Abstract—Automatically describing video content with text description is a challenging but important task, which has been attracting a lot of attention in computer vision community. Previous works mainly strive for the accuracy of the generated sentences, while ignoring the sentences diversity, which is inconsistent with human behavior. In this paper, we aim to caption each video with multiple descriptions and propose a novel framework. Concretely, for a given video, the intermediate latent variables of conventional encode-decode process are utilized as input to the conditional generative adversarial network (CGAN) with the purpose of generating diverse sentences. We adopt different Convolutional Neural Networks (CNNs) as our generator that produces descriptions conditioned on latent variables and discriminator that assesses the quality of generated sentences. Simultaneously, a novel DCE metric is designed to assess the diverse captions. We evaluate our method on the benchmark datasets, where it demonstrates its ability to generate diverse descriptions and achieves superior results against other state-of-the-art methods.

Index Terms—Diversity, caption, conditional generative adversarial network, CNNs.

I. INTRODUCTION

Video captioning has recently received increased interest and become an important task in computer vision. Most of the research efforts have been trying to generate captions for videos, including the template based methods and the neural network based models. In the template based methods, subjects, verbs and objects are firstly detected and then filled into a pre-defined template to generate the corresponding sentence. However, the sentences yielded by these methods are very rigid and limited. They are still far from being perfect.

Deep learning has rapidly developed, and significant advancements have been made in image classification using convolutional neural network (CNN) [1] and machine translation utilizing recurrent neural network (RNN) [2]. Benefit from these achievements, neural network based approaches have become the mainstream to generate descriptions for videos. They are mostly built upon the encoder-decoder framework and optimized in an end-to-end manner. In these approaches, a CNN is utilized as an encoder to generate the visual representation and a RNN as a decoder to generate sequence of words.

In view of the effectiveness of the encoder-decoder framework, we follow this elegant recipe in our work. However, previous research primarily focuses on the fidelity of sentences, while another essential property, diversity, is not taken into account. More specifically, most models are trained to select words with maximum probability, which results in a monotonous set of generated sentences since these sentences bear high resemblance to training data.

Towards the goal of building a model that can selectively output a single high-accuracy description or the diverse descriptions, we propose a novel approach on top of the encoder-decoder framework and the conditional GAN, as depicted in Fig. 1. We refer to our proposed method as DCM (Diverse Captioning Model), which consists of two components: (1) An attention-based LSTM model is trained to generate diverse sentences. Concretely, we employ a Bi-directional LSTM [3] to encode the video features. Following it, a temporal attention mechanism is utilized to make a soft-selection over them. Afterwards, we adopt the hierarchical LSTM to generate the descriptions. (2) A conditional GAN whose input is the latent variables of the attention-based model is trained to generate diverse sentences. Specifically, in CGAN, we get rid of LSTM and adopt the fully convolutional network as our generator to produce descriptions based on the latent variables. For the discriminator, both the sentences and video features are used as input to evaluate the quality of generated sentences. The former component is designed to effectively model video-sentence pairs, and the latter aims at simultaneously considering both the fidelity and diversity of the generated descriptions.

The contributions of this work can be summarized as follows: (1) A dual-stage framework that can generate both high-accuracy caption and diverse caption is proposed. Simultaneously, our diversity module is compatible with almost all models based on the encoder-decoder structure, as long as their decoders are simply extended to a hierarchical architecture. (2) We propose a novel performance evaluation metric named Diverse Captioning Evaluation (DCE), which considers not only the differences between sentences, but also the rationality of sentences (i.e., whether the video content is correctly described). (3) Extensive experiments and ablation studies on benchmark MSVD and MSR-VTT datasets demonstrate the superiority of our proposed model in comparison to the state-of-the-art methods.

II. RELATED WORK

A. RNN for Video Captioning

As a crucial challenge for visual content understanding, captioning tasks have attracted much attention for many years. Venugopalan et al. [4] transferred knowledge from image caption models via adopting the CNN as the encoder and LSTM as the decoder. Pan et al. [5] used the mean-pooling caption model with joint visual and sentence embedding.
Fig. 1. Illustration of the proposed method. Our model consists of an encoder-decoder architecture and a conditional GAN. The former aims to learn an effective coding method for video and text, and the latter is designed to generate diverse descriptions conditioned on the latent variables extracted from the encoder-decoder model. The design of hierarchical LSTM enables the encoder-decoder model to generate a single accurate sentence while also assisting the generator to generate diverse sentences. During testing, when we use LSTM2 to generate high-precision captions, the input of LSTM1 at each step is the previous generated word of LSTM2. And if we want to generate diverse descriptions, the input of LSTM1 is the previous generated word of generator. This allows our encoder-decoder model and generator to generate different sentences.

To better encode the temporal structures of video, Yao et al. [6] incorporated the local C3D features and a global temporal attention mechanism to select the most relevant temporal segments. More recently, a novel encoder-decoder-reconstruction network was proposed by [7] to utilize both the forward and backward flows for video captioning. Aafaq et al. [8] embedded temporal dynamics in visual features by hierarchically applying Short Fourier Transform to CNN features.

