

# SHOT RETRIEVAL AND ASSEMBLY WITH TEXT SCRIPT FOR VIDEO MONTAGE GENERATION

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

With the development of video sharing websites, numerous users desire to create their own attractive video montages. However, it is difficult for inexperienced users to create a well-edited video montage due to the lack of professional expertise. In the meantime, it is time-consuming even for experts to create video montages of high quality, which requires effectively selecting shots from abundant candidates and assembling them together. Instead of manual creation, a number of automatic methods have been proposed for video montage generation. However, these methods typically take a single sentence as input for text-to-shot retrieval, and ignore the semantic cross-sentence coherence given complicated text script of multiple sentences. To overcome this drawback, we propose a novel model for video montage generation by retrieving and assembling shots with arbitrary text scripts. To this end, a sequence consistency transformer is devised for cross-sentence coherence modeling. More importantly, with this transformer, two novel sequence-level tasks are defined for sentence-shot alignment in sequence-level: Cross-Modal Sequence Matching (CMSM) task, and Chaotic Sequence Recovering (CSR) task. To facilitate the research on video montage generation, we construct a new, highly-varied dataset which collects thousands of video-script pairs in documentary. Extensive experiments on the constructed dataset demonstrate the superior performance of the proposed model. The dataset and generated video demos are available at <https://github.com/RATVDemo/RATV>.

## 1 INTRODUCTION

In recent years, with the rapid development of video sharing websites, users can conveniently share their own edited short videos (i.e., video montages), resulting in numerous video content creators that desire to create attractive video montages. However, editing video montage well is not easy for most of users due to the lack of professional expertise and aesthetic knowledge for video editing. In addition, for experts that master the video editing skills, it is time-consuming and cumbersome to create a video montage of high quality, because they have to carefully select shots from abundant candidates and then assemble the selected shots into a consecutive video montage that precisely expresses the desired content. Automatically generating video montages from descriptive sentences thus becomes topical, which aims to effectively retrieve shots from candidates and assemble them according to given text scripts.

Over the past few years, a number of previous methods have been proposed to address this task based on deep learning (Truong et al., 2016; Wang et al., 2019; Xiong et al., 2021). QuickCut (Truong et al., 2016) presents an interactive tool for narrated video editing, which can quickly create the story outline from raw video footage. It focuses on speeding up the process of establishing the script, but requires the users to manually select the shots from candidates. Write-A-Video (Wang et al., 2019) and Transcript-to-Video (Xiong et al., 2021) propose automatic methods to retrieve shots with given texts from a huge shot gallery and then arrange them. However, they only take a single sentence as input for text-to-shot retrieval without modeling the cross-sentence coherence, which limits their performance given complicated text script of multiple sentences.

In this work, we propose a novel model to automatically **R**etrieve and **A**ssemble shots with arbitrary **T**ext scripts for **V**ideo montage generation (abbreviated RATV). Our proposed RATV consists of three main components: a textual encoder, a visual encoder, and a sequence consistency transformer.

Table 1: Brief comparison among our RATV and existing works. **Sentence Num.** denotes the number of sentences contained in the input text. **Retrieved shots** denotes whether the method uses the retrieved shots for retrieval. **Seq. Con. Trans.** is short for sequence consistency transformer.

Method	Retrieval		Assembly	
	Sentence Num.	Retrieved Shots	Strategy	Text Used
QuickCut <sup>†</sup> (Truong et al., 2016)	Multiple	No	DP Algorithm	No
Write-A-Video <sup>†</sup> (Wang et al., 2019)	Single	No	Predefined Rules	No
Transcript-to-Video <sup>†</sup> (Xiong et al., 2021)	Single	Yes	Score of Classifier	No
RATV (ours)	Multiple	Yes	Seq. Con. Trans.	Yes

Similar to Write-A-Video (Wang et al., 2019), the textual encoder and visual encoder aim to encode texts and shots into a joint feature space for matching video-script pairs. Differently, we directly employ a large-scale pre-training visual-language model (CLIP (Radford et al., 2021)) to obtain our textual encoder and visual encoder instead of training them with text-shot pairs with keywords from scratch. Note that CLIP is trained on text-image retrieval task, but we generalize it to text-video matching task by simply considering the average of the frame embeddings as the embedding of corresponding video. Based on the embeddings extracted by textual encoder and visual encoder, the novel sequence consistency transformer learns to match sentence sequence and shot sequence, which can better retrieve shots according to complicated texts of multiple sentences and assemble the retrieved shots. Specifically, with this transformer, we devise two novel training tasks for better sentence-shot alignment in sequence-level: Cross-Modal Sequence Matching (CMSM) task and Chaotic Sequence Recovering (CSR) task. The CMSM task is induced to explicitly encourage the model to learn sentence-shot alignment by distinguishing the positive and negative samples. The CSR task enforces the model to learn to recover the order of chaotic shot sequences according to paired texts, which benefits to learning both the sequence coherence and sentence-shot alignment. Overall, the difference between our proposed RATV and existing methods is shown in Table 1.

To our best knowledge, there is no publicly available dataset for the research on video montage generation. Furthermore, despite the significant progress in video understanding datasets (Xu et al., 2016; Fouhey et al., 2018; Miech et al., 2019), these datasets focus on human actions (mainly with a single shot per video), which can not meet the demand of video montage generation with text script. Therefore, to fill the gap in dataset construction for this task, we create a new, highly varied dataset (Video-Script Pairs in Documentary, VSPD) to facilitate the community, which consists of rich and diverse video-script pairs collected from publicly available documentaries.

Our main contributions are four-fold: (1) We propose a novel model to automatically generate video montages by retrieving and assembling shots with arbitrary text scripts. (2) We devise novel Cross-Modal Sequence Matching (CMSM) and Chaotic Sequence Recovering (CSR) tasks, which are beneficial to learning both sentence-shot alignment in sequence-level and the coherence of shot sequence with text scripts. (3) To fill the gap in the dataset construction for video montage generation, we introduce the VSPD dataset that consists of diverse and highly varied video-script pairs from documentary videos. Meanwhile, we establish a benchmark for video montage generation task to facilitate the community. (4) Extensive experiments on the constructed VSPD dataset demonstrate the effectiveness and superior performance of our proposed method.

