense captioning, which aims to generate captions for every meaningful regions in an image. Ren et al. [7] proposed a novel decision-making framework for image captioning. With a "policy network" providing local guidance and a "value network" providing global guidance, [7] is capable of generating more accurate captions.

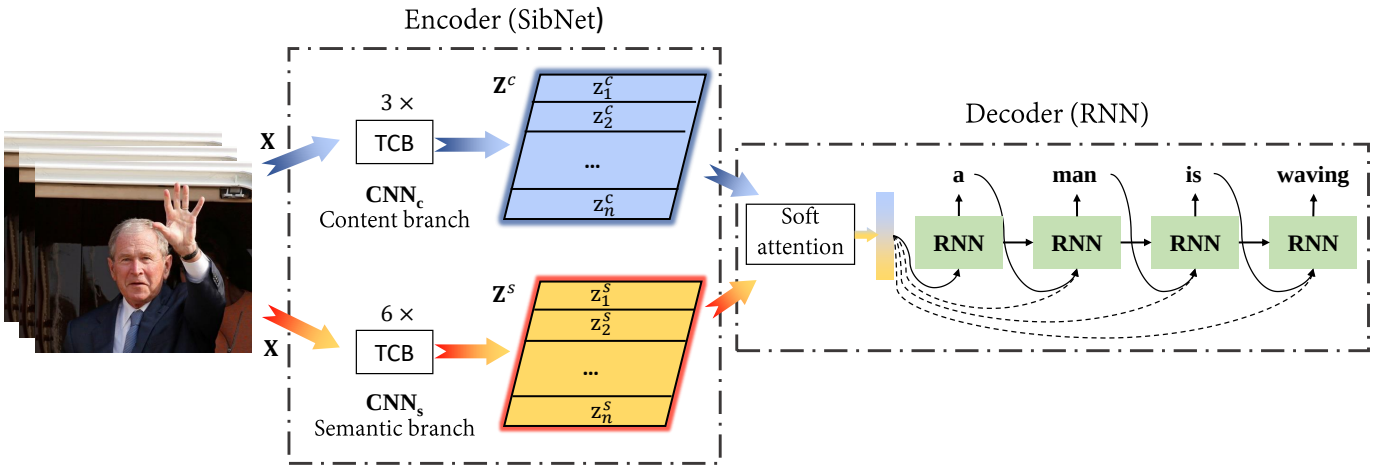


Fig. 2: Illustration of the proposed Sibling Convolutional Encoder (SibNet), which is composed of the content branch and the semantic branch, denoted as $\text{CNN}_c$ and $\text{CNN}_s$, respectively. We construct both branches by stacking 3 and 6 identical temporal convolutional blocks (TCBs) (we will introduce TCB in Section 3.1.3). Soft-attention mechanism is utilized in our RNN decoder.

### 2.3 Video Captioning

Recently, encoder-decoder architecture based on deep learning demonstrates its effectiveness in video captioning [2]–[4], [6], [8], [9], [15], [34]–[36]. Specifically, these models first adopt an encoder to represent videos with feature vectors and then employ a decoder to generate natural language captions.

However, unlike image captioning, where CNNs [30]–[32], [37] emerge as a paradigm to encode image contents in most state-of-the-art methods [5], [7], [26], [27], for the task of video captioning, how to effectively encode video contents is still an open problem. Venugopal et al. [2] proposed to represent video information using a fixed-length vector which is generated by performing mean pooling over frame-level feature vectors. In order to better leverage rich temporal information in videos, Venugopal et al. [3] proposed to encode frame-level feature vectors into a vector representation with a RNN instead of mean pooling layer. Pan et al. [6] adopted a hierarchical long short-term memory (LSTM) and Zhu et al. employed a bidirectional LSTM [38] to encode video contents. With the success of attention mechanism in neural machine translation [12] and neural image captioning [5], [27], Yao et al. [39] proposed to utilize attention mechanism for video captioning. Gan et al. [8] and Pan et al. [9] attempted to improve previous methods by detecting manually defined semantic concepts (attributes) and using detected concepts during decoding phase. Chen et al. [40]–[42] leveraged latent topic information of videos to train a video captioning model which is more proficient at utilizing words and phrases within a topic. Chen et al. [43] proposed to generate natural language captions based on a few informative frames picked by their method rather than all the frames of the video. Wang et al. [44] used multi-modal memory to model long-term dependency between video contents and natural language descriptions. Video captioning methods whose decoder leverages linguistic information mined from external training corpus [4] and whose decoder reconstructs video representation produced by the encoder [36] have been presented as well.

Comparing with the aforementioned models, which mostly encode video information in a single flow, our proposed SibNet learns to explicitly and effectively encode the visual content and semantic information of videos using a dual-branch architecture.

## 3 Model

Video captioning is the task of generating natural language description, *i.e.*, a sentence $\mathcal{Y}$, for a given video $V$. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ denote the ordered feature vectors extracted from $n$ frames in video $V$, $\mathbf{X} \in \mathbb{R}^{n \times d}$. Given $\mathbf{X}$ as input, an encoder generates a compact representation $\mathbf{Z}$ of visual information in $\mathbf{X}$. Video representation $\mathbf{Z}$ is either a fixed-length vector or a matrix composed of $n$ fixed-length vectors. Then, a decoder decodes video representation $\mathbf{Z}$ into a sentence $\mathcal{Y} = [y_1, y_2, \ldots, y_m]$, *i.e.*, a sequence of $m$ words.

We follow the encoder-decoder pipeline but propose a novel Sibling Convolutional Encoder (SibNet) to encode videos.

### 3.1 Sibling Convolutional Encoder (SibNet)

As shown in Figure 2, SibNet is comprised of two branches, namely the content branch and the semantic branch, which are denoted as $\text{CNN}_c$ and $\text{CNN}_s$, respectively. The content branch is designed to encode video content information, while the semantic branch is designed to encode video semantic information. Unlike existing encoders, whose encoded feature $\mathbf{Z}$ is either a fixed-length vector or a matrix, the representation $\mathbf{Z}$ learned by SibNet is composed of *two* matrices $\mathbf{Z}^c$ and $\mathbf{Z}^s$ both of which are formed by $n$ vectors. We name our model SibNet because $\text{CNN}_c$ and $\text{CNN}_s$ possess common properties: firstly, they share the same input $\mathbf{X}$. Besides, both branches are formed by a stack

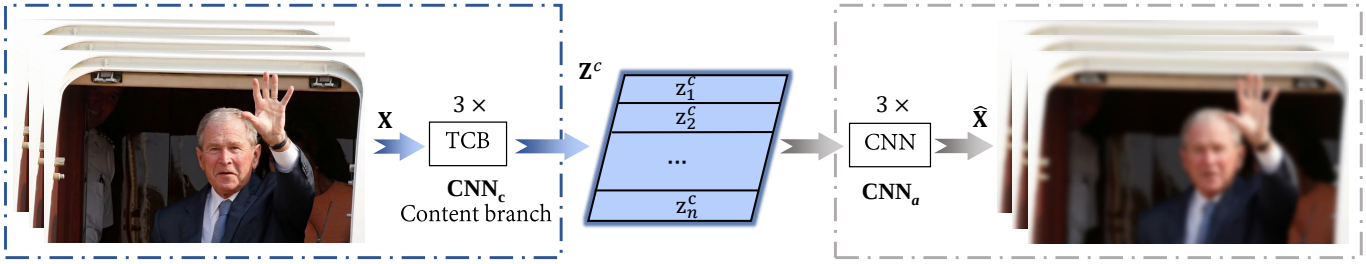


Fig. 3: Illustration of the content branch CNN$_c$ implemented via an autoencoder. Note that the content loss of the autoencoder is one component of our final training loss.

of temporal convolutional blocks (TCBs) (we will introduce TCB in Section 3.1.3). Now let us introduce both branches in details.

#### 3.1.1 Content branch

The role of our content branch is to encode visual content information. Motivated by the success of autoencoders [45]–[47] in visual representation learning, we propose to push our content branch to play its role with an autoencoder, which has the ability to learn a compact representation that captures most crucial visual content information contained in its input.

As shown Figure 3, our autoencoder takes $\mathbf{X}$ as input, and then encodes video contents in $\mathbf{X}$ into a compact representation $\mathbf{Z}^c$, whose dimension is less than the dimension of $\mathbf{X}$, with our content branch CNN$_c$. CNN$_c$ is composed of 3 TCBs (we will introduce TCB in Section 3.1.3). After that, the autoencoder attempts to reconstruct $\mathbf{X}$, its original input, from $\mathbf{Z}^c$ using a simple 3-layer convolutional neural network denoted as CNN$_a$. Let $\hat{\mathbf{X}}$ denote the reconstruction generated by CNN$_a$. We adopt mean squared error between $\mathbf{X}$ and $\hat{\mathbf{X}}$, which reflects how well we could reconstruct the original input $\mathbf{X}$ from representation $\mathbf{Z}^c$, as content loss $L_c$. Specifically, $L_c$ is defined as follows:

$$L_c = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\mathbf{x}}_i)^2, \tag{1}$$

where $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$ denote the $i$-th vectors of $\mathbf{X}$ and $\hat{\mathbf{X}}$, respectively. This unsupervised reconstruction loss of our autoencoder forces our content branch to produce high-quality representation $\mathbf{Z}^c$ capturing salient visual content information and is incorporated in the final training loss.