Multi-sentence description for videos has been explored in several works [9]–[12]. Shin et al. [9] temporally segmented the video with action localization and then generated multiple captions for those segments. A hierarchical model containing a sentence generator that produces short sentences and a paragraph generator that captures the inter-sentence dependencies was proposed by [10]. DVC [12] exploited the spatial region information and further explored the correspondence between sentences and region-sequences. Different from these methods which focus on temporally segmentation or spatial region information, MS-RNN [11] modeled the uncertainty observed in the data using latent stochastic variables. It can thereby generate multiple sentences with consideration of different random factors. In this paper we also pay attention to the latent variables, but we try to generate diverse captions from the video-level.

B. GAN for Natural Language Processing

Generative adversarial network (GAN) [13] has become increasingly popular and achieved promising advancements in generating continuous data and discrete data [14]. It introduces a competing process between a generative model and a discriminative model through a minimax game where the generative model is encouraged to produce highly imitated data and the discriminative model learns to distinguish them. To produce specific data, conditional GAN (CGAN) was first proposed by [15] to generate MNIST digits conditioned on class labels. Since then it has been widely applied to other fields, such as image synthesis and text generation. For instance, Reed et al. [16] used CGAN to generate images based on the text descriptions. Dai et al. [17] built a conditional sequence generative adversarial net to improve the image description.

Motivated by previous researches, in this paper we incorporate CGAN with encoder-decoder model to describe video
content. With this design, our proposed method is able to selectively output a single high-accuracy description or diverse descriptions.

III. THE PROPOSED METHOD

A. Single Captioning with Encoder-Decoder Framework

We propose in this paper a stacked LSTM based model for single video captioning, as shown in Fig. 2. We refer to it Ours-ED. Given a video, \( \mathbf{V} \), with \( N \) frames, the extracted visual features and the embedded textual features can be represented as \( \mathbf{v} = \{v_1, v_2, \ldots, v_N\} \) and \( \mathbf{w} = \{w_1, w_2, \ldots, w_T\} \), where \( v_i \in \mathbb{R}^{D_v \times 1} \), \( w_i \in \mathbb{R}^{D_w \times 1} \), and \( T \) is the sentence length. Specifically, \( D_v \) and \( D_w \) are the respective dimensions of the frame-level features and vocabulary. We use a bi-directional LSTM (BLSTM) which can capture both forward and backward temporal relationships to encode the extracted visual features. The activation vectors are obtained as:

\[
h_t = h_t^{(f)} + h_t^{(b)}
\]

where \( h_t^{(f)} \) and \( h_t^{(b)} \) are the forward and backward hidden activation vectors. Inspired by the recent success of attention mechanism in various tasks, a visual attention mechanism is incorporated following the BLSTM. To avoid imposing visual attention on non-visual words [18], we append a blank feature whose values are all zeros to the encoded video features. Thus, we can eliminate the impact of visual information if the predicting word is irrelevant to the high-level visual representations Accordingly, the output context vector at time step \( t \) can be represented as:

\[
a_t = \sum_{i=1}^{N+1} \alpha_{t,i} h_i
\]

In (2), \( h_{N+1} \) is the blank feature, and \( \alpha_{t,i} \) is the attention weight which can be computed as:

\[
\alpha_{t,i} = \text{softmax}(e_{t,i})
\]

\[
e_{t,i} = w^T \tanh(W_a h_i + V_a h_{t-1}^i + b_a)
\]

where \( w, W_a, V_a \) and \( b_a \) are the learned parameters, \( h_{t-1}^i \) is the hidden state of the decoder LSTM (LSTM2) at the \( (t-1) \)-th time step.

Our decoder integrates two LSTMs. LSTM1 is used to efficiently encode the previous words and outputs \( q_t \). Then LSTM2 generates the next word based on the concatenation of the visual information \( a_t \) and the textual information \( q_t \). With this hierarchical design, our model can not only generate a concise and accurate description, but also assist the diversity module to generate varied descriptions, which will be introduced in the following section. According to the above analysis, at time step \( t \), our model utilizes \( \mathbf{V} \) and the previous words \( w_{<t} \) to predict a word \( w_t \) with the maximal probability \( P(w_t|w_{<t}, \mathbf{V}) \), until we reach the end of the sentence. Thus, the loss function of our encoder-decoder model can be defined as:

\[
L_{xe} = - \sum_{t=1}^{T} \log P(w_t|w_{<t}, \mathbf{V}; \theta)
\]

where \( \theta \) is the model parameter set.

B. Diverse Captioning with Conditional GAN

As introduced above, we use the conditional GAN to implement the diverse captioning for video. In our framework, multiple CNNs are adopted as the generator \( G \) to generate descriptions, and the discriminator \( D \) trying to assess the sentences quality. They are optimized by a minimax two-player game until they reach an equilibrium.