## 2 RELATED WORK

**Video Montage Generation.** Video montage generation with text script has been proposed for a long time (Chua & Ruan, 1995; Ahanger & Little, 1998), but it was barely studied in the community. In recent years, with the rapid development of video sharing websites, the demand of video montage generation becomes higher, and thus this task starts to draw more attention (Shen et al., 2009; Leake et al., 2017; Truong et al., 2016; Wang et al., 2019; Xiong et al., 2021). Particularly, QuickCut (Truong et al., 2016) presents an interactive tool for narrated video editing, which aims to help users to efficiently create the story outline of narrated videos. However, it only supplies the video segments corresponding to selected footage and requires users to manually select shots from them. In contrast, Write-A-Video (Wang et al., 2019) presents an interactive tool that enables users to automatically retrieve shots from a huge shot gallery and assemble them based on pre-defined rules. Transcript-to-Video (Xiong et al., 2021) proposes an automatic method to retrieve and assemble shots according to given texts. Although they can automatically generate video montages from texts without extra manual work, they only consider one sentence as input during retrieval and

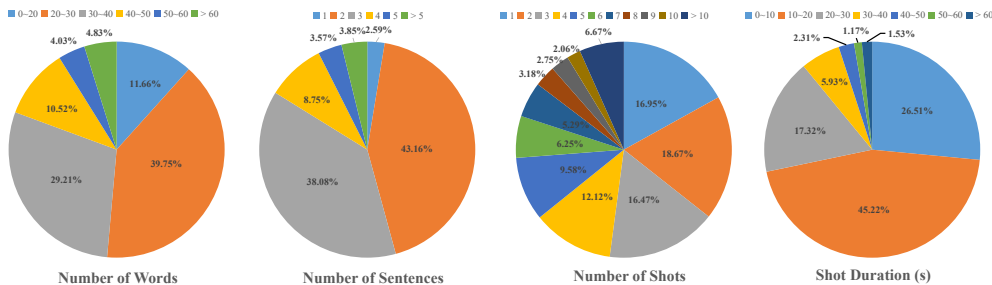


Figure 1: The statistics of text scripts and videos contained in our VSPD dataset.

assembly, which limits their performance given complicated texts containing multiple sentences. In this work, we propose a novel automatic method to retrieve and assemble shots with texts for video montages generation, which expands the input from only one sentence to a sequence of sentences.

**Vision-Language Representation Learning.** Visual Semantic Embeddings (VSE) (Frome et al., 2013; Faghri et al., 2018) are commonly adopted in multi-modal tasks to learn vision-language joint representations (Andonian et al., 2020; Huang et al., 2018; Krishna et al., 2017; Lei et al., 2018; Tapaswi et al., 2016). Recently, large-scale pre-training has achieved great success in vision-language representation learning (Chen et al., 2020; Li et al., 2020c; Radford et al., 2021; Jia et al., 2021; Huo et al., 2021), and shown superior performance in various downstream tasks (e.g., action recognition in videos, and zero-shot classification), where the extracted embeddings are directly employed without further training. In this work, we similarly leverage the embeddings extracted by CLIP (Radford et al., 2021) in video montage generation. Differently, we generalize the image-text representation to video-text representation and learn video-text alignment in sequence-level.

**Training Tasks for Transformer-Based Modeling.** Masked Language Model (MLM) is firstly proposed by BERT (Devlin et al., 2019) and then widely used for NLP with transformer. The MLM task randomly replaces a word token with the mask token (or another word token) with certain probability and enforces the model to predict the original word token, which has shown its superior power in representation learning. Inspired by BERT, ViLBERT (Lu et al., 2019) employs MLM, Masked Object Classification (MOC) and Visual-Linguistic Matching (VLM) tasks for vision-language representation learning. More recent works basically follow these training tasks (Li et al., 2019; 2020a; Tan & Bansal, 2019; Su et al., 2020; Chen et al., 2020; Li et al., 2020c; Kim et al., 2021). Different from these training tasks, we devise novel Cross-Modal Sequence Matching (CMSM) and Chaotic Sequence Recovering (CSR) tasks to encourage our transformer-based model to better learn the video-text joint representation as well as the sequence coherence.

### 3 DATASET CONSTRUCTION

A dataset for video montage generation should consist of thousands of video-script pairs, each having a narrated video with multiple shots and a paired script with descriptive sentences. To our best knowledge, there is no public dataset devoted to video montage generation. Similar to video montage generation, text-video retrieval aims to retrieve desired videos from candidates according to given texts, which has become topical in recent works (Bain et al., 2021; Lei et al., 2021; Li et al., 2020b; Wang et al., 2021; Zhu & Yang, 2020; Yu et al., 2018) along with many public datasets (Rohrbach et al., 2012; Xu et al., 2016; Fouhey et al., 2018; Miech et al., 2019). However, these datasets are not suitable for video montage generation, because they mainly focus on human actions (with a single shot per video). Furthermore, the texts in these datasets either are too general or only cover a part of the video content. To fill the gap in dataset construction for video montage generation, we create a new, highly varied dataset, named VSPD (shorted for Video-Script Pairs in Documentary). Our VSPD dataset consists of 4,365 video-script pairs, with its statistics shown in Figure 1. We randomly select 200 video-script pairs for test and the other 4,165 pairs for training. Our considerations in collecting documentary videos are two-fold: (1) Most of documentary videos are narrated with well-aligned captions, and thus we can conveniently collect numerous video-script pairs that are highly consistent in semantics; (2) Documentary videos are commonly carefully edited, resulting in extensive consecutive video clips with multiple shots. Importantly, although our VSPD consists of videos from documentaries, our proposed RATV model can be adopted to generate videos with various themes, which mainly depends on the input text scripts instead of theme of videos in the gallery. More details about our VSPD dataset are given in Appendix C.

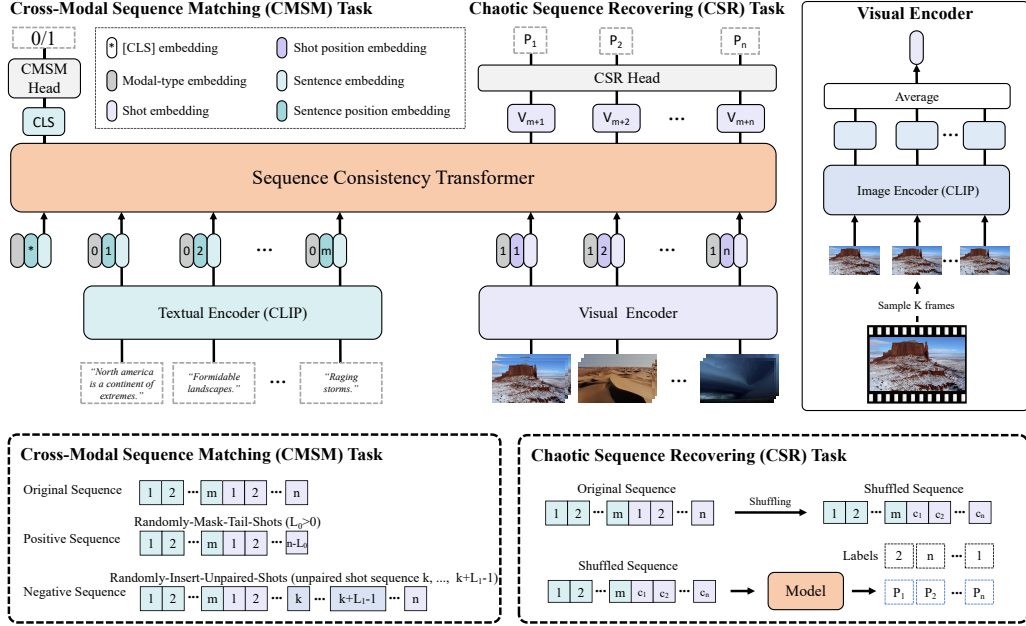


Figure 2: A schematic illustration of our proposed RATV model. RATV learns text-video alignment in sequence-level for video montage generation. Our novel Cross-Modal Sequence Matching (CMSM) and Chaotic Sequence Recovering (CSR) tasks are two key components for text-video alignment, which are only considered in the training phase.