#### 3.1.2 Semantic branch

The goal of our semantic branch is to generate a representation of semantic-relevant information in videos. The success of visual-semantic joint embedding in image classification [48], [49] and image retrieval [50], [51] inspires us to implement our semantic branch via visual-semantic joint embedding.

As shown in Figure 4, our visual-semantic joint embedding model is composed of a captioning embedding module and a video embedding module. These two modules map captions and videos into a *common* semantic space in which the embedding of a video and its corresponding captions is close. Hence, the distance between caption embedding and video embedding is empowered with semantic meaning.

Given a caption $\mathcal{Y}$ that consists of $m$ word, we represent it with a matrix $\mathbf{W}$ that is composed of $m$ word vectors, $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m]$. $w_i$ $(1 \leq i \leq m)$ denotes the word vector of the $i$-th word. Instead of using the average of the $m$ word vectors [52] as the captioning embedding vector $\mathbf{c}_e$, we utilize self-attentive network (SAN) [53] to embed $\mathbf{W}$ into $\mathbf{c}_e$:

$$\mathbf{c}_e = \sum_{i=1}^{m}\eta_i \mathbf{w}_i. \tag{2}$$

Here, $\eta_i \in \mathbb{R}$ denotes the weight value assigned to the $i$-th word vector $\mathbf{w}_i$. The weight values $(\eta_1, \eta_2, \ldots, \eta_m) \in \mathbb{R}^{1\times m}$ are computed as follows:

$$(\eta_1, \eta_2, \ldots, \eta_m) = \text{softmax}(\mathcal{F}(\mathbf{W})), \tag{3}$$

where $\mathcal{F}$ represents a nonlinear function that SAN, *i.e.*, a 2-layer feed-forward neural network implements. We use SAN because it has been proven in [53] that SAN is able to capture semantic information that only presents in certain meaningful parts in its input, *e.g.*, representation of a caption $\mathbf{W}$.

A video embedding vector $\mathbf{v}_e$ is computed similar to the way $\mathbf{c}_e$ is computed. The video embedding module first utilize the semantic branch CNN$_s$ to map $\mathbf{X}$ to $\mathbf{Z}^s$ and then employ SAN to map $\mathbf{Z}^s$ to a $\mathbf{v}_e$ by computing a weighted average of vectors in $\mathbf{Z}^s$.

Following [7], we utilize bi-directional ranking loss as our semantic loss to make CNN$_s$ effectively encode semantic information. Specifically, we define semantic loss $L_s$ as follows:

$$\begin{aligned} L_s = &\sum_{\mathbf{v}_e}\sum_{\mathbf{c}_e^-}\max(0, m - \mathbf{v}_e\cdot\mathbf{c}_e + \mathbf{v}_e\cdot\mathbf{c}_e^-)) \\ &+ \sum_{\mathbf{c}_e}\sum_{\mathbf{v}_e^-}\max(0, m - \mathbf{c}_e\cdot\mathbf{v}_e + \mathbf{c}_e\cdot\mathbf{v}_e^-)), \end{aligned} \tag{4}$$

where $\cdot$ designates dot product operation. $m$ is the margin set to be $0.1$ by cross-validation. Given a video $V$ with embedding vector $\mathbf{v}_e$, $\mathbf{c}_e$ denotes embedding of its ground truth caption, $\mathbf{c}_e^-$ denotes embedding of a negative caption that describes videos other than $\mathbf{v}_e$; and vice-versa with $\mathbf{v}_e^-$. This semantic loss is incorporated into our final training loss, which pushes our semantic branch to play its role.

#### 3.1.3 Temporal convolutional block (TCB)

Now we introduce temporal convolutional block (TCB) (shown in Figure 5), which is the basic component in both

our content branch and our semantic branch. It is of great importance to exploit temporal information in videos so as to learn a holistic representation that works well for video captioning [39] and video understanding [54]. Different from previous works which employs RNN for video temporal structure modelling [9], [34], [39], we propose a simpler temporal modeling architecture, *i.e.*, TCB, which works effectively in our experiments.

As shown in Figure 2, both our content and semantic branches consist of a stack of TCBs. Let $\mathbf{X}_k = [\mathbf{x}_1^k, \mathbf{x}_2^k, \ldots, \mathbf{x}_n^k]$ denote input of the $k$-th TCB in either branch, where each $\mathbf{x}_i^k$ is a $d_k$-dimensional vector, $\mathbf{X}_k \in \mathbf{R}^{n \times d_k}$. Firstly, the $k$-th TCB employs a bottleneck architecture to reduce the dimension of its input $\mathbf{X}_k$ by passing it through $\text{TCN}_1$, a temporal convolutional layer, *i.e.*, a 1-dimensional convolutional layer, with kernel size 1, and ReLU activation function. The output of the ReLU activation function, $\mathbf{X}_k^{'} \in \mathbb{R}^{n \times d_k^{'}}$, has less dimensions than $\mathbf{X}_k$ (*i.e.*, $d_k^{'} \leq d_k$), owing to the bottleneck architecture. Then, $\mathbf{X}_k^{'}$ is passed through $\text{TCN}_2$, another temporal convolutional layer with kernel size 3, and GLU [55] activation function, which is defined as follows:

$$\mathcal{F}(\mathbf{X}_k^{'}) = \tanh(\mathbf{W}_k * \mathbf{X}_k^{'}) \odot \sigma(\mathbf{W}_k * \mathbf{X}_k^{'}), \tag{5}$$

where $\mathcal{F}(\mathbf{X}_k^{'})$ represents the output of GLU, $\mathbf{W}_k$ denotes learnable parameters of $\text{TCN}_2$, $*$ represents convolution operator, $\odot$ denotes element-wise multiplication, $\sigma(\cdot)$ and $\tanh(\cdot)$ denote sigmoid and hyperbolic tangent functions. In our experiments, we find that using GLU rather than ReLU as the activation function for $\text{TCN}_2$ improves the performance of SibNet. We believe it can be because GLU injects more non-linearity and increases the representation capacity of SibNet. Inspired by the idea of DenseNet [56], use dense connection, which concatenates the output of GLU and the original input $\mathbf{X}_k$, to get the output of the $k$-th TCB, *i.e.*, $cat(\mathcal{F}(\mathbf{X}_k^{'}), \mathbf{X}_k)$. Using dense connection improve the flow of visual information and gradients [56] throughout both branches of SibNet. The output of the $k$-th TCB, *i.e.*, $cat(\mathcal{F}(\mathbf{X}_k^{'}), \mathbf{X}_k)$, becomes the input of the $(k+1)$-th TCB.

As shown in Figure 2, our content branch is composed of a stack of 3 TCBs, while our semantic branch is composed of a stack of 6 TCBs. In Section 4.3.3, we will investigate the impact of the TCB numbers in both branches and thus explain why we choose such numbers as above.

### 3.2 Decoder

Following previous work, we use a RNN to decode the encoded representation $\mathbf{Z}$, *i.e.*, $\{\mathbf{Z}^c, \mathbf{Z}^s\}$ into a sentence $\mathcal{Y}$. After we obtain the encoded representation $\mathbf{Z}$, *i.e.*, $\{\mathbf{Z}^c, \mathbf{Z}^s\}$, we follow previous work to use a RNN to decode it into a sentence $\mathcal{Y}$. More specifically, given $\mathbf{Z}^c$ and $\mathbf{Z}^s$, the decoder predicts joint probability $p(\mathcal{Y})$ of caption $\mathcal{Y}$ by sequentially predicting the probability of each word $y_i$ in sentence $\mathcal{Y}$. It can be seen from Figure 2 that our decoder is auto-regressive, indicating that it takes the output at the previous time step as additional input.