Concretely, our generator strives to produce high-quality sentences conditioned on the random vector \( \mathbf{z} \) sampled from the normal distribution \( \mathcal{N}(0,1) \) and the latent variables \( \mathbf{L} \) obtained from the encoder-decoder model, where \( \mathbf{z} = \{z_1, z_2, \ldots, z_T\} \) and \( \mathbf{L} = \{L_1, L_2, \ldots, L_T\} \). Among them, \( L_t \) consists of \( \bar{a}_t \) and \( q_t \), and \( \bar{a}_t \) is calculated in the same way as \( a_t \), except that the output of the CNN is used as input. At each step \( t \), suppose \( s_t \) is the concatenation of \( z_t \) and \( L_t \), and \( S_{1:t} = \{s_1, s_2, \ldots, s_t\} \) is the concatenation of \( s_t \in \mathbb{R}^{k \times \bar{k}} \) takes input \( S_{1:t} \) and produces a context vector \( C_t \) that contains the information of all the previously generated words. Afterwards, a MLP layer is utilized to encode \( C_t \) and yields a conditional distribution \( P(w_t|z_t, L_{1:t}) \) over the vocabulary. We choose CNN since it is being penalized for producing less-peaky word probability distributions, giving it the potential to explore the sentence diversity [19]. Overall, the random vector allows the generator to generate diverse descriptions and the latent variable guarantees the fidelity of the generated sentences. Specifically, when the discriminator is fixed, the generator can be optimized by minimizing the following formulation:

\[
\min_{\phi} \mathbb{E}_{z \sim \mathcal{N}}[\log(1 - D(G_{\phi}(\mathbf{z}, \mathbf{L})))]
\]

here, \( \phi \) represents the parameter set of generator \( G \). Considering both the accuracy and diversity of the generated sentences, we balance the generator with an extra cross entropy loss, which also prevents our conditional GAN from deviating from
its correct orbit during training. Therefore, we minimize the objective function $L_G$ as follows when updating $G$:

$$L_G = -\lambda \mathbb{E}[\log P( \mathbf{w} | \mathbf{z}, \mathbf{L})] + (1 - \lambda) \mathbb{E}_{z \sim \mathcal{N}}[\log(1 - D(\mathbf{G}(\mathbf{z}, \mathbf{L})))].$$ (7)

where $\lambda$ is the tradeoff parameter.

In the discriminator $D$, our goal is to judge whether the given video-text pair as matched or not and how well a sentence describes a given video. Inspired by the superior performance of convolutional neural network in text classification, we also choose CNN as our discriminator. Suppose the embedded matrix of the generated sentence is represented as $X_{1:T} \in \mathbb{R}^{T \times k} = (x^1, x^2, ..., x^T)$ by concatenating the word embeddings as columns, where $k$ is the dimension of the word embedding. Then, a convolutional operation with a kernel $K^d \in \mathbb{R}^{l \times k}$ is used to encode the sentence and produce a feature map:

$$f_i = g(K^d * X_{i:i+l-1} + b)$$ (8)

here, * is the convolution operator, $b$ is the bias term and $g(\cdot)$ is the non-linear function (we use ReLU in our experiments). A max-over-time pooling operation is then applied over the generated feature maps, $f = \max \{ f_1, f_2, ..., f_{T-l+1} \}$. Moreover, we use different kernels to extract different features of sentence and combine them as our final sentence representation.

Once the above operations are completed, we concatenate the sentence representation with its corresponding video feature $h_N$ extracted from the hidden layer of BLSTM. Then, a fully connected layer is utilized to map this concatenated feature $H$ to a low-dimensional space and a softmax layer is incorporated to output the probability that indicates the quality of sentence. Formally, this process can be described as:

$$D(X) = \text{softmax}(W_P H + b_P)$$ (9)

where $W_P$ and $b_P$ are parameters to be learned, $D(X)$ is projected onto the range of $[0, 1]$. An output close to 1 indicates a bigger probability that $X$ is drawn from the real data distribution or not. The optimization target of $D$ is to maximize the probability of correctly distinguishing the ground truth from the generated sentences. For $G$ fixed, $D$ can be optimized as:

$$\max_{\eta} \mathbb{E}_{x \sim P_{data}}[\log(D_{\eta}(x))] + \mathbb{E}_{z \sim \mathcal{N}}[\log(1 - D_{\eta}(\mathbf{G}(\mathbf{z}, \mathbf{L})))].$$ (10)

where $\eta$ is the parameter set of discriminator $D$, $x$ represents the video-sentence pair in the training set $P_{data}$.

Unlike a typical GAN setting, in which the generator receives rewards at each intermediate step, our captioning model only receives the reward at the end (the reward signal is meaningful only for the completed sentence), which may lead to several difficulties in training such as vanishing gradients and error propagation. To mitigate the lack of intermediate rewards, the Monte Carlo rollouts [20] is employed to provide early feedback. Moreover, we model the captioning process via policy gradient [21] to overcome the problem that gradients cannot be back-propagated directly since the text generation procedure is non-differentiable. Specifically, following [21], when there is no intermediate reward, the objective of the generator (when optimizing the adversarial loss) is to generate a sequence $W_{1:T} = (w_1, w_2, ..., w_T)$ from the start state $s_0$ to maximize its expected end reward:

$$\mathbb{E}[R_T | s_0, \phi] = \sum_{w_1 \in \gamma} G_{\phi}(w_1 | s_0) \cdot Q_{G_{\phi}}(s_0, w_1)$$ (11)

where $s_0$ is the start state, $R_T$ is the reward for a complete sequence, $\gamma$ is the vocabulary, $G_{\phi}$ is the generator policy which influences the action that generates the next word, and $Q_{G_{\phi}}(s, a)$ indicates the action-value function of a sequence (i.e. the expected accumulative reward starting from state $s$, taking action $a$, and then following policy $G_{\phi}$). In our experiments we use the estimated probability of being real by the discriminator $D_{\eta}(W_{1:T}, \hat{v})$ as the reward. Thus, we have:

$$Q_{G_{\phi}}(s = W_{1:T-1}, a = w_T) = D_{\eta}(W_{1:T}, \hat{v})$$ (12)