## 4 METHODOLOGY

### 4.1 MODEL OVERVIEW

Let  $\mathcal{S}$  denote the set of candidate shots. Given an input text script  $t$ , the goal of video montage generation is to firstly retrieve a subset of shots  $\tilde{S} = \{\tilde{s}_i | i = 1, \dots, N\}$  from  $\mathcal{S}$  according to  $t$ , where  $\tilde{s}_i \in \mathcal{S}$  and  $N$  is the number of retrieved shots. The retrieved shots are then arranged in a certain order to create the final video montage  $V = (v_1, v_2, \dots, v_N)$ , where  $v_i \in \tilde{S}$  and  $v_i \neq v_j (\forall i, j = 1, 2, \dots, N, i \neq j)$ . Note that  $V$  should be temporally consecutive and consistent with the input text script  $t$  in semantics. To achieve this, we propose a novel model termed RATV. As illustrated in Figure 2, the main components of our RATV are textual encoder  $TE$ , visual encoder  $VE$  and sequence consistency transformer  $ST$ . In this work, the textual encoder and visual encoder can be formed with CLIP (Radford et al., 2021), both of which are frozen during training. Note that we employ image encoder of CLIP instead of other pre-trained video encoders due to its superior performance in vision-language semantic alignment (see Sec. 5.2). Given an input text script  $t$ , we first split it into sentences  $t = (\tilde{t}_1, \tilde{t}_2, \dots, \tilde{t}_m)$ , where  $\tilde{t}_i$  denotes the  $i$ -th sentence in text script  $t$  and  $m$  is the total number of sentences. We then encode each of them into a feature vector  $v_i^T (i = 1, 2, \dots, m)$  with the text encoder  $TE$ , and finally obtain a sequence of sentence embeddings  $(v_1^T, v_2^T, \dots, v_m^T)$ . The process of text encoding is defined as:

$$(v_1^T, v_2^T, \dots, v_m^T) = TE(\tilde{t}_1, \tilde{t}_2, \dots, \tilde{t}_m) = (TE(\tilde{t}_1), TE(\tilde{t}_2), \dots, TE(\tilde{t}_m)). \quad (1)$$

Similarly, we encode every shot in a shot sequence  $(s_1, s_2, \dots, s_n)$  into a feature vector  $v_i^S (i = 1, 2, \dots, n)$  with visual encoder  $VE$  and then concatenate them into a sequence of shot embeddings  $(v_1^S, v_2^S, \dots, v_n^S)$ . Specifically, for each shot in the sequence, we sample  $K$  frames from the original shot and encode every frame into feature vector  $v_{i,j}^F (i = 1, 2, \dots, n, j = 1, 2, \dots, K)$  with the image encoder  $VE$ . We simply average the feature vectors of  $K$  frames and take it as the feature vector of the corresponding shot. Formally, the process of extracting shot feature vectors is:

$$v_i^S = Avg(v_{i,1}^F, v_{i,2}^F, \dots, v_{i,K}^F) = Avg(VE(f_{i,1}), VE(f_{i,2}), \dots, VE(f_{i,K})), \quad (2)$$

where  $Avg(\cdot)$  is the average function and  $f_{i,j}$  is the  $j$ -th frame of the  $i$ -th shot.

On the top of the sequences of text embeddings and shot embeddings, we propose a novel sequence consistency transformer  $ST$  to learn the text-video joint representation in sequence-level. Note that

our proposed sequence consistency transformer supports one or multiple shots as input for flexibility (i.e.,  $n \geq 0$  and  $n$  is an integer). Concretely, following previous works (Kim et al., 2021; Chen et al., 2020), different token type embeddings are firstly added to each embedding in these two sequence respectively to discriminate text embedding and shot embedding. We then concatenate these two sequences into one sequence:

$$(\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_{m+n}) = (v_1^T + x, v_2^T + x, \dots, v_m^T + x, v_1^S + \tilde{x}, v_2^S + \tilde{x}, \dots, v_n^S + \tilde{x}), \quad (3)$$

where  $x$  and  $\tilde{x}$  denote the token type embedding of text and shot, respectively. Furthermore, the embedding of special [CLS] token is appended to the start of the concatenated sequence. The sequence consistency transformer  $ST$  is defined as:

$$(v_c, v_1, v_2, \dots, v_{m+n}) = ST(\tilde{v}_c, \tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_{m+n}), \quad (4)$$

where  $\tilde{v}_c$  denotes the embedding of [CLS] token. To encourage the  $ST$  to learn the text-video joint representation in sequence-level well, we devise Cross-Modal Sequence Matching (CMSM) Task (see Sec. 4.2) and Chaotic Sequence Recovering (CSR) (see Sec. 4.3) tasks for model training.

## 4.2 CROSS-MODAL SEQUENCE MATCHING (CMSM)

Similar to the Image-Text Matching (ITM) task Lu et al. (2019), we devise a novel Cross-Modal Sequence Matching (CMSM) Task to encourage our RATV model to learn text-video alignment in sequence-level. Specifically, we extract the representation of [CLS] token as the joint representation of the input sequence, and then feed it into a FC layer with a sigmoid function to predict a score between 0 and 1, which indicates the probability of that the input text and shot sequences are matching. Differently, we generate the positive and negative samples with two novel mechanisms. To create the positive samples, we introduce a randomly-mask-tail-shots mechanism. Concretely, we sample a script-video pair at each step during training, where the script and paired video are denoted as sentence sequence  $(\tilde{t}_1, \tilde{t}_2, \dots, \tilde{t}_m)$  and shot sequence  $(s_1, s_2, \dots, s_n)$ , respectively. We then randomly decide the number of shots to mask and mask the desired number of shots from the tail of the shot sequence to generate a positive sample. Formally, the randomly-mask-tail-shots mechanism is:

$$(s_1, s_2, \dots, s_{n-L_0}) = R((s_1, s_2, \dots, s_n), L_0), \quad (5)$$

where  $R(\cdot, \cdot)$  denotes the randomly-mask-tail-shots function and  $L_0$  is the randomly decided number of shots to mask ( $0 \leq L_0 < n$ ). Moreover, we also introduce a randomly-insert-unpaired-shots mechanism to create the negative samples. For a text-video pair, we randomly select another shot sequence  $(\hat{s}_1, \hat{s}_2, \dots, \hat{s}_{\hat{n}})$  from the training set, which is unpaired with the script. We randomly select one of the sub-sequences of unpaired shot sequence and insert it into the paired shot sequence in a random position. Formally, we randomly decide the length  $L_1$  and start position  $k$  ( $1 \leq L_1 \leq \hat{n}$ ,  $1 \leq k \leq \hat{n}$ ,  $k + L_1 \leq \hat{n} + 1$ ), and obtain the sub-sequence with them:

$$(\hat{s}_k, \hat{s}_{k+1}, \dots, \hat{s}_{k+L_1-1}) = G((\hat{s}_1, \hat{s}_2, \dots, \hat{s}_{\hat{n}}), k, L_1), \quad (6)$$

where  $G(\cdot, \cdot, \cdot)$  denotes the function of obtaining sub-sequence from the input sequence. The randomly-insert-unpaired-shots mechanism is formally defined as:

$$\begin{aligned} &(s_1, s_2, \dots, s_{n-L_2}, \hat{s}_k, \hat{s}_{k+1}, \dots, \hat{s}_{k+L_1-1}, s_{n-L_2+1}, \dots, s_n) \\ &= I((s_1, s_2, \dots, s_n), (\hat{s}_k, \hat{s}_{k+1}, \dots, \hat{s}_{k+L_1-1}), L_2), \end{aligned} \quad (7)$$

where  $I(\cdot, \cdot, \cdot)$  denotes the random insert shots function and  $L_2$  is the randomly decided position to insert the sub-sequence ( $0 \leq L_2 \leq n$ ). Since we create positive and negative samples by manipulating the shot sequence, the CMSM task encourages the model to learn a well-aligned text-video joint representation space in sequence level.