We maximize the probability of generating the ground truth caption $\mathcal{Y}$ by minimizing the decoder loss $L_d$, which is cross-entropy loss defined as follows:

$$L_d = -\log(p(\mathcal{Y}|\mathbf{Z}^c, \mathbf{Z}^s)). \tag{6}$$

#### 3.2.1 Soft-attention mechanism

How to effectively combine $\mathbf{Z}^c$ and $\mathbf{Z}^s$ is the key problem in decoding process. We utilize a soft-attention mechanism. Originally proposed in [12], variants of soft attention have been successfully applied to machine translation [12], image captioning [27] and video captioning [15], [39], etc. Different from standard soft-attention mechanism [12] which returns a fixed-length vector encoding information of one single matrix, our soft-attention mechanism merges visual information of *two* matrices $\mathbf{Z}^c$ and $\mathbf{Z}^s$ in a fixed-length vector. At decoding time step $i$ when generating the $i$-th word, our soft-attention mechanism computes the input vector $\mathbf{u}_i$ of RNN decoder as follows:

$$\mathbf{u}_i = \sum_{j=1}^{n} \text{softmax}_j(\mathbf{s}_i) \cdot \mathbf{z}_j^c \qquad j \in [1, n], \tag{7}$$

where $\text{softmax}_j(\cdot)$ denotes the $j$-th element of the output vector of the softmax function, $\mathbf{z}_j^c$ is the $j$-th element of $\mathbf{Z}^c$ that encodes video content information, and $\mathbf{s}_i = [s_{i,1}, s_{i,2}, \ldots, s_{i,n}]$ is defined as follows:

$$s_{i,k} = \mathbf{W_s}^T \tanh(\mathbf{W}_h \mathbf{h}_i + \mathbf{W}_z \mathbf{z}_k^s) \qquad k \in [1, n]. \tag{8}$$

Here, $s_{i,k}$ is a real value; $\mathbf{W_s}, \mathbf{W}_h$ and $\mathbf{W}_z$ are learnable weight matrices; $\mathbf{h}_i$, a fixed-length vector, denotes the hidden state of the RNN decoder at the $i$-th time step; $\mathbf{z}_k^s$ is the $k$-th element of $\mathbf{Z}^s$, which encodes video semantic information.

As shown in Equation 7 and 8, the soft-attention mechanism utilizes semantic information in $\mathbf{Z}^s$ to determine a weighting value $\mathbf{s}_i$, which then effectively combine the visual content representation $\mathbf{Z}^c$ to generate a input vector $\mathbf{u}_i$ for RNN decoder. Such soft-attention mechanism is able to ensure our decoder pay more "attention" to the visual content of certain frames if they contain important semantic information. As we can see, by using the proposed soft-attention mechanism, the content and semantic branch in SibNet are effectively combined in a complementary fashion.

### 3.3 Training

We jointly train all the components of our model, the content branch, the semantic branch, and the RNN decoder in an end-to-end manner. As introduced before, autoencoder and visual-semantic joint embedding are utilized to impose more fine-grained supervision for both branches of SibNet. Our autoencoder forces the content branch to encode crucial visual content information, while our visual semantic joint embedding model leverages ground truth captions to ensure the semantic branch captures semantic information needed to generate precise captions. Thus, the final training loss function is defined as follows:

$$L = L_d + \alpha L_c + \beta L_s, \tag{9}$$

where $L_d, L_c$ and $L_s$ denote the decoder loss, content loss and semantic loss, as defined in Equation 6, Equation 1 and Equation 4, respectively; $\alpha$ and $\beta$ are two scalars that control the influence of content loss and semantic loss during training. We set $\alpha$ and $\beta$ to be 0.4 and 1 by cross validation.

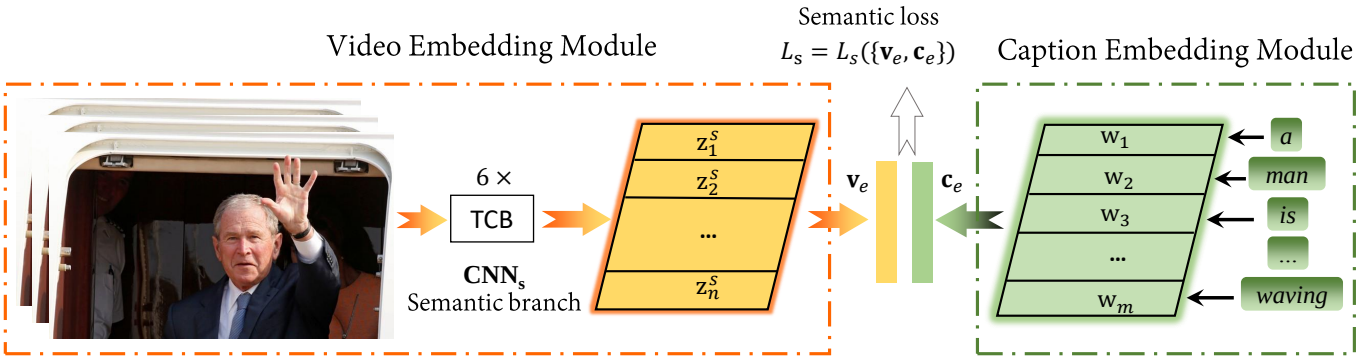


Fig. 4: Illustration of the semantic branch CNN$_s$ implemented via visual-semantic joint embedding. Note that the semantic loss of visual-semantic embedding is one component of our final training loss.

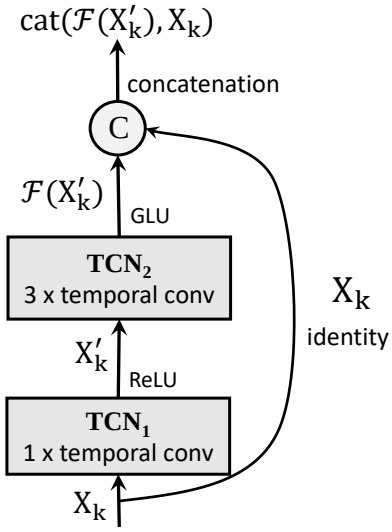


Fig. 5: Illustration of our temporal convolutional block (TCB), which is the basic component of both the content branch and the semantic branch.

## 4 Experiments

We test the proposed SibNet on two video captioning benchmarks, YouTube2Text (MSVD) [16] and MSR-VTT [17]. For fair comparison, all the reported results are obtained using Microsoft COCO caption evaluation tool [57]. We utilize Bleu [58], METEOR [59], CIDEr [60] and ROUGE [61] as our evaluation metrics, which are commonly used for performance evaluation of video captioning methods.

### 4.1 Experiment Setup

#### 4.1.1 Datasets

YouTube2Text is composed of 1970 YouTube videos and 78,800 captions (40 captions per video, on average) annotated by Amazon Mechanical Turk (AMT) annotators. For fair comparison, we adopt the same evaluation scheme proposed in [3], which used 1200 videos for training, 100 videos for validation and 670 videos for testing. MSR-VTT is a large-scale video captioning dataset, which is comprised of 10,000 videos and 200,000 captions (20 unique captions per video). We adopt the standard dataset splits proposed in [17], which used 6513 videos for training, 497 videos for validation and 2990 videos for testing.

For both YouTube2Text and MSR-VTT datasets, we uniformly sample the videos with a sampling rate of 3 frames per second. We then extract visual features using GoogLeNet [32] for YouTube2Text dataset and Inception [31] for MSR-VTT dataset. Both GoogLeNet and Inception are trained by Wang et al. [62]. It is worth noting that most state-of-the-art methods [6], [8]–[10], [34], [39], [40], [63], [64] take a combination of multiple complementary features, including frame-level CNN features (ResNet [30]), clip-level CNN features (C3D [65]) and audio features (MFCC [66]), as input to their encoders. We do not adopt feature combination in our experiments.

#### 4.1.2 Network architecture

For the content branch, we set the output dimensions of the TCN$_1$ and the TCN$_2$, two temporal convolutional layers, in each TCB to be 180 and 45, respectively; for the semantic branch, we set them to be 120 and 30, respectively. We adopt 1-layer LSTM [67] with 1024-dimensional hidden state as our RNN decoder.[1] Many variants of RNN have been proposed in literature, *e.g.*, GRU [68], Peephole LSTM [69]. Some state-of-the-art methods have utilized them and have reported better performance. Although we choose a basic LSTM as decoder in our experiments, our model is modular w.r.t. the decoder architecture.

#### 4.1.3 Training details

We train our model using Adam [70] algorithm with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e-8$. For YouTube2Text dataset, we set the batch size to be 32. The initial learning rates are set to be 8e-5 for the encoder and 4e-5 for the decoder, respectively. For MSR-VTT dataset, we set the batch size to be 64. The initial learning rates are set to be 6e-5 for the encoder and 3e-5 for the decoder. For both datasets, the learning rates are divided by 5 after 10 epochs. We perform gradient clipping with a threshold of 2, and adopt weight initialization method proposed in [71]. We also regularize our model by applying dropout [72] to the output of each

1. Although SibNet, abbreviation for Sibling Convolutional Encoder, only refers to the encoder of our model, we also use it to refer to a combination of our encoder and the RNN decoder in the following sections.