Here, $\hat{v}$ indicates the corresponding visual feature. To evaluate the action-value for an intermediate state, the Monte Carlo rollouts is utilized to sample the unknown last $T - t$ tokens. To reduce the variance, we run the rollout policy starting from current state till the end of the sequence for $K$ times. Thus, we have:

$$Q_{G_{\phi}}(s = W_{1:t-1}, a = w_t) =$$

$$\left\{ \begin{array}{ll}
\frac{1}{K} \sum_{n=1}^{K} D_{\eta}(W_{1:T}^n, \hat{v}), & W_{1:T}^n \in \text{MC}(W_{1:t}; K) \quad \text{for } t < T \\
D_{\eta}(W_{1:t}, \hat{v}) & \text{for } t = T
\end{array} \right. \quad \text{(13)}$$

### Algorithm 1: Training process of DCM

**Require:** encoder-decoder model $\pi_{\theta}$; generator $G_{\phi}$; discriminator $D_{\eta}$; training data set $P_{data}$

1. Initialize $\pi_{\theta}$, $G_{\phi}$ and $D_{\eta}$ with random weights $\theta$, $\phi$ and $\eta$;
2. Pre-train $\pi_{\theta}$ on $P_{data}$ by Eq. (5);
3. Generate negative samples in three ways mentioned in section III-B;
4. Pre-train $D_{\eta}$ via minimizing the cross entropy;
5. repeat
6. for g-step do
7. Generate a sequence $W_{1:T} = (w_1, w_2, ..., w_T)$ using $G_{\phi}$;
8. for $t$ in $1:T$ do
9. Compute $Q(s = W_{1:t-1}, a = w_t)$ by Eq. (13);
10. end
11. Update generator parameters by Eq. (7);
12. end
13. for d-steps do
14. Use current $G_{\phi}$ to generate negative examples and combine with given positive examples of $P_{data}$;
15. Train discriminator $D_{\eta}$ for $p$ epochs by Eq. (10);
16. end
17 until DCM converges;
where $W^n_{1:t} = (w^n_1, ..., w^n_t)$ and $W^{n+1}_{t+1:T}$ is sampled based on the rollout policy and the current state. In summary, Algorithm 1 shows full details of the proposed DCM. Aiming at reducing the instability in training process, we pre-train our encoder-decoder model and discriminator aforementioned to have a warm start. When pre-training our discriminator, the positive examples are from the given dataset, whereas the negative examples consist of two parts. One part is generated from our generator, and the other part is manually configured. Concretely, mismatched video-sentence pairs are utilized as one of the negative examples (model the inter-sentence relationship). Meanwhile, with the purpose of evaluating sentences more accurately, we exchange the word positions of the sentences in the positive examples and regard them as negative examples (model the intra-sentence relationship). The objective function $L_D$ for pre-training can be formalized into a cross entropy loss as follow:

$$L_D = -\frac{1}{m} \sum_{i=1}^{m} [Y_i \log(D(w_i)) + (1 - Y_i) \log(1 - D(w_i))]$$

where $Y_i$ and $D(w_i)$ denote the real label and the predicted value of discriminator respectively, and $m$ is the number of examples in a batch. It is worth noting that during testing, when we use LSTM2 to generate high-precision captions, the input of LSTM1 at each step is the previous generated word of LSTM2. And if we want to generate diverse descriptions, the input of LSTM1 is the previous generated word of generator.

### IV. Experiments

#### A. Datasets

Two benchmark public datasets including MSVD [31] and MSR-VTT [22] are employed to evaluate the proposed diverse captioning model. The MSVD dataset consists of 1,970 video clips collected from YouTube. It covers a lot of topics and is well-suited for training and evaluating a video captioning model. We adopt the same data splits as provided in [22] with 1,200 videos for training, 100 videos for validation and 670 videos for testing. As for MSR-VTT, there are 10K video clips, each of which contains 20 reference sentences annotated by human. We follow the public split method: 6,513 videos for training, 497 videos for validation, and 2,990 videos for testing.

#### B. Experimental Settings

We uniformly sample 60 frames from each clip and use Inception-v3 [33] to extract frame-level features. To capture the video temporal information, the C3D network [34] is utilized to extract the dynamic features of video. The dynamic features are then encoded by a LSTM whose final output is concatenated with all the feature-level features. For MSR-VTT, we integrate the audio features extracted from the pre-trained VGGish model [35] with other video features. We convert all the sentences to lower cases, remove punctuation characters and tokenize the sentences. We retain all the words in the dataset and thus obtain a vocabulary of 13,375 words for MSVD, 29,040 words for MSR-VTT. To evaluate the performance of our model, we utilize METEOR [36], BLEU [37], CIDEr [38], and ROUGE-L [39] as our evaluation metrics, which are commonly used for performance evaluation of video captioning methods.