During training, we randomly decide the input sample is positive or negative with probability 0.5, and give the corresponding binary label  $y \in \{0, 1\}$  (i.e., 0 is negative and 1 is positive). We then apply the randomly-mask-tail-shots or randomly-insert-unpaired-shots mechanism on the sample and feed it into the sequence consistency transformer  $ST$  to obtain the matching score  $M$ . The binary cross-entropy loss is taken on board for optimization:

$$\mathcal{L}_V = \mathbb{E}[-(y \log M + (1 - y) \log(1 - M))]. \quad (8)$$

### 4.3 CHAOTIC SEQUENCE RECOVERING (CSR)

We further devise another novel pretext task termed Chaotic Sequence Recovering (CSR) for text-video joint representation learning. The CSR task aims to recover the original order of the shots in the chaotic shot sequence with the paired text. Formally, given the input sentence sequence  $(\tilde{t}_1, \tilde{t}_1, \dots, \tilde{t}_m)$  and its paired shot sequence  $(s_1, s_2, \dots, s_n)$ , we first shuffle the shot sequence:

$$(c_1, c_2, \dots, c_n) = S(s_1, s_2, \dots, s_n), \quad (9)$$

where  $S(\cdot)$  denotes the function of shuffling the input sequence,  $c_i \in \{s_1, s_2, \dots, s_n\}$  and  $c_i \neq c_j$  ( $\forall i, j = 1, 2, \dots, n, i \neq j$ ). Meanwhile, we can obtain the original positions of all shots in the shuffled sequence, which are defined as  $(\tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_n)$ . After feature extraction, the shuffled sequence is transformed to a sequence of shot embeddings, which is then concatenated with the sequence of sentence embeddings and embedding of [CLS] token. According to Eq. (3), we have the concatenated sequence of embeddings  $(\tilde{v}_c, \tilde{v}_1^T, \tilde{v}_2^T, \dots, \tilde{v}_m^T, \tilde{v}_1^S, \tilde{v}_2^S, \dots, \tilde{v}_n^S)$ . Note that we use different symbols to distinguish the sentence and shot embeddings here for easier understanding. We finally feed the sequence into  $ST$  and employ a FC layer with a softmax function on every outputted shot embedding to predict the score of  $i$ -th position:

$$p_i = SM(FC(v_i^S)), i = 1, 2, \dots, n, \quad (10)$$

where  $SM(\cdot)$  is the softmax function,  $FC(\cdot)$  is the FC layer. Note that the length of shot sequence  $n$  can vary for different input and the output dimension of FC thus can not be defined. To address this, we follow previous transformer based methods to set the max length  $L_{max}$ . If the length of input shot sequence is less than  $L_{max}$ , we pad the sequence so that its length is equal to  $L_{max}$ , and mask the padding tokens when the sequence is fed into the ST. If the length of input shot sequence is greater than  $L_{max}$ , we truncate the sequence so that its length is equal to  $L_{max}$ . With this setting, the FC layer is defined for  $L_{max}$ -class classification task (i.e.,  $p_i \in \mathbb{R}^{L_{max}}$ ). The CSR task minimizes the cross-entropy loss with  $(\tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_n)$  as ground-truth labels:

$$\mathcal{L}_R = \mathbb{E} \left[ \sum_{i=1}^n CE(p_i, \tilde{p}_i) \right], \quad (11)$$

where  $CE(\cdot, \cdot)$  denotes the cross-entropy function. Note that the CSR task is parallel with the CMSM task, i.e., our model is trained with these two tasks together at each step during training. The overall loss can be defined as:

$$\mathcal{L}_{RATV} = \mathcal{L}_V + \lambda \mathcal{L}_R, \quad (12)$$

where  $\lambda$  is the hyperparameter to balance the two losses.

### 4.4 INFERENCE

In the inference phase, our proposed RATV generates video montages from given text scripts by retrieving shots from the set of candidates  $\mathcal{S}$  iteratively. Formally, given a query text  $t$ , we encode it into sentence embedding sequence  $(v_1^T, v_2^T, \dots, v_m^T)$  according to Eq. (1). We then retrieve a shot from candidates at a time, resulting in retrieved shot sequence  $(\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_{t-1})$  at step  $t$  (the sequence is empty when  $t = 1$ ). For shot  $s_c$  in leftover candidates (i.e.,  $s_c \in \mathcal{S}$  and  $s_c \notin \{\tilde{s}_i | i = 1, 2, \dots, t-1\}$ ), we calculate the scores in both instance-level and sequence-level, and add them as the ensemble score for shot  $s_c$ . Concretely, we get the shot embedding  $v_c^S$  with Eq. (2), and calculate the score in instance-level to measure whether the shot  $s_c$  and input text  $t$  are matching in semantics:

$$I_c = \cos(\text{Avg}(v_1^T, v_2^T, \dots, v_m^T), v_c^S), \quad (13)$$

where  $\cos(u, v) = u^T v / \|u\| \|v\|$  denotes the cosine similarity between the two vectors  $u$  and  $v$ . In addition, we append the shot  $s_c$  to the end of retrieved shot sequence and feed it with input text to sequence consistency transformer  $ST$  to get the output matching score  $M_c$  as the score in sequence-level. The ensemble score is  $E_c = M_c + \lambda_s I_c$ , where  $\lambda_s$  is the weight hyperparameter. The shot with highest ensemble score is taken as the retrieved shot  $\tilde{s}_t$  and appended to the end of retrieved shot sequence. We set a threshold hyperparameter  $\epsilon$  for automatically terminating the inference process. That is, when the highest ensemble score  $E_h \leq \epsilon$ , our RATV terminates the process and returns the retrieved shot sequence as the generated video montage. Crucially, our proposed RATV learns video-text joint representation in sequence-level with carefully designed two mechanisms in

Table 2: Quantitative results on the VSPD dataset. † denotes directly using the pre-trained model for retrieval without fine-tuning. ↓ means that lower is better while ↑ means the opposite. CLIP-A denotes that we split the input text into sentences and encode each sentence with CLIP, and then use the average of the sentence embeddings for retrieval.