TABLE 1: Performance comparisons on YouTube2Text (MSVD) dataset. * indicates that external datasets were used to train these models.

| Methods | Bleu-4 | METEOR | CIDEr | ROUGE |
|---|---|---|---|---|
| Joint-BiLSTM [74] | - | 30.3 | - | - |
| S2VT [3] | 37.0 | 29.8 | - | - |
| Temporal Attention [39] | 41.9 | 29.6 | 51.7 | - |
| BA Encoder [75] | 42.5 | 32.4 | 63.5 | - |
| GRU-RCN [76] | 43.3 | 31.6 | 68.0 | - |
| aLSTM [77] | 44.9 | 30.4 | 60.1 | - |
| LSTM-E [34] | 45.3 | 31.0 | - | - |
| HRNE + Attention [6] | 46.7 | 33.9 | - | - |
| p-RNN [63] | 49.9 | 32.6 | 65.8 | - |
| Latent Topic [40] | 48.8 | 34.4 | 80.5 | - |
| mGRU [38] | 49.5 | 33.4 | 75.5 | - |
| STAT [78] | 51.1 | 32.7 | 67.5 | - |
| DMRM [79] | 51.1 | 33.6 | 74.8 | - |
| MA-LSTM [10] | 52.3 | 33.6 | 70.4 | - |
| RecNet [36] | 52.3 | 34.1 | 80.3 | 69.8 |
| MMM [44] | 52.8 | 33.3 | - | - |
| TSA-ED [80] | 51.7 | 34.0 | 74.9 | - |
| PickNet [43] | 46.1 | 33.1 | 76.0 | 69.2 |
| LSTM-YT* [2] | 31.2 | 26.9 | - | - |
| GloVe + DeepFussion* [4] | 42.1 | 31.4 | - | - |
| SCN* [8] | 50.2 | 33.4 | 77.0 | - |
| LSTM-TSA* [9] | 52.8 | 33.5 | 74.0 | - |
| Ours | **55.7** | **35.5** | **88.8** | **72.6** |

TABLE 2: Performance comparisons on the test set of MSR-VTT: comparisons with state-of-the-art methods and methods that rank top-4 on the Leaderboard of MSR-VTT Challenge. * indicates that extra training data was used during training. $^{e}$ and $^{v}$ indicate that the reported performance was achieved by an *ensemble* of multiple models or was achieved on the *validation* set, respectively. (As WSDC [64] conducts extensive data augmentation, which none of the others conducts, we report the performance of [64] achieved on the *validation* set under similar settings as our model.)

| Methods | Bleu-4 | METEOR | CIDEr | ROUGE |
|---|---|---|---|---|
| Rank1: v2t-navigator [81] | 40.8 | 28.2 | 44.8 | 60.9 |
| Rank2: Aalto [82] | 39.8 | 26.9 | 45.7 | 59.8 |
| Rank3: VideoLAB [83] | 39.1 | 27.7 | 44.1 | 60.6 |
| Rank4: ruc-uva [84] | 38.7 | 26.9 | 45.9 | 58.7 |
| Mean Pooling [2] | 30.4 | 23.7 | 35.0 | 52.0 |
| Temporal Attention [39] | 28.5 | 25.0 | 37.1 | 53.3 |
| S2VT [3] | 31.4 | 25.7 | 35.2 | 55.9 |
| MA-LSTM [10] | 36.3 | 26.3 | 40.1 | 59.1 |
| aLSTM [77] | 38.0 | 26.1 | - | - |
| STAT [78] | 37.4 | 26.6 | 41.5 | - |
| RecNet [36] | 39.1 | 26.6 | 42.7 | 59.3 |
| MMM [44] | 38.1 | 26.6 | - | - |
| PickNet [43] | 38.9 | 27.2 | 42.1 | 59.5 |
| MTLE [85] | 39.2 | 26.6 | 42.1 | 59.3 |
| M2M*$^{e}$ [15] | 40.8 | **28.8** | 47.1 | 60.2 |
| WSDC$^{v}$ [64] | 39.0 | 27.7 | 44.0 | 60.1 |
| Ours | **41.2** | 27.8 | **48.6** | **60.8** |

TCB with a rate of 0.2. Additional regularization methods, *e.g.*, weight decay [73], are not utilized.

### 4.2 Comparison with the State-of-the-Art

#### 4.2.1 Results on YouTube2Text

In Table 1, we present the results of SibNet and existing methods on YouTube2Text dataset. As we can see, our model achieves the best performance across all metrics, improving Bleu-4 from 52.8 to 55.7, METEOR from 34.4 to 35.5, CIDEr from 80.5 to 88.8, ROUGE from 69.8 to 72.6, respectively. It is worth noting that large-scale external datasets (at least *two* times larger than YouTube2Text dataset) are utilized by LSTM-TSA [9], SCN [8], LSTM-YT [2] and GloVe + DeepFussion [4]. Surprisingly, even without using extra training data, our model significantly outperforms all of them. Besides, both LSTM-TSA [9] and SCN [8] rely on hundreds of dataset-specific "semantic attributes", which are manually selected from thousands of candidates. The laborious "semantic attribute" selection prevents [8], [9] to be applied to large dataset with more candidates. On the contrary, our model automatically learns representation of high-level semantics using the proposed semantic branch.

#### 4.2.2 Results on MSR-VTT

In Table 2, we show a comparison of SibNet and previous state-of-the-art methods on MSR-VTT dataset. We also compare SibNet with methods that occupy top-4 positions of the Leaderboard of MSR-VTT Challenge [86], denoted as Rank1: v2t-navigator [81], Rank2: Aalto [82], Rank3: VideoLAB [83] and Rank4: ruc-uva [84]. our model achieves the best performance across three of the four metrics. Note that the current best performing method, M2M [15], not only relies on two large-scale external datasets UCF101 [87] and SNLI [88] for training, but also utilizes an ensemble of multiple models. However, our model is trained without using extra training data and tested without model ensemble.

From Table 1 and Table 2, it can be seen that SibNet consistently outperforms state-of-the-art methods by a large margin even without extra training data and model ensemble, which validates the effectiveness of encoding video contents using the proposed dual-branch architecture.

#### 4.2.3 Qualitative analysis

We present a qualitative comparison of our model, *i.e.*, SibNet, and previous state-of-the-art methods, including S2VT [3] and Temporal Attention [39] in Figure 9, where "`S2VT`" ,"`TA`" and "`Ours`" denote captions generated by S2VT [3], Temporal Attention [39] and our model, respectively. "`GT`" denotes ground truth captions. We highlight both *incorrect* (shown in blue) and *correct* (shown in red) words or phrases in the generated captions.

We can see that our model can generate captions that correctly describe their corresponding videos in terms of both high-level semantics (as shown in the first example in the third and forth rows of Figure 9,) and low-level details (as shown in the two examples in the fifth row of Figure 9), while S2VT [3] and Temporal Attention [39] sometimes can not. This verifies the ability of our model to encode rich visual information in the videos.

TABLE 3: A comparison of the performance of TCB-based network and its RNN-based counterparts, including networks based on GRU, bidirectional GRU (BiGRU), LSTM and bidirectional LSTM (BiLSTM). The numbers of parameters (No. Param) in these networks are compared as well. The dimensions of the hidden states of GRU and LSTM are set to be 512, and those of BiGRU and BiLSTM are set to be 256 for a fair comparison.

| Methods | Bleu-1 | Bleu-2 | Bleu-3 | Bleu-4 | METEOR | CIDEr | ROUGE | No. Param |
|---|---|---|---|---|---|---|---|---|
| *GRU* | 80.9 | 69.7 | 60.6 | 51.7 | 32.9 | 79.2 | 70.4 | 4.7M |
| *BiGRU* | 80.7 | 69.5 | 60.1 | 51.0 | 33.0 | 78.6 | 70.2 | 4.7M |
| *LSTM* | 81.0 | 70.1 | 60.9 | 52.6 | 33.5 | 80.4 | 70.5 | 6.3M |
| *BiLSTM* | 81.1 | 70.3 | 61.2 | 52.9 | 33.6 | 82.2 | 70.4 | 6.3M |
| *Ours* | **82.7** | **72.1** | **63.7** | **55.7** | **35.5** | **88.8** | **72.6** | **0.8M** |

TABLE 4: Performance of different variants of the proposed SibNet, whose content branch and semantic branch are formed by a stack of different variants of TCB, on YouTube2Text dataset.

| Methods | Bleu-1 | Bleu-2 | Bleu-3 | Bleu-4 | METEOR | CIDEr | ROUGE | No. Param |
|---|---|---|---|---|---|---|---|---|
| *TConv* | 81.7 | 71.3 | 62.3 | 53.3 | 33.7 | 87.0 | 71.4 | 2.3M |
| *TCB (conf)* | 81.5 | 70.8 | 62.4 | 54.2 | 34.8 | 88.2 | 71.7 | 2.3M |
| *TCB (tanh)* | 81.6 | 70.9 | 62.5 | 54.3 | 34.0 | 87.9 | 71.7 | **0.8M** |
| *TCB (ReLU)* | 82.0 | 71.3 | 62.9 | 55.1 | 34.4 | 87.7 | 71.6 | **0.8M** |
| *TCB (w/o btl)* | 82.5 | 72.0 | 63.7 | 55.2 | 34.5 | **89.1** | 72.5 | 1.2M |
| *Ours* | **82.7** | **72.1** | **63.7** | **55.7** | **35.5** | 88.8 | **72.6** | **0.8M** |

### 4.3 Ablation Study

Since SibNet fundamentally differs from the encoders employed by existing video captioning methods, in this section we perform detailed ablation study to get a better understanding of the proposed SibNet.