In our experiments, with an initial learning rate $10^{-5}$ to avoid the gradient explosion, the LSTM unit size and word embedding size are set as 512, empirically. We train our model with mini-batch 64 using ADAM optimizer [40], and the length of sentence $T$ is set as 25. For sentence with fewer than 25 words, we pad the remaining inputs with zeros. To regularize the training and avoid overfitting, we apply dropout with rate of 0.5 on the outputs of LSTMs. In the training process, we first pre-train the encoder-decoder model and discriminator, and then optimize the CGAN module. When training CGAN, we freeze the parameters of the encoder-decoder model. During testing, beam search with beam width of 5 is used to generate descriptions.

#### C. Comparison with State-of-the-Art Models

**1) Quantitative Analysis: Single Caption Evaluation.** Table I demonstrates the result comparison among our proposed method and some state-of-the-art models. The comparing algorithms include the encoder-decoder based architectures (S2VT [22], LSTM-E [5], LSTM-TSA [24], MS-RNN [11], GRU-EVE [8]), and the attention based methods (h-RNN [10], HRNE [23], aLSTMs [25], hLSTMat [18], HMM [26], Rec-Net [7]). For MSVD dataset, the results of Ours-ED indicate that our method outperforms previous video captioning models.

### Table I

Comparison with state-of-the-art methods on MSVD dataset. (-) indicates such metric is unreported.

| Method   | METEOR | BLEU-4 | CIDEr | ROUGE-L |
|----------|--------|--------|-------|---------|
| S2VT     | 29.8   | -      | -     | -       |
| LSTM-E   | 31.0   | 45.3   | -     | -       |
| h-RNN    | 32.6   | 49.9   | -     | -       |
| HRNE     | 33.9   | 46.7   | -     | -       |
| LSTM-TSA | 33.5   | 52.8   | 74.0  | -       |
| aLSTMs   | 33.3   | 50.8   | 74.8  | -       |
| MS-RNN   | 33.8   | 53.3   | 74.8  | 70.2    |
| hLSTMat  | 33.6   | 53.0   | 73.8  | -       |
| RecNet   | 34.1   | 52.3   | 80.3  | 69.8    |
| HMM      | 33.8   | 52.9   | 74.5  | -       |
| GRU-EVE  | 35.0   | 47.9   | 78.1  | 71.5    |
| Ours-ED  | 35.6   | 53.3   | 83.1  | 71.2    |

### Table II

Comparison with state-of-the-art methods on MSR-VTT dataset. (-) indicates such metric is unreported.

| Method      | METEOR | BLEU-4 | CIDEr | ROUGE-L |
|-------------|--------|--------|-------|---------|
| LSTM-GAN    | 26.1   | 36.0   | -     | -       |
| aLSTMs      | 26.1   | 38.0   | 43.2  | -       |
| MS-RNN      | 26.1   | 38.8   | 40.9  | 59.3    |
| hLSTMat     | 26.3   | 38.3   | -     | -       |
| RecNet      | 26.6   | 39.1   | 42.7  | -       |
| DVC         | 28.3   | 41.4   | 48.9  | 61.1    |
| MARN        | 28.1   | 40.4   | 47.1  | 60.7    |
| OA-RTG      | 28.2   | 41.4   | 46.9  | -       |
| CVC-PSG     | 28.2   | 42.0   | 48.7  | 61.6    |
| Ours-ED     | 28.7   | 43.4   | 47.2  | 61.6    |
on most metrics. In particular, compared with the best counterpart (i.e., GRU-EVE), Ours-ED achieves better performance, with 0.6%, 5.4%, and 5.0% increases on METEOR, BLEU-4, and CIDEr respectively, verifying the superiority of our proposed approach.

For MSR-VTT dataset, we also compare our models with some recent state-of-the-art methods [28–30]. MARN [28] designs a memory structure to explore the full-spectrum correspondence between a word and its various similar visual contexts across videos. OA-BTG [29] captures detailed temporal dynamics for salient objects and learns discriminative spatio-temporal representations based on object-aware aggregation with bidirectional temporal graph. CVC-PSG [30] guides the video caption generation with Part-of-Speech information and constructs a novel gated fusion network to fuse different types of video representations. The results are reported in Table II. Note that DVC adopts data augmentation during training to obtain better accuracy. Nevertheless, Ours-ED also achieves the best performance on METEOR, BLEU-4, and ROUGE-L. It demonstrates the effectiveness of our proposed method.