Method	Automated Metrics				User Study	
	IoU ↑	UMS ↓	SMS ↑	CS ↑	Semantic ↑	Coherence ↑
VSE (Frome et al., 2013)	0.017	4.925	0.013	0.194	2.47	3.34
VSE++ (Faghri et al., 2018)	0.020	6.141	0.007	0.040	2.14	3.08
MIL-NCE† (Miech et al., 2020)	0.011	5.431	0.002	0.116	–	–
MIL-NCE	0.023	6.111	0.009	0.035	2.03	2.74
Frozen-in-Time† (Bain et al., 2021)	0.077	4.723	0.054	0.143	–	–
Frozen-in-Time	0.085	5.213	0.066	0.178	2.95	3.29
CLIP† (Radford et al., 2021)	0.072	5.026	0.034	0.073	–	–
CLIP-A†	0.104	4.669	0.072	0.095	3.09	3.24
Write-A-Video† (Wang et al., 2019)	0.104	4.669	0.079	0.097	3.09	3.24
Transcript-to-Video† (Xiong et al., 2021)	0.096	4.621	0.064	0.124	2.69	3.16
RATV (ours)	<b>0.144</b>	<b>3.393</b>	<b>0.090</b>	<b>0.685</b>	<b>3.40</b>	<b>3.76</b>

the CMSM task. Therefore, the matching score measures whether the candidate shot  $s_c$  is matching with the input text and retrieved shot sequence in semantics and coherence at the same time. As a result, our RATV can directly return the retrieved shots in sequence of retrieval order without rearranging them. The illustration of inference process is shown in Figure 6 in Appendix A.

## 5 EXPERIMENTS

### 5.1 EXPERIMENTAL SETUP

We adopt four metrics for quantitative evaluation. **(1) Intersection over Union (IoU)**. Recall is considered as an important metric for retrieval task in previous works, which measures whether the target in the top-K retrieved results (resulting in Recall@K). Note that the target in retrieval task commonly contains a single instance, but the target in video montage generation contains multiple shots. Therefore, we adopt IoU instead of recall, which is applied on the retrieved shots and the (ground truth) shots paired with each query text. **(2) Unmatching Score (UMS)**. This metric is defined as follows: for all shots in a generated video, we find those shots not in the (ground truth) video paired with the input/query text, then calculate the dissimilarity  $dsim = 1 - sim$  for each of the found shots, where  $sim$  is the similarity between shot and the input text calculated by CLIP. We finally add all dissimilarities together as the UMS. Note that ResNet50×4 (not used by any competitors) from CLIP is used to calculate the similarity. **(3) Sequence Matching Score (SMS)**. This metric is used to evaluate the overall quality of the generated video montage. Specifically, given a query/input text  $t$ , we define the paired (ground truth) shot sequence and retrieved shot sequence as  $\{s_1, s_2, \dots, s_n\}$  and  $\{\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_{\tilde{n}}\}$ , respectively. The SMS is defined as:  $SMS = \frac{1}{n} \sum_{i=1}^{\min(n, \tilde{n})} \mathbb{B}(s_i = \tilde{s}_i)$ , where  $\mathbb{B}(\cdot)$  is the indicator function (indicating whether two shots are the same). **(4) Consistency Score (CS)**. This metric is used to evaluate the consistency of the generated video given the input text. To obtain this score for arbitrary video-text pair, a binary classifier is needed to distinguish whether all shots in the video are both temporally consecutive and consistent with the context of the text. In this work, we train the binary classifier over a large set of video-text pairs, where the positive pairs are directly obtained from the training set but the negative pairs are generated by shuffling the shots in the video from each positive pair. The classifier can achieve over 90% prediction accuracy on the test set. Additionally, other settings are given in Appendix B.

### 5.2 QUANTITATIVE RESULTS

The quantitative results are shown in Table 2. It can be observed that: **(1)** Our RATV outperforms all competitors with large margins on all metrics, indicating that our method can retrieve shots from candidates more precisely and assemble them into final video montage with better coherence. **(2)** CLIP-A leads to significant improvements over CLIP (Radford et al., 2021), which shows that CLIP still suffers from large information loss when it directly encodes complicated texts of multiple sentences. This is mainly due to the fact that only short/brief texts are used for pre-training CLIP. **(3)** Among the three methods (i.e., Write-A-Video, Transcript-to-Video, and our RATV) that consider the coherence during video montage generation, our RATV performs the best because of expanding from instance-level modeling to sequence-level modeling.

Table 3: Ablation study results for our full RATV model. Base denotes the RATV model trained with the ITM task. RM and RI denote the randomly-mask-tail-shots and randomly-insert-unpaired-shots mechanisms, respectively. The full CMSM task is equal to RM+RI.

Method	IoU $\uparrow$	UMS $\downarrow$	SMS $\uparrow$	CS $\uparrow$
Base	0.093	4.741	0.051	0.079
Base+RM	0.109	4.513	0.056	0.131
Base+RI	0.096	4.373	0.078	0.256
Base+CMSM	0.142	3.535	0.083	0.645
Base+CSR	0.099	4.608	0.065	0.136
Base+RM+CSR	0.116	4.235	0.070	0.164
Base+RI+CSR	0.099	4.282	0.087	0.337
RATV (ours)	<b>0.144</b>	<b>3.393</b>	<b>0.090</b>	<b>0.685</b>

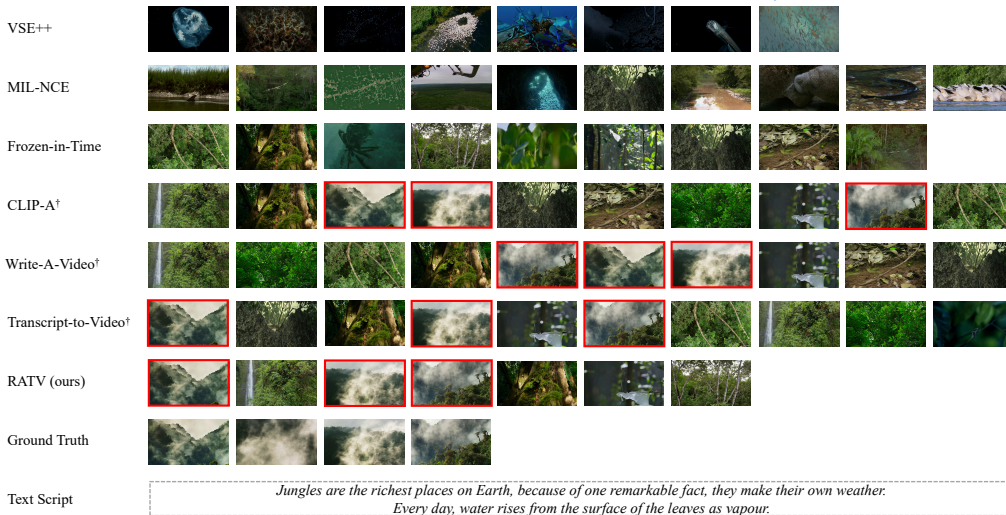


Figure 3: Qualitative results for video montage generation on the VSPD dataset. Every image here denotes a shot, and the image in red box means that the shot falls in the ground-truth video. Each row (except the last two rows) shows the shots in the video generated by a compared method. These videos are all generated according to the text script in the last row. Note that the number of shots of some generated videos is less than that of the other videos due to the earlier termination by threshold.

We further conduct user study to evaluate the quality of generated videos under human perception. Table 2 shows the user study results. As expected, our RATV outperforms all competitors on both semantic consistency and temporal coherence. Interestingly, Write-A-Video and Transcript-to-Video are even inferior to CLIP-A and Frozen-in-Time under human perception, showing that their assembly techniques (using pre-defined rules or coherence classifier) are not that effective without considering the context of input text. Details of user study are given in Appendix D.