#### *4.3.1 Merits of our temporal convolutional block (TCB)*

To verify the effectiveness of the proposed TCB, we compare TCB with RNN-based video temporal structure modeling components utilized by previous works [6], [9], [39] including GRU, bidirectional GRU (BiGRU), LSTM and bidirectional LSTM (BiLSTM) on YouTube2Text dataset. We also compare the numbers of parameters in these networks are compared as well. For a fair comparison, all of these networks are trained with the same loss function defined in Equation 9. As demonstrated in Table 3, TCB (denoted as *Ours*) outperforms its four RNN-based counterparts, including the bidirectional ones, across all of the evaluation metrics. Comparing to GRU and BiGRU, TCB improves the Bleu-4 by 4.0 and 4.7, METEOR by 2.5 and 2.6, CIDER by 9.6 and 10.2, ROUGE by 2.2 and 2.4. In comparison with LSTM and BiLSTM, TCB improves Bleu-4 from 52.7 and 52.8 to 55.7, METEOR from 33.3 and 33.3 to 35.5, CIDER from 80.4 and 82.2 to 88.8, ROUGE from 70.6 and 70.7 to 72.6, respectively. In addition, the number of parameters in TCB-based SibNet is less than $18\%$ of those in the GRU-based networks and is less than $13\%$ of those in the LSTM-based networks. The performance improvement brought by TCB is partially attributed to such a notable reduction of the number of parameters, as it is easier to train a network with less parameters. TCB outperforms widely used variants of RNN both in terms of performance and number of parameters as a tool for video temporal structure modeling.

To further validate the effectiveness of TCB, we conduct experiments to examine the contribution of the three major design choices of TCB: (1) using GLU rather than ReLU and tanh as the activation function (2) using concatenation (dense connection) instead of element-wise summation (residual connection) to merge the input of TCB and the output of GLU, *i.e.*, $\mathcal{F}(\mathbf{X}_k^{'})$ shown in Equation 5, and (3) using bottleneck architecture. We compare the performance and the number of parameters of five variants of TCB, denoted as: *TConv*, *TCB (conf)*, *TCB (tanh)*, *TCB (ReLU)* and *TCB (w/o btl)*. Firstly, *TConv)*, which uses a temporal convolutional layer for video temporal structure modeling, could be viewed as our baseline. Besides, *TCB (conf)*, the TCB proposed in Section 3.1.2 of the conference version of this manuscript, improves *TConv)* by introducing residual connection [30]. Finally, *TCB tanh* and *TCB (ReLU)* denote variants of TCB that adopt tanh and ReLU as the activation function applied to the output of $\text{TCN}_2$, while *TCB (w/o btl)* represents variant that does not have the bottleneck architecture. These three variants differ from TCB proposed in Section 3.1.3 (denoted as *Ours*) only in activation function or bottleneck architecture, respectively.

From Table 4 that shows the results of all variants above on YouTube2Text dataset, we observe that:

1) By using dense connection to facilitate the flow of visual information and gradient in the proposed TCB, *Ours* outperforms both *TConv* and *TCB (conf)* by a large margin, thus verifying the unique merits of using dense connection instead of temporal convolution only or residual connection.
2) Comparing with *TCB (tanh)* and *TCB (ReLU)*, *TCB* achieves solid performance improvements over all metrics, thus, validating the advantage of GLU over tanh and ReLU as the activation function.
3) *TCB (w/o btl)* performs worse than *TCB*, hence illustrating the advantages of performing dimensionality reduction with bottleneck architecture.

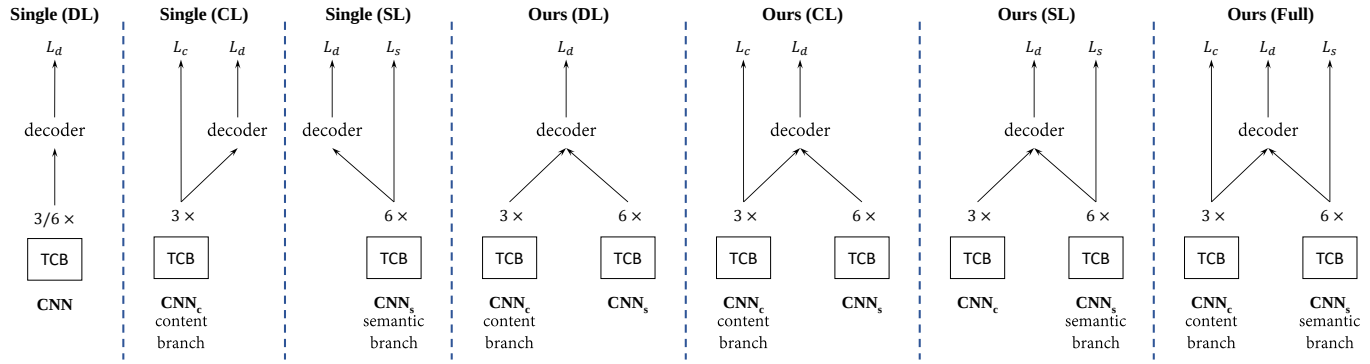


Fig. 6: Variants of SibNet, which we compare in Table 5. Here, $CNN_c$, $CNN_s$ and $RNN_d$ represent the content branch, the semantic branch, and the decoder, respectively. "AE" and "JE" denote the remaining modules in the autoencoder and the visual-semantic joint embedding model. $L_c$, $L_s$ and $L_d$ represent the content loss, the semantic loss and the decoder loss, which are defined in Equation 4, 1 and 6.

TABLE 5: Performance of different variants of the proposed SibNet on YouTube2Text and MSR-VTT datasets.

| Methods | Dataset | $L_d$ | $L_c$ | $L_s$ | Bleu-1 | Bleu-2 | Bleu-3 | Bleu-4 | METEOR | CIDEr | ROUGE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *Single (DL-3)* | MSR-VTT | ✓ | | | 76.5 | 63.1 | 50.6 | 39.3 | 27.0 | 45.0 | 59.3 |
| *Single (DL-6)* | | ✓ | | | 76.1 | 63.3 | 51.0 | 40.0 | 27.0 | 45.9 | 59.7 |
| *Ours (DL)* | | ✓ | | | 77.5 | 64.4 | 51.9 | 40.7 | 27.2 | 46.3 | 60.2 |
| *Ours (CL)* | | ✓ | ✓ | | 77.2 | 64.6 | 52.3 | 41.0 | 27.5 | 47.0 | 60.2 |
| *Ours (SL)* | | ✓ | | ✓ | 77.4 | 64.9 | 52.7 | **41.5** | 27.5 | 47.8 | 60.5 |
| *Ours (Full)* | | ✓ | ✓ | ✓ | **78.3** | **65.4** | **52.8** | 41.2 | **27.8** | **48.6** | **60.8** |
| *Single (DL-3)* | YouTube2Text | ✓ | | | 80.4 | 69.1 | 59.7 | 49.9 | 32.2 | 79.8 | 70.0 |
| *Single (DL-6)* | | ✓ | | | 80.7 | 69.3 | 59.9 | 50.1 | 32.6 | 78.5 | 70.2 |
| *Ours (DL)* | | ✓ | | | 81.0 | 70.4 | 61.5 | 52.6 | 34.2 | 84.7 | 70.8 |
| *Ours (CL)* | | ✓ | ✓ | | 82.2 | 71.2 | 62.4 | 54.1 | 34.9 | 88.3 | 72.0 |
| *Ours (SL)* | | ✓ | | ✓ | 81.8 | 71.3 | 62.7 | 54.5 | 34.9 | **88.8** | 71.6 |
| *Ours (Full)* | | ✓ | ✓ | ✓ | **82.7** | **72.1** | **63.7** | **55.7** | **35.5** | **88.8** | **72.6** |

#### 4.3.2 How much does each component contribute?

In order to analyze the impact of different components of our proposed model on the performance of video captioning, we evaluate five variants of our model (shown in Figure 6): *Single (DL-3)*, *Single (DL-6)*, *Ours (DL)*, *Ours (CL)* and *Ours (SL)*, respectively. First of all, *Single (DL-3)* and *Single (DL-6)* denote two single-branch encoders which only consist of 3 and 6 identical TCBs. These two variants, which encode visual information using a single branch, could be viewed as the baseline of our model. And both of them are trained using decoder loss $L_d$ alone. To validate the superiority of the proposed dual-branch architecture over the baseline, we construct *Ours (DL)*, which has both the content branch and the semantic branch. But *Ours (DL)* is also trained with decoder loss $L_d$ alone. To evaluate the effectiveness of our proposed training scheme which provides more fine-grained training supervision, we construct two variants: *Ours (CL)* and *Ours (SL)*. *Ours (CL)* incorporates the autoencoder to impose a content loss $L_c$ as defined in Equation 1 to *Ours (DL)*. Likewise, *Ours (SL)* incorporates visual-semantic embedding to impose a semantic loss $L_s$ as defined in Equation 4 to *Ours (DL)*. Lastly, we evaluate *Ours (Full)*, which is our full model.