**Diversity Evaluation.** To evaluate the caption diversity quantitatively, [12] pay attention to its opposite - the similarity of captions. They use latent semantic analysis (LSA) [42] to represent sentence by first generating sentence bag-of-words representation and then mapping it to LSA space. However, this method suffers important limitations. On one hand, it ignores the rationality of sentences. Two very different sentences can get high diversity score using this method, but their descriptions may be wrong. The generated sentences should be diverse on the premise of correctly describing the video content. On the other hand, this LSA method cannot capture polysemy and ignores the order of words in sentences, which limits its representation of sentence structure. Therefore, we propose in this subsection a diverse captioning evaluation (DCE) metric. It is mainly designed from two perspectives: the difference between sentences and the sentences reasonableness. About the difference of sentences, different from [12], we use the Jaccard similarity coefficient [43] which is effective in dealing with discrete data to model the word-level similarity, and use BERT [44] which shows superior performance in various NLP tasks and addresses the limitations of bag-of-words model to generate sentence-level representation. Meanwhile, we use METEOR as an assessment of sentences reasonableness since it has shown better correlation with human judgment compared with other metrics. Formally, the DCE can be calculated as:

\[
DCE = \frac{1}{m_v} \sum_{k=1}^{m_v} \sum_{n} 1 \cdot \sum_{s', s \in S, i \neq j} [M(s') + M(s)] \cdot [\delta(1 - J(s', s^j)) + (1 - \delta)(1 - <BT(s'), BT(s^j)>)]
\]

where S is the sentence set with cardinality n (i.e. if each video is accompanied with 5 generated sentences, n will be 10 because each sentence is used to calculate the similarities with others), \(m_v\) is the number of videos in the dataset, \(\delta\) is the adjustment coefficient (we set \(\delta = 0.5\) in our experiments), \(M(s)\) is the METEOR score of candidate sentence s, \(BT(s)\) is the sentence vector encoded by BERT. J and < denote the Jaccard similarity and cosine similarity.

We evaluate the diversity degree of our generated captions on the test set of two benchmark datasets. In addition to DCE, we also use the following methods: (1) **DCE-Variant.** The proposed DCE metric combines both accuracy and diversity. To interpret the role of each part more clearly, we evaluate the captions with the variant of DCE that removes the METEOR score. Thus we can see how much of the score comes from the sentence diversity. (2) **Word Types.** We measure the number of word types in the generated sentences. This reflects from the side whether the generated sentences are diverse. In general, higher word types indicates higher diversity. (3) **Mixed mBLEU.** Given a set of captions, BLEU score is computed between each caption against the rest. Mean of these BLEU scores is the mBLEU score. Following [45], we consider the mixed mBLEU score which represents the mean of mBLEU-n, where n is the n-gram. A lower mixed mBLEU value indicates higher diversity. (4) **Self-CIDEr.** To measure the diversity of image captions, Self-CIDEr [45] is designed to consider phrases and sentence structures. It shows better correlation with human judgement than the LSA-based metric. Higher is more diverse.

The results are shown in Table III, from which we observe that our DCM outperforms MS-RNN on all the evaluation metrics, showing the superiority of our proposed method.

**TABLE III**

| Dataset | MSVD | MSR-VTT |
|---------|------|---------|
| Model   | DCE  | DCE-Variant | Word Types | Mixed mBLEU | Self-CIDEr | DCE  | DCE-Variant | Word Types | Mixed mBLEU | Self-CIDEr |
|---------|------|-------------|-------------|-------------|-------------|------|-------------|-------------|-------------|-------------|
| MS-RNN  | 0.09 | 0.12        | 239         | 0.91        | 0.26        | 0.13 | 0.21        | 620         | 0.81        | 0.48        |
| DCM     | 0.25 | 0.37        | 2210        | 0.64        | 0.79        | 0.21 | 0.39        | 6492        | 0.68        | 0.81        |
| AG-CVAE | 0.18 |            |             |             |             |      |             |             |             |             |
| DVC     | 0.19 |            |             |             |             |      |             |             |             |             |
| DCM     | 0.21 |            |             |             |             |      |             |             |             |             |

**TABLE IV**

| Model   | DCE-Variant | DCE | Human Evaluation |
|---------|-------------|-----|------------------|
| Ours-ED | 0.30        | 0.19| 3.14             |
| DCM     | 0.39        | 0.26| 3.92             |
| GT      | 0.53        | 1.05| 5.96             |

**TABLE V**

| Model   | DCE |
|---------|-----|
| AG-CVAE | 0.18|
| DVC     | 0.19|
| DCM     | 0.21|

**REFERENCES:****

[11] JOURNAL OF L

**ISSUES:**

[14] 6
Fig. 3. Examples of video captioning results. The results of MS-RNN and DVC are reported in [11] and [12].

the same time, it also proves the effectiveness of DCE which performs consistent with other metrics (e.g. Self-CIDEr). We observe that the values of our DCE are lower than DCE-Variant. This is because DCE penalizes the inaccurate descriptions. Different from other existing metrics, DCE is the first one that considers both diversity and accuracy of captions, which is vital for video captioning task. Because the diversity of captions is meaningful only when the video content is accurately described. At the same time, DCE evaluates the diversity from the word level and sentence level, and uses the advanced model BERT in NLP field to encode sentences, so the results are more accurate and reasonable.