### 5.3 ABLATION STUDY

We conduct ablation study to show the contribution of our proposed CMSM and CSR tasks. We firstly adopt simple ITM task (Chen et al., 2020) to train our proposed RATV model, which is denoted as Base. To further explore the contribution of the two novel tasks, we then gradually add randomly-mask-tail-shots mechanism (RM), randomly-insert-unpaired-shots mechanism (RI) and CSR task on the top of Base. When we adopt RM (RI) along, we follow the strategy of ITM task to create negative (positive) samples. Our full RATV is actually Base+RI+RM+CSR, which is trained with full CMSM (i.e., RI+RM) and CSR tasks. The results of ablation study are shown in Table 3.

We have the following observations: (1) The RM and RI lead to improvements on all metrics, indicating that both mechanisms are beneficial to text-video joint representation learning for video montage generation (2) The combination of RM and RI in the CMSM task yields significant improvements on all metrics over adopting them alone, which shows the complementarity of the two mechanisms. Crucially, our RATV trained with the CMSM task has now outperformed CLIP-A (see Base+CMSM in Table 3 vs. CLIP-A<sup>†</sup> in Table 2), which directly verifies the effectiveness of our proposed RATV framework for video montage generation. (3) When combined with either of the two mechanisms, the CSR task leads to further improvements on all metrics, especially on the SMS and CS. This still holds for CMSM+CSR (i.e., our full RATV).



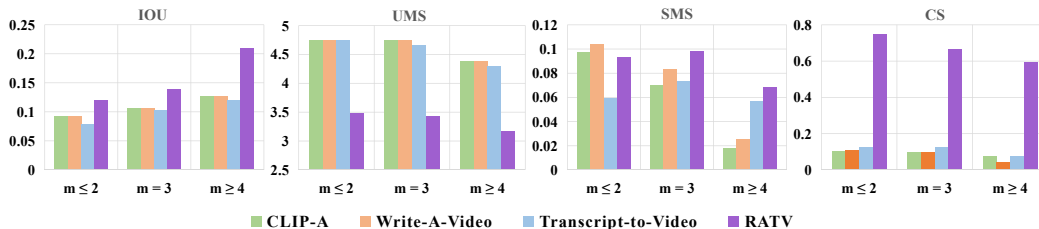


Figure 4: Comparative results on three subsets of the test set. These test subsets are obtained by splitting the test set according to the number of sentences  $m$  in each input text script.

#### 5.4 QUALITATIVE RESULTS

The qualitative results on the VSPD dataset are shown in Figure 3. Among the competitors, VSE++ (Faghri et al., 2018), MIL-NCE (Miech et al., 2020), and Frozen-in-Time (Bain et al., 2021) are trained on the VSPD dataset. We can observe that: (1) Compared with the other competitors, the shots retrieved by CLIP-based methods can express the text script more precisely, which also contain the 3 shots from the ground-truth video. (2) Based on CLIP, Write-A-Video and Transcript-to-Video can both retrieve the shots that are well aligned with the text script. Importantly, the shots in the generated videos are more consecutive due to the pre-defined rules and the coherence classifier (as compared with CLIP-A). However, some shots in the generated videos are still in an unreasonable order or not so relevant to the text script, because the context of the text script is ignored when arranging the retrieved shots. (3) Our RATV can generate the video that precisely expresses the whole text script. Importantly, our model can consider the semantic alignment and temporal coherence in the meantime during video montage generation. The generated video thus does not contain any shots irrelevant to the text script. Although our model happens to miss the second shot of the ground-truth video, this shot is not found by any competitors. More results are given in Appendix E.

#### 5.5 FURTHER EVALUATION

**Number of Input Sentences.** We make further comparison under different test conditions. Concretely, we split the test set into three subsets according to the number of sentences  $m$  ( $m \leq 2$ ,  $m = 3$ ,  $m \geq 4$ ) in each input text script, and report comparative results on each test subset in Figure 4. We can observe that our RATV consistently outperforms all competitors on three (i.e., IOU, UMS, and CS) out of four metrics over all test subsets. When it comes to SMS, our RATV is slightly inferior to CLIP-A and Write-A-Video when  $m \leq 2$ . However, as  $m$  increases, the performance of CLIP-A and Write-A-Video in terms of SMS decreases sharply, thus becoming worse than that of our RATV. Overall, the superior performance of our RATV with  $m > 2$  shows that our RATV is indeed effective in video montage generation given complicated text script of multiple sentences.

**Wild Text Scripts.** Although the VSPD dataset contains the videos from documentaries, the goal of our RATV is not to generate documentaries only. In contrast, it devotes to generating videos on various themes which mainly depend on the input text scripts. To demonstrate this, we deploy our RATV to generate travel vlog with narrated text scripts, MV for music with lyrics, and background video with poem, which are shown in Figure 10 in Appendix F. The results suggest that our RATV has a good ability of generating videos on various themes with wild text scripts.

## 6 CONCLUSION

In this work, we have proposed a novel framework termed RATV to automatically generate video montages by retrieving and assembling shots with arbitrary text scripts. Due to the novel Cross-Modal Sequence Matching (CMSM) and Chaotic Sequence Recovering (CSR) tasks, our proposed RATV can effectively learn the text-video joint representation in sequence-level and also the coherence of shot sequence. To our best knowledge, our RATV is the first model for video montage generation based on text-to-sequence retrieval, which can generate video montages more consistent with the input text scripts. To fill the gap in dataset construction for video montage generation, we create a new, highly varied dataset called VSPD, which contains thousands of diverse video-script pairs. Extensive experiments on the VSPD dataset demonstrate the effectiveness of our RATV.

## REFERENCES

- Gulrukh Ahanger and Thomas D. C. Little. Automatic composition techniques for video production. *IEEE Transactions on Knowledge and Data Engineering*, 10(6):967–987, 1998.
- Alex Andonian, Camilo Fosco, Mathew Monfort, Allen Lee, Rogério Feris, Carl Vondrick, and Aude Oliva. We have so much in common: Modeling semantic relational set abstractions in videos. In *ECCV*, volume 12363, pp. 18–34, 2020.
- Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, pp. 1708–1718, 2021.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: universal image-text representation learning. In *ECCV*, volume 12375, pp. 104–120, 2020.
- Tat-Seng Chua and Li-Qun Ruan. A video retrieval and sequencing system. *ACM Transactions on Information Systems*, 13(4):373–407, 1995.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186, 2019.
- Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. VSE++: improving visual-semantic embeddings with hard negatives. In *BMVC*, pp. 12, 2018.
- David F. Fouhey, Weicheng Kuo, Alexei A. Efros, and Jitendra Malik. From lifestyle vlogs to everyday interactions. In *CVPR*, pp. 4991–5000, 2018.
- Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc’Aurelio Ranzato, and Tomáš Mikolov. Devise: A deep visual-semantic embedding model. In *NeurIPS*, pp. 2121–2129, 2013.
- De-An Huang, Shyamal Buch, Lucio M. Dery, Animesh Garg, Li Fei-Fei, and Juan Carlos Niebles. Finding ”it”: Weakly-supervised reference-aware visual grounding in instructional videos. In *CVPR*, pp. 5948–5957, 2018.
- Yuqi Huo, Manli Zhang, Guangzhen Liu, Haoyu Lu, Yizhao Gao, Guoxing Yang, Jingyuan Wen, Heng Zhang, Baogui Xu, Weihao Zheng, et al. WenLan: Bridging vision and language by large-scale multi-modal pre-training. *arXiv preprint arXiv:2103.06561*, 2021. URL <https://arxiv.org/abs/2103.06561>.
- Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, pp. 4904–4916, 2021.
- Wonjae Kim, Bokyung Son, and Ildoo Kim. ViLT: Vision-and-language transformer without convolution or region supervision. In *ICML*, volume 139, pp. 5583–5594, 2021.
- Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, pp. 706–715, 2017.
- Mackenzie Leake, Abe Davis, Anh Truong, and Maneesh Agrawala. Computational video editing for dialogue-driven scenes. *ACM Transactions on Graphics*, 36(4):130:1–130:14, 2017.
- Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L. Berg. TVQA: localized, compositional video question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1369–1379, 2018.
- Jie Lei, Linjie Li, Luwei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: ClipBERT for video-and-language learning via sparse sampling. *CVPR*, pp. 7331–7341, 2021.

- Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training. In *AAAI*, pp. 11336–11344, 2020a.
- Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. HERO: Hierarchical encoder for video+ language omni-representation pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2046–2065, 2020b.
- Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. URL <https://arxiv.org/abs/1908.03557>.
- Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantic aligned pre-training for vision-language tasks. In *ECCV*, volume 12375, pp. 121–137, 2020c.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visual-linguistic representations for vision-and-language tasks. In *NeurIPS*, pp. 13–23, 2019.
- Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, pp. 2630–2640, 2019.
- Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *CVPR*, 2020.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, pp. 8748–8763, 2021.
- Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. A database for fine grained activity detection of cooking activities. In *CVPR*, 2012.
- Edward Yu-Te Shen, Henry Lieberman, and Glorianna Davenport. What’s next?: emergent storytelling from video collection. In *SIGCHI Conference on Human Factors in Computing Systems*, pp. 809–818, 2009.
- Tomáš Souček and Jakub Lokoč. TransNet V2: An effective deep network architecture for fast shot transition detection. *arXiv preprint arXiv:2008.04838*, 2020. URL <https://arxiv.org/abs/2008.04838>.
- Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: pre-training of generic visual-linguistic representations. In *ICLR*, 2020. URL <https://openreview.net/forum?id=SygXPaEYvH>.
- Hao Tan and Mohit Bansal. LXMERT: learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5099–5110, 2019.
- Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *CVPR*, pp. 4631–4640, 2016.
- Anh Truong, Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. Quickcut: An interactive tool for editing narrated video. In *Annual Symposium on User Interface Software and Technology (UIST)*, pp. 497–507, 2016.

- Miao Wang, Guo-Wei Yang, Shi-Min Hu, Shing-Tung Yau, and Ariel Shamir. Write-a-video: computational video montage from themed text. *ACM Transactions on Graphics*, 38(6):177:1–177:13, 2019.
- Xiaohan Wang, Linchao Zhu, and Yi Yang. T2VLAD: global-local sequence alignment for text-video retrieval. In *CVPR*, pp. 5079–5088, 2021.
- Yu Xiong, Fabian Caba Heilbron, and Dahua Lin. Transcript to video: Efficient clip sequencing from texts. *arXiv preprint arXiv:2107.11851*, 2021. URL <https://arxiv.org/abs/2107.11851>.
- Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, pp. 5288–5296, 2016.
- Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *ECCV*, pp. 487–503, 2018.
- Linchao Zhu and Yi Yang. ActBERT: Learning global-local video-text representations. In *CVPR*, pp. 8743–8752, 2020.

## A SUPPLEMENTAL ILLUSTRATION

We provide the supplemental illustration of the comparison between our RATV and existing methods in Figure 5. For easier understanding, we also provide the visualization for inference pipeline in Figure 6. The detailed descriptions about these two figures have been given in Sec. 2 and Sec. 4.4 of the main paper, respectively.

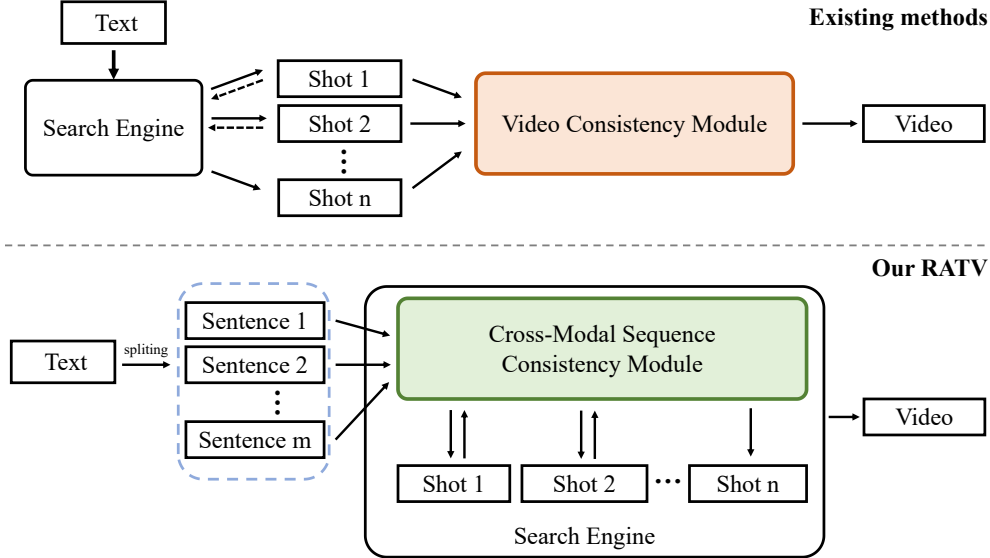


Figure 5: Pipelines of existing methods and our proposed RATV for video montage generation. Input text of existing methods is typically a single sentence.

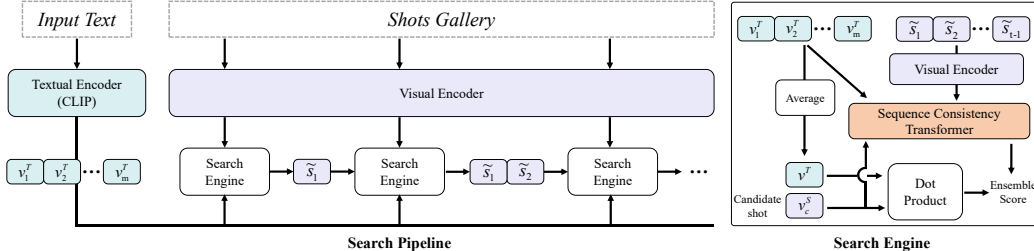


Figure 6: The inference pipeline of our proposed RATV. At each step, RATV retrieves one shot from shot gallery with search engine. Note that we adopt the ensemble technique to exploit the similarity between input text and candidates in both instance-level and sequence-level for text-to-shot retrieval.