From Table 5 that shows the results of all variants above on both MSR-VTT and YouTube2Text, we observe that:

1) Comparing with *Ours (DL)*, *Single (DL-3)* and *Single (DL-6)* have worse performance. This indicates the necessity of encoding visual information using our proposed dual-branch architecture. It is worth noting that the performance of *Single (DL-3)* and *Single (DL-6)* is on a par with many existing methods, which validates the effectiveness of modeling video temporal structures of videos using TCB as described in Section 3.1.3.
2) By adding content loss $L_c$ to decoder loss $L_d$ used by *Ours (DL)*, *Ours (CL)* achieves better performance than *Ours (DL)*. This verifies the efficiency of regularizing the content branch using autoencoder. Similarly, by adding semantic loss $L_s$, *Ours (SL)* also outperforms *Ours (DL)* by a large margin. This validates the importance of regularizing the semantic branch by leveraging visual-semantic joint embedding. Finally, we can see that *Ours (Full)* performs slightly better than both *Ours (CL)* and *Ours (SL)*. Hence, we can conclude that our autoencoder and visual-semantic embedding collaboratively provide *complementary* training guidance to the proposed encoder.

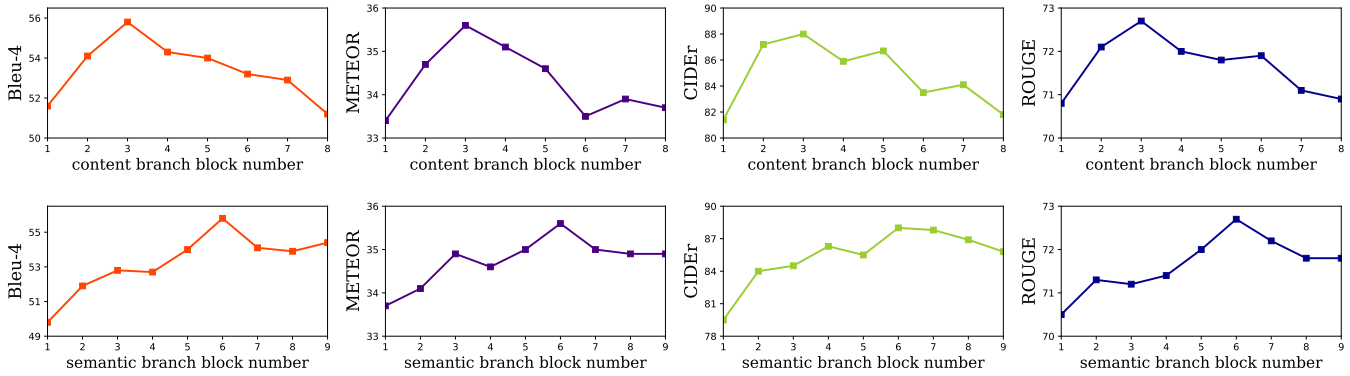


Fig. 7: Evaluation of the impact of both branches' depths on the performance of our model. First row: impact of the TCB block number in content branch, where the TCB number in semantic branch is fixed to 6. Second row: impact of the TCB block number in semantic branch, where the TCB number in content branch is fixed to 3.

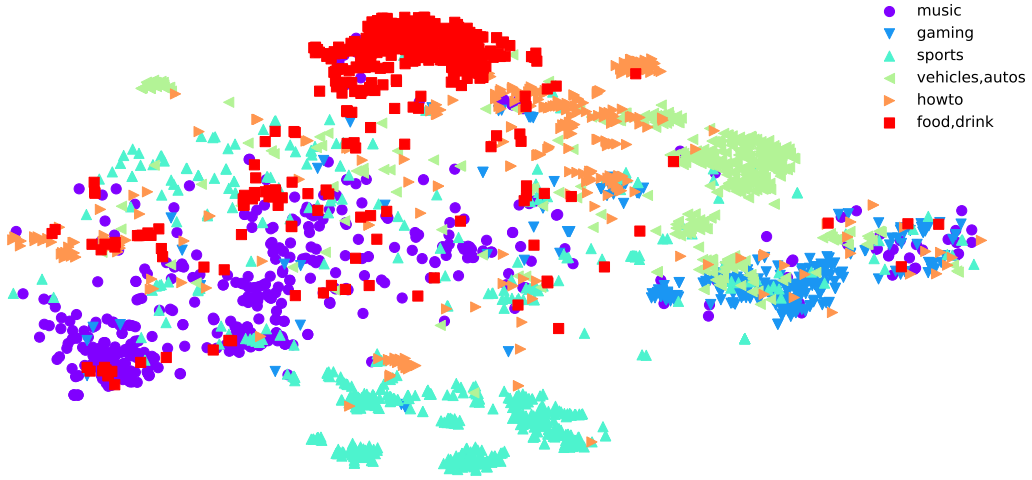


Fig. 8: A t-SNE visualization of video embeddings generated by our visual-semantic joint embedding model. Videos belonging to the same category have the same kind of maker. The category label information is provided by MSR-VTT dataset.

We present a qualitative comparison between variants of our model, *i.e.*, SibNet, in Figure 10. In Figure 10, "`Single`" ,"`DL`" and "`Ours`" denote captions generated by variants named *Single (DL-3)*, *Ours (DL)* and *Ours (Full)*, respectively. "`GT`" denotes ground truth captions. We highlight both *incorrect* (shown in blue) and *correct* (shown in red) words or phrases in the generated captions.

As we can see, captions generated by our full model is better aligned with ground truth captions than captions generated by its variants. In particular, our full model generates captions that 1) describe important semantic concepts, *e.g.*, the first example in the second and third rows of Figure 10, 2) contain detailed content information, *e.g.*, the two examples shown in the fifth row of Figure 10. This demonstrates the effectiveness of our proposed dual-branch architecture and training scheme which provides more fine-grained training supervision using autoencoder and visual-semantic joint embedding.

#### 4.3.3 *Why semantic branch is deeper than content branch?*

In this section, we discuss the impact of the depths of the two branches. We first increase the number of TCB blocks in the content branch from 1 to 8 while the number of blocks in the semantic branch to is fixed to 6. As shown in the first row of Figure 7, the performance drops in a monotonic manner as number of blocks in the content branch goes from 3 to 1 or from 3 to 8. We notice that when the number of blocks is 3, our model can achieve the best performance overall.

We also change the number of blocks in the semantic branch from 1 to 9 while fixing the number of blocks in the content branch as 3. The results are demonstrated in the second row of Figure 7. We can see that consistent performance drop exists when number of blocks in the semantic branch goes from 6 to 1 or from 6 to 9. In particular, using less than 3 blocks in the semantic branch severely affects the performance. This validates that in order to encode semantic information, which has a high level of abstraction, it is better to use deeper semantic branch. Another benefit for stacking more blocks is that, as the number of blocks in our semantic branch goes up, the temporal receptive field of it increases, which enables it to model longer temporal dynamics of videos. Based on the results shown in Figure 7, we empirically choose 3 TCBs to form the content branch and 6 TCBs to form the semantic branch.

TABLE 6: The number of parameters of SibNet and previous state-of-the-art methods.

| Methods | Parameters |
|---|---|
| STAT [78] | 36.2M |
| S2VT [3] | 26.4M |
| SCN [8] | 20.1M |
| M2M [15] | 14.9M |
| LSTM-TSA [9] | 12.8M |
| Ours | 9.9M |
| - $\text{CNN}_s$ | 0.4M |
| - $\text{CNN}_c$ | 0.3M |
| - $\text{RNN}_d$ | 9.2M |

#### 4.3.4 What does our semantic branch learn?

Figure 8 presents a t-SNE [89] visualization of video embeddings by performing visual-semantic joint embedding as used in our semantic branch. Each point in the figure represents the embedding of a video from MSR-VTT dataset. As we can see, the joint embedding model learns to "cluster" videos that belong to the same category together, *e.g.*, sports, food. This demonstrates that our semantic branch well captures video semantic feature, which can serve as complementary encoding for the content branch. By combining both branches, SibNet is able to encode videos effectively.