To further verify the rationality of DCE, we also conduct human evaluation by inviting volunteers to assess the captions obtained by Ours-ED, DCM and Ground Truth on the test set of MSR-VTT. Note that Ours-ED also can generate multiple descriptions using beam search. However, with beam search, Ours-ED can only generate up to n different sentences for the same model, while DCM can generate more, even several times Ours-ED. We randomly select 100 videos and sample 5 sentences from each model per video since Ours-ED has at most 5 different sentences. Each video is evaluated by 5 volunteers, and the average score is obtained. The given score (from 1 to 7) reflects the diversity of the set of captions. The higher the score, the better the diversity. From Table IV we can see that the DCE evaluation is consistent with the human evaluation, showing the advantage of DCE and proving that DCM is better than Ours-ED in diversity. Besides, compared with DCE-Variant, we notice that the gap between DCM and Ground Truth is larger under DCE evaluation. This indicates DCE takes both accuracy and diversity into account. Our DCE can be further improved if there is a better accuracy assessment method.

We also compare our method with DVC [12] and AG-CVAE [41] on the test set of MSR-VTT in Table V. AG-CVAE proposes a simple novel conditioning mechanism with an additive Gaussian prior to structure the latent space in a way that can directly reflect object co-occurrence. We observe that our DCM achieves the best performance in terms of DCE (the LSA score reported in [12] is 0.501, and we also surpass it (Ours is 0.53)). In addition, observing the fact that some of the expressions generated by our model do not even appear in the reference sentences, we believe that it reflects the potential of our model to enrich the annotation sentences in the database, which is of great significance.

2) Qualitative Analysis: In our experiments, a beam search with size 5 is used to generate descriptions. Thus, we can obtain 5 sentences with each input z. We run the testing 10 times and remove duplicate sentences. To gain an intuition of the improvement on generated diverse descriptions of our DCM, we present some video examples with the video description from MS-RNN [11] and DVC [12] as comparison to our system in Fig. 3. We can see that our DCM performs better than MS-RNN and DVC in generating diverse sentences. Meanwhile, compared with these models, DCM generates more accurate and detailed descriptions for diverse video topics. For example, in the last video, DVC simply produces “a man is hitting a ping pong ball in a stadium”, while our model can generate sentence like “two men compete in a game of table tennis by vigorously volleying the ping pong ball over the net”, which shows the superiority of our method. At the same time, our model correctly describes “two players” or “two men”. Conditioned on latent variables, our DCM can generate diverse descriptions while maintaining the semantic relevance. In general, compared with these two models, the sentences generated by DCM are superior in terms of video content description and sentence style generation.

To better understand our proposed method, in Fig. 4 we show some captioning examples generated by our model. From the generated results we can see that both Ours-ED and DCM are able to capture the core video subjects. Surprisingly, for a same model, in addition to describing video content from different aspects, DCM can generate sentences in different voices. For example, for the top video in the right column, DCM generates “a man is being interviewed on a news show” and “a man is giving a news interview”. The former describes the video content in a passive voice, while the latter uses the active voice, which is a significant improvement over previous research works [10]–[12] that only produce a single voice. Combining the characteristic of CGAN and the design
of fully convolutional network, our model achieves superior performance in generating diverse descriptions. Although our model achieves satisfactory results, improvements can be made. Fig. 5 shows two failure cases, where our DCM approach fails to describe “kayak” and “badminton”. For the first video, neither Ours-ED nor DCM can accurately describe “kayak”. This is due to the reason that our training dataset does not provide adequate training samples to distinguish this object. This indicates that to date, existing video captioning training datasets are still incomplete and require further refinement. For the second video, Ours-ED accurately describes “badminton”, while DCM fails. This is because when generating a description, in addition to relying on the latent variables extended from Ours-ED, DCM also relies on the generator of full convolution structure, which is less accurate than LSTM structure in generating natural language.
reported in Fig. 6, from which we notice that when \( \lambda \) varies from 0.1 to 0.9 on the MSR-VTT dataset. The results are shown in Table VI, where our model achieves the highest DCE score. Thus we set \( \lambda = 0.6 \) in the following experiments.

(2) **Effect of Rollout Times** \( K \). Obviously, the larger \( K \) is, the more time it costs. A suitable \( K \) should improve the performance of the model without costing too much time. We increase the rollout times \( K \) from 1 to 9 at intervals of 2 on MSR-VTT. Here we randomly select 10 sentences for each video and evaluate the diversity. As shown in Fig. 7, while we have \( K = 5 \), our model achieves the best performance on DCE. Therefore, we adopt \( K = 5 \).

(3) **Effect of \( \delta \)**. To investigate how \( \delta \) affects DCE score, several ablation studies are conducted on MSR-VTT. Using Jaccard similarity coefficient for word-level diversity evaluation is a “hard” method, while using BERT for sentence-level evaluation is relatively “soft”. For instance, for the sentences “a woman is cooking” and “a woman is showing how to cook a dish”, their diversity (except accuracy) at the word level is 0.7, while the diversity at the sentence level is 0.12, which proves that the diversity evaluation at the sentence level is more stringent. As can be seen from Fig. 8, the larger the \( \delta \), the greater the proportion of word-level evaluation, so the higher the DCE score.

4) **Effect of Different Architectures of Generator**. To design a better diverse captioning model, we also conduct some ablation studies on different architectures of generator, as shown in Fig. 9. The details are as follow:

**Single LSTM Layer**. Here a single LSTM is utilized as our generator to generate diverse captions. At each time step \( t \), it directly takes \( z_t \) and \( L_t \) as input and produces the context vector \( C_t \).

**Single CNN Layer**. Here a single CNN is utilized as our generator. At each time step \( t \), suppose \( s_t \) is the concatenation of \( z_t \) and \( L_t \), and \( E_t \in \mathbb{R}^{T \times k} \) is the concatenation of \( s_{\leq t} \) and \( (T - t) \text{ pad} \), where \( \text{pad} \) is the zero vector with the same dimension as the \( s_t \). Afterwards, the CNN with kernel \( K^g_c \in \mathbb{R}^{T \times k} \) takes input \( E_t \) and produces the context vector \( C_t \).

**Single LSTM layer + Single CNN layer**. Here we combine a LSTM and a CNN as our generator. At each time step \( t \), the LSTM takes input \( z_t \) and \( L_t \), and outputs \( q_t \) which is then combined with all the previous outputs \( q_{<t} \) to produce a context vector \( Q_t \), where \( Q_t = \{q_1, \ldots, q_t, \text{pad}, \ldots, \text{pad} \} \) and \( \text{pad} \) is the zero vector with the same dimension \( k \) as the LSTM output. Afterwards, a CNN with kernel \( K^g_c \in \mathbb{R}^{T \times k} \) is utilized to encode \( Q_t \) and yields \( C_t \).

**Single LSTM layer + Multiple CNN layers**. Here we combine a LSTM and multiple CNNs as our generator. At each time step \( t \), the LSTM takes input \( z_t \) and \( L_t \) as input and outputs \( o_t \) which is then combined with all the previous outputs \( o_{<t} \) to produce a context vector \( O_t = (o_1, o_2, \ldots, o_t) \). Afterwards, a CNN, with kernel \( K^g_c \in \mathbb{R}^{T \times k} \) is utilized to encode \( O_t \) and produces \( C_t \).

---

**Table VI**

| Model                       | Mixed mBLEU | Self-CIDEr | DCE-Variant | DCE |
|-----------------------------|-------------|------------|-------------|-----|
| Single LSTM Layer           | 0.831       | 0.565      | 0.297       | 0.188 |
| Single CNN Layer            | 0.772       | 0.744      | 0.343       | 0.192 |
| Single LSTM Layer + Single CNN Layer | 0.830     | 0.673      | 0.304       | 0.194 |
| Single LSTM Layer + Multiple CNN Layers | 0.792    | 0.709      | 0.336       | 0.202 |
| Multiple CNN Layers         | **0.687**   | **0.803**  | **0.387**   | **0.210** |

---

**Fig. 6. Effect of \( \lambda \)**

**Fig. 7. Effect of \( K \)**

**Fig. 8. Effect of \( \delta \)**
Multiple CNN Layers. This is our adopted method introduced above in the manuscript.

For each method we select the top 10 different sentences for each video and evaluate the diversity using Mixed mBLEU, Self-CIDEr, DCE-Variant and DCE. The results are presented in Table VI, showing that the architecture of multiple CNN layers performs best on all evaluation metrics. In particular, multiple CNN layers and single CNN layer achieve the top 2 performance in terms of Mixed mBLEU, Self-CIDEr and DCE-Variant, which demonstrates the potential of CNN architecture in generating diverse sentences. LSTM is good at producing peaky word probability distributions at the output [19]. This is conducive to generating high-precision descriptions, but limits the ability to generate diverse sentences. Compared to LSTM, CNN has the characteristic of generating lower peaky word probability distributions, giving it the potential to produce more word combinations. Benefit from the characteristics of CGAN, the diversity of single CNN architecture has been further improved, making it not only superior to the LSTM architecture, but also better than the combined architectures of LSTM and CNN on all metrics except DCE. Note that although the architectures of single LSTM layer + single CNN layer and single LSTM layer + multiple CNN layers are inferior to single CNN layer in terms of sentence diversity due to the LSTM limitations, they achieve better performance on DCE metric since LSTM makes the generated sentences more accurate, and DCE takes accuracy into account. For the last architecture, multiple CNNs are used for decoding. Each of them is responsible for predicting the next word based on a fixed length of input. Therefore, the division of labor of each CNN is very clear, and the information of all the previously generated words can be explicitly utilized when generating the next word, which is deficient in LSTM and single CNN. Compared to a single CNN, the architecture of multiple CNNs has advantages in generating high-precision sentences. In sum, it has a good performance in terms of both sentence diversity and accuracy. That is why it achieves the best results on all metrics. Thus, we adopt this multiple CNNs as our final architecture of generator.

V. CONCLUSION

In this work, a diverse captioning model DCM is proposed for video captioning, which simultaneously considers the fidelity and the diversity of descriptions. We obtain the latent variables that contain rich visual information and textual information from the well-designed encoder-decoder model, and then utilize them as input to a conditional GAN of full convolution design with the motivation to generate diverse sentences. Moreover, we develop a new evaluation metric named DCE to assess the diversity of a caption set. The potential of our method to generate diverse captions is demonstrated experimentally, through an elaborate experimental study involving two benchmark datasets MSVD and MSR-VTT.