#### 4.3.5 Number of parameters

The number of parameters in SibNet and previous state-of-the-art methods are reported in Table 6. The reported numbers do not include parameters in the decoder's fully connected layer, whose output is then normalized by softmax function to generate probability distribution of words in the vocabulary. Because the number of parameters in it is proportional to the vocabulary size, it is not reported in most previous work. It can be seen from Table 6 that SibNet has much smaller number of parameters than previous state-of-the-art approaches ($27.3\%$ of [78], $38\%$ of [3], $50\%$ of [8], $69\%$ of [15] and $78\%$ of [9]). It is worth noting that the number of parameters in our encoder ($0.7M$) is less than $9\%$ of that of the RNN decoder ($9.2M$). Our encoder is able to achieve greater representation power with far less number of parameters than existing encoders employed by previous methods.

## 5 Conclusions

In this paper, we introduce a novel encoder with dual-branch architecture, *i.e.*, SibNet, for visual captioning. SibNet uses its content branch to encode salient visual content information with the help of autoencoder and the semantic branch to encode high-level semantic information with the guidance of ground truth captions brought by visual-semantic joint embedding. We train all major components of our model, *i.e.*, the content branch, the semantic branch and the decoder jointly by minimizing our proposed loss function, which incorporates three loss terms that push the three components to complement each other. We showcase the effectiveness of SibNet with extensive experiments on standard video captioning datasets. Our proposed SibNet outperforms previous state-of-the-art video captioning models by a large margin.

## 6 Acknowledgement

This work is supported in part by the start-up funds of University at Buffalo and gift grants from Snap Research.

## References

[1] R. Xu, C. Xiong, W. Chen, and J. J. Corso, "Jointly modeling deep video and compositional text to bridge vision and language in a unified framework." in *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, 2015.

[2] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," in *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2014.

[3] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4534–4542.

[4] S. Venugopalan, L. A. Hendricks, R. Mooney, and K. Saenko, "Improving lstm-based video description with linguistic knowledge mined from text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2016.

[5] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4651–4659.

[6] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang, "Hierarchical recurrent neural encoder for video representation with application to captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1029–1038.

[7] Z. Ren, X. Wang, N. Zhang, X. Lv, and L.-J. Li, "Deep reinforcement learning-based image captioning with embedding reward," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[8] Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, and L. Deng, "Semantic compositional networks for visual captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[9] Y. Pan, T. Yao, H. Li, and T. Mei, "Video captioning with transferred semantic attributes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[10] J. Xu, T. Yao, Y. Zhang, and T. Mei, "Learning multimodal attention lstm networks for video captioning," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 537–545.

[11] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 3104–3112.

[12] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[13] L. Zhou, C. Xu, P. Koch, and J. J. Corso, "Watch what you just said: Image captioning with text-conditional attention," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, 2017.

[14] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen, "Video captioning with attention-based lstm and semantic consistency," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2045–2055, 2017.

[15] R. Pasunuru and M. Bansal, "Multi-task video captioning with video and entailment generation," in *Proceedings of Association for Computational Linguistics (ACL)*. ACL, 2017.

[16] D. L. Chen and W. B. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL, 2011, pp. 190–200.

[17] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (ICCV)*, 2016, pp. 5288–5296.

[18] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013, pp. 1700–1709.

[19] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2014.

[20] O. Vinyals, Ł. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton, "Grammar as a foreign language," in *Advances in Neural Information Processing Systems*, 2015, pp. 2773–2781.

[21] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith, "Transition-based dependency parsing with stack long short-term memory," in *Proceedings of Association for Computational Linguistics (ACL)*, 2015.

[22] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

[23] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2016.

[24] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2017.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[26] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR*, 2015, pp. 3156–3164.

[27] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning (ICML)*, 2015, pp. 2048–2057.

[28] S. Tsutsui and D. Crandall, "Using artificial tokens to control languages for multilingual image caption generation," *arXiv preprint arXiv:1706.06275*, 2017.

[29] A. K. Vijayakumar, M. Cogswell, R. R. Selvaraju, Q. Sun, S. Lee, D. J. Crandall, and D. Batra, "Diverse beam search for improved description of complex scenes." in *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[33] J. Johnson, A. Karpathy, and L. Fei-Fei, "Densecap: Fully convolutional localization networks for dense captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4565–4574.

[34] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, "Jointly modeling embedding and translation to bridge video and language," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[35] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, "End-to-end dense video captioning with masked transformer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[36] B. Wang, L. Ma, W. Zhang, and W. Liu, "Reconstruction network for video captioning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[37] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[38] L. Zhu, Z. Xu, and Y. Yang, "Bidirectional multirate reconstruction for temporal modeling in videos," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[39] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4507–4515.

[40] S. Chen, J. Chen, Q. Jin, and A. Hauptmann, "Video captioning with guidance of multimodal latent topics," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 1838–1846.

[41] S. Chen, J. Chen, and Q. Jin, "Generating video descriptions with topic guidance," in *ACM International Conference on Multimedia Retrieval*. ACM, 2017.

[42] S. Chen, Q. Jin, J. Chen, and A. Hauptmann, "Generating video descriptions with latent topic guidance," *IEEE Transactions on Multimedia*, 2019.

[43] Y. Chen, S. Wang, W. Zhang, and Q. Huang, "Less is more: Picking informative frames for video captioning," in *European Conference on Computer Vision*, 2018.

[44] J. Wang, W. Wang, Y. Huang, L. Wang, and T. Tan, "M3: Multimodal memory modelling for video captioning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7512–7520.

[45] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," *Biological cybernetics*, vol. 59, no. 4-5, pp. 291–294, 1988.

[46] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length and helmholtz free energy," in *Advances in neural information processing systems*, 1994, pp. 3–10.

[47] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[48] Z. Ren, H. Jin, Z. Lin, C. Fang, and A. Yuille, "Joint image-text representation by gaussian visual-semantic embedding," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 207–211.

[49] ——, "Multiple instance visual-semantic embedding," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2017.

[50] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5005–5013.

[51] Z. Ren, "Joint image-text representation learning," Ph.D. dissertation, University of California, Los Angeles, 2016.

[52] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning," in *Proceedings of Association for Computational Linguistics (ACL)*. ACL, 2010, pp. 384–394.

[53] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," in *International Conference on Learning Representations (ICLR)*, 2017.

[54] D.-A. Huang, V. Ramanathan, D. Mahajan, L. Torresani, M. Paluri, L. Fei-Fei, and J. Carlos Niebles, "What makes a video a video: Analyzing temporal information in video understanding models and datasets," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*, 2018.

[55] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," *arXiv preprint arXiv:1612.08083*, 2016.

[56] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[57] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, "Microsoft coco captions: Data collection and evaluation server," *arXiv preprint arXiv:1504.00325*, 2015.

[58] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*. ACL, 2002, pp. 311–318.

[59] J. R. Crouse, J. S. Raichlen, W. A. Riley, G. W. Evans, M. K. Palmer, D. H. O'Leary, D. E. Grobbee, M. L. Bots, M. S. Group *et al.*, "Effect of rosuvastatin on progression of carotid intima-media thickness in low-risk individuals with subclinical atherosclerosis: the meteor trial," *Jama*, vol. 297, no. 12, pp. 1344–1353, 2007.

[60] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4566–4575.

[61] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," *Text Summarization Branches Out*, 2004.

[62] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 20–36.

[63] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, "Video paragraph captioning using hierarchical recurrent neural networks,"

in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4584–4593.
[64] Z. Shen, J. Li, Z. Su, M. Li, Y. Chen, Y.-G. Jiang, and X. Xue, "Weakly supervised dense video captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
[65] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4489–4497.
[66] B. Logan *et al.*, "Mel frequency cepstral coefficients for music modeling," in *ISMIR*, vol. 270, 2000, pp. 1–11.
[67] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[68] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.
[69] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *Journal of Machine Learning Research*, vol. 3, no. Aug, pp. 115–143, 2002.
[70] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2014.
[71] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.
[72] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
[73] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Advances in Neural Information Processing Systems (NIPS)*, 1992, pp. 950–957.
[74] Y. Bin, Y. Yang, F. Shen, X. Xu, and H. T. Shen, "Bidirectional long-short term memory for video description," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 436–440.
[75] L. Baraldi, C. Grana, and R. Cucchiara, "Hierarchical boundary-aware neural encoder for video captioning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 3185–3194.
[76] N. Ballas, L. Yao, C. Pal, and A. Courville, "Delving deeper into convolutional networks for learning video representations," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.
[77] Z. Guo, L. Gao, J. Song, X. Xu, J. Shao, and H. T. Shen, "Attention-based lstm with semantic consistency for videos captioning," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 357–361.
[78] Y. Tu, X. Zhang, B. Liu, and C. Yan, "Video description with spatial-temporal attention," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 1014–1022.
[79] Z. Yang, Y. Han, and Z. Wang, "Catching the temporal regions-of-interest for video captioning," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 146–153.
[80] X. Wu, G. Li, Q. Cao, Q. Ji, and L. Lin, "Interpretable video captioning via trajectory structured localization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6829–6837.
[81] Q. Jin, J. Chen, S. Chen, Y. Xiong, and A. Hauptmann, "Describing videos using multi-modal fusion," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 1087–1091.
[82] R. Shetty and J. Laaksonen, "Frame-and segment-level features and candidate pool evaluation for video caption generation," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 1073–1076.
[83] V. Ramanishka, A. Das, D. H. Park, S. Venugopalan, L. A. Hendricks, M. Rohrbach, and K. Saenko, "Multimodal video description," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 1092–1096.
[84] J. Dong, X. Li, W. Lan, Y. Huo, and C. G. Snoek, "Early embedding and late reranking for video captioning," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 1082–1086.
[85] O. Nina, W. Garcia, S. Clouse, and A. Yilmaz, "Mtle: A multitask learning encoder of visual feature representations for video and movie description," *arXiv preprint arXiv:1809.07257*, 2018.
[86] T. Mei, Y. Rui, X. Tian, and T. Yao, "Msr-vtt challenge," http://ms-multimedia-challenge.com/2017/challenge, 2017.
[87] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.
[88] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2015.
[89] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

**Sheng Liu** Sheng Liu is currently a Ph.D. student at University at Buffalo, State University of New York. Before that, he was a project officer at School of Electrical and Electronic Engineering (EEE), Nanyang Technological University (NTU) from 2017 to 2018. He received B.Eng. from the Special Class for the Gifted Young of Xi'an Jiaotong University (XJTU).

**Zhou Ren** Zhou Ren is currently a Research Lead/Senior Researcher at Wormpex AI Research. Before that, he was a Senior Research Scientist at Snap Inc. from 2016 to 2018. He received his Ph.D. from the Computer Science Department at University of California, Los Angeles and M.Eng. from Nanyang Technological University, Singapore, in 2016 and 2012 respectively. He is currently an Associate Editor of The Visual Computer Journal. His current research interests include computer vision, video analysis, multimedia mining, etc. Zhou Ren was the recipient of the 2016 Best Paper Award from IEEE Transactions on Multimedia and the runner-up winner of the NIPS 2017 Adversarial Attack and Defense Competition. He was also nominated to the CVPR 2017 Best Student Paper Award.

**Junsong Yuan** Dr. Junsong Yuan is currently an Associate Professor and Director of Visual Computing Lab at Department of Computer Science and Engineering (CSE), State University of New York at Buffalo, USA. Before that he was an Associate Professor at Nanyang Technological University (NTU), Singapore. He obtained his Ph.D. from Northwestern University, M.Eng. from National University of Singapore and B.Eng from the Special Program for the Gifted Young of Huazhong University of Science and Technology (HUST), China. His research interests include computer vision, pattern recognition, video analytics, gesture and action analysis, large-scale visual search and mining. He received Best Paper Award from IEEE Trans. on Multimedia, Nanyang Assistant Professorship from NTU, and Outstanding EECS Ph.D. Thesis award from Northwestern University. He is currently Senior Area Editor of Journal of Visual Communications and Image Representation (JVCI), Associate Editor of IEEE Trans. on Image Processing (T-IP) and IEEE Trans. on Circuits and Systems for Video Technology (T-CSVT), and served as Guest Editor of International Journal of Computer Vision (IJCV). He is Program Co-Chair of IEEE Conf. on Multimedia Expo (ICME'18) and Steering Committee Member of ICME (2018-2019). He also served as Area Chair for CVPR'20'19'17 and ACM MM'18 etc. He is a Fellow of International Association of Pattern Recognition (IAPR).

**S2VT** [1]: *a woman* is talking *about a movie*
**TA** [2]: *a woman* is talking
**Ours**: *a man and woman* are having a conversation
**GT**: a man and woman sitting in bed talking in a foreign language

**S2VT** [1]: *a girl* is dancing
**TA** [2]: *girls* are dancing
**Ours**: *cartoon characters* are dancing
**GT**: female cartoon characters are dancing


**S2VT** [1]: *a man* is playing *a game*
**TA** [2]: *a man* is playing *a game*
**Ours**: *two men* are playing *table tennis in a stadium*
**GT**: two men are playing table tennis

**S2VT** [1]: *a cartoon boat* is flying
**TA** [2]: a person is *playing a video game*
**Ours**: a person is *flying a helicopter*
**GT**: a helicopter is flying very closely to the ground fighting a war

**S2VT** [1]: *a cartoon of a video game*
**TA** [2]: *a video game*
**Ours**: *fireworks are being shown*
**GT**: some fireworks are being launched into the night sky


**S2VT** [1]: a man is playing *a man in a kitchen*
**TA** [2]: a man is talking about *a pan*
**Ours**: there is a man is talking about *the moon*
**GT**: a person is talking about moon

**S2VT** [1]: *a man is talking about a movie*
**TA** [2]: a group of people *are shown*
**Ours**: *soldiers* are *fighting in a war*
**GT**: soldiers are fighting each other in the battle

**S2VT** [1]: a man is *talking about a man*
**TA** [2]: a man is *talking about a phone*
**Ours**: a man is *shooting a gun*
**GT**: a man is about to shoot someone in forest


**S2VT** [1]: a woman is *cooking* food
**TA** [2]: a person is *cooking*
**Ours**: a person is *mixing* some food *in a bowl*
**GT**: a man is mixing something with other food item

**S2VT** [1]: *a man* is *talking* about the movie
**TA** [2]: *a man* is *talking*
**Ours**: *a man and a woman* are *fighting* each other
**GT**: the woman with muscular body hits the man down

Fig. 9: Qualitative comparison of our model, *i.e.*, SibNet, and previous state-of-the-art methods, including S2VT [3] and Temporal Attention [39]. "`S2VT`" ,"`TA`" and "`Ours`" denote captions generated by S2VT [3], Temporal Attention [39] and our model, respectively. "`GT`" denotes ground truth captions. We highlight both *incorrect* (shown in blue) and *correct* (shown in red) words or phrases in the generated captions.

**Single**: a person is ***playing a video game***
**Sib-DL**: ***people*** are riding motorcycles
**Ours**: ***a man*** is ***riding a motorcycle***
**GT**: a person wearing a red and black suit riding a motorcycle

**Single**: a person is ***making a dish***
**Sib-DL**: a person is ***cooking in a pan***
**Ours**: a person is ***mixing a egg in a bowl***
**GT**: in the kitchen a woman mixing an egg in a bowl

**Single**: people are ***playing soccer***
**Sib-DL**: people are ***playing sports***
**Ours**: two men are ***fighting on a field***
**GT**: a group of soccer players fighting on a field

**Single**: a man is ***talking*** on stage
**Sib-DL**: a man is ***talking***
**Ours**: a man is ***giving a speech***
**GT**: a man gives a speech about stem education

**Single**: ***a man is talking about something***
**Sib-DL**: ***a man is talking***
**Ours**: a group of people are ***walking in the snow***
**GT**: soldiers taking orders and marching in the cold

**Single**: a man is singing
**Sib-DL**: ***a woman*** is singing
**Ours**: ***a man*** is singing ***a song***
**GT**: a man singing a song in a stage

**Single**: a woman is talking
**Sib-DL**: ***a man*** is talking about the news
**Ours**: ***a woman*** is talking about ***the news***
**GT**: a woman telling a news story about a book release

**Single**: a woman is ***talking***
**Sib-DL**: a girl is ***sitting*** on a phone
**Ours**: a woman is ***using*** a phone
**GT**: a girl using her smartphone

**Single**: ***a video game is being played***
**Sib-DL**: two men are wrestling
**Ours**: two men are wrestling ***in a ring***
**GT**: two wrestlers are fighting in the ring

**Single**: a crowd of fireworks
**Sib-DL**: fireworks are being shown
**Ours**: fireworks are ***exploding in the sky***
**GT**: fireworks are going off

Fig. 10: Qualitative comparison between variants of our model, *i.e.*, SibNet. "`Single`" ,"`DL`" and "`Ours`" denote captions generated by variants named *Single (DL-3)*, *Ours (DL)* and *Ours (Full)* which are introduced in Section 4.3.2 respectively. "`GT`" denotes ground truth captions. We highlight both *incorrect* (shown in blue) and *correct* (shown in red) words or phrases in the generated captions.