## B SETTINGS

**Implementation Details.** For extracting the shot embeddings with the pre-trained CLIP (Radford et al., 2021), we follow previous works (Miech et al., 2020; Bain et al., 2021) to resize all selected frames in the shot videos to  $256 \times 256$  and centrally crop them into  $224 \times 224$  (i.e., the input image size of CLIP). In the training phase, we set the batch size to 64 and total epochs to 200 to train our RATV model (with pre-trained CLIP frozen). We adopt 2-layer (with 512 hidden units and 8 heads) transformer as our Sequence Consistency Transformer. For better training stability, we adopt AdamW (Loshchilov & Hutter, 2019) with the learning rate  $5e-5$ ,  $\beta_0 = 0.9$ ,  $\beta_1 = 0.98$  as the optimizer for training. We adopt the strategy of applying weight decay as in ViLT (Kim et al., 2021). The hyperparameters are empirically set to  $\lambda = 1$ ,  $\lambda_s = 0.75$  and  $\epsilon = 1.1$ . Our RATV is trained on PyTorch with a single TITAN RTX GPU, which takes about 2 hours for training.

**Baseline Methods.** Since cross-modal retrieval is one of the core steps of video montage generation, we directly employ existing cross-modal retrieval methods as baseline methods for video

montage generation, in addition to the state-of-the-art methods (Wang et al., 2019; Xiong et al., 2021). Concretely, we compare our RATV with VSE (Frome et al., 2013), VSE++ (Faghri et al., 2018), MIL-NCE (Miech et al., 2020), Frozen-in-Time (Bain et al., 2021), CLIP (Radford et al., 2021), Write-A-Video (Wang et al., 2019) and Transcript-to-Video (Xiong et al., 2021). VSE and VSE++ are retrained with the default setting on our VSPD dataset, which originally focus on image-text retrieval but can be generalized to text-shot retrieval by averaging the frame embeddings as shot embedding. MIL-NCE and Froze-in-Time both devote to text-video retrieval, which are pre-trained on HowTo100M (Miech et al., 2019) and WebVid2M (Bain et al., 2021), respectively. CLIP is a large-scale visual-language pre-training model, which is pre-trained over 400M text-image pairs for text-image retrieval. We generalize CLIP to text-shot retrieval just as VSE and VSE++. Further, we also split the input text into sentences and encode each sentence with CLIP, similar to our RATV. The average of the sentence embeddings is considered as embedding of the input text and then used for retrieval, which is denoted as CLIP-A. Note that all of cross-modal retrieval methods arrange the retrieved shots in descending order by similarity. We re-implement Write-A-Video by replacing the backbone of its retrieval network with CLIP and adopting the pre-defined rules introduced in Wang et al. (2019). Similarly, for Transcript-to-Video, we also replace its backbone with CLIP and employ the beam search strategy with an extra coherence classifier trained on VSPD for re-implement it, which encodes the input texts like CLIP-A. Note that the threshold for terminating the retrieval process has influence on the final results, which requires carefully design for different methods. To alleviate this, we set the max length of retrieved shot sequence to 10 for all competitors because more than 90% videos in the VSPD dataset contain less than 10 shots. For performance evaluation, we generate 200 video montages with the text scripts in the test set of our VSPD using each compared method. The shot candidates consist of all shots from the 200 ground-truth videos in the test set. We average the results of 200 generated videos as the final results for each method.

## C DATASET CONSTRUCTION

Note that video montage generation task is a highly subjective task, in which there can be multiple generated samples that could be almost equally good. Therefore, establishing a dataset to well evaluate this task is extreme challenging. To alleviate this issue, we collect video-script pairs from videos that have been well edited by the experts (e.g., movie, cartoon and MV). However, these thress types of videos are not good choice: (1) The captions of movie and cartoon are commonly not consistent with the shots because most of captions are dialogues. (2) The shots in MV are commonly not temporal coherence. As a result, we propose to collect the video-script pairs from documentaries, whose captions are descriptive and consistent with the temporally consecutive shots (as stated in the Section 3). More importantly, the documentaries are commonly shot and edited by the experts that have professional expertise. Therefore, the collected videos (i.e., the continuous clips of documentaries) in our VSPD dataset can indeed be considered as the perfect videos w.r.t. semantic consistency, temporal coherence and aesthetics, i.e., they can be used as the ground-truth for training and evaluation. With evaluation on such dataset, a model has to consider all of the factors (e.g., text-video alignment, temporal coherence, and aesthetics) for video montage generation..

In this work, we search documentary videos on YouTube and filter them with the type of ‘movie’. These candidate videos are then filtered according to their themes to remove those unsuitable for this work. We also remove the videos without available caption files. We finally download 51 documentary videos with their caption files. Note that these videos consist of documentaries with various themes (e.g., animal, ocean and forest), resulting in a large diversity of shots in our VSPD dataset. As the *shot* is the basic unit in video montage generation, we divide these chosen videos into shots with a accurate shot boundary detector (Souček & Lokoč, 2020). The captions in the caption files are also split into a series of short texts of multiple sentences automatically. Concretely, we consider each line of a caption text file as the basic unit, which contains the timestamp and several words (not a complete sentence sometimes). We firstly combine the consecutive words into a sentence according to stop punctuation and process the timestamp at the same time. Further, if the time interval between two sentences is less than one second, we consider they are semantically relevant and combine them into a short text script, and also fuse the timestamps of these two sentences. Finally, we obtain a series of short text scripts and their timestamps. Note that the timestamps are used to obtain the shots that correspond to the text script. To avoid meaningless or wrong text scripts, we manually check all text scripts and remove some inappropriate ones (e.g., dialogue) by ourselves. We then form each video-script pair by combining the short text scripts and the corresponding shots in the

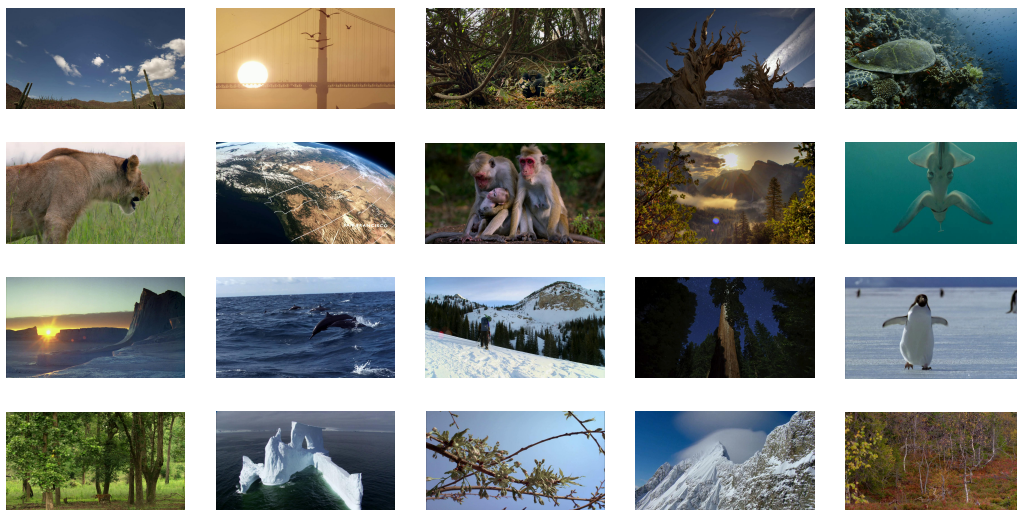


Figure 7: Samples of video shots in our constructed VSPD dataset. Each picture here denotes a shot.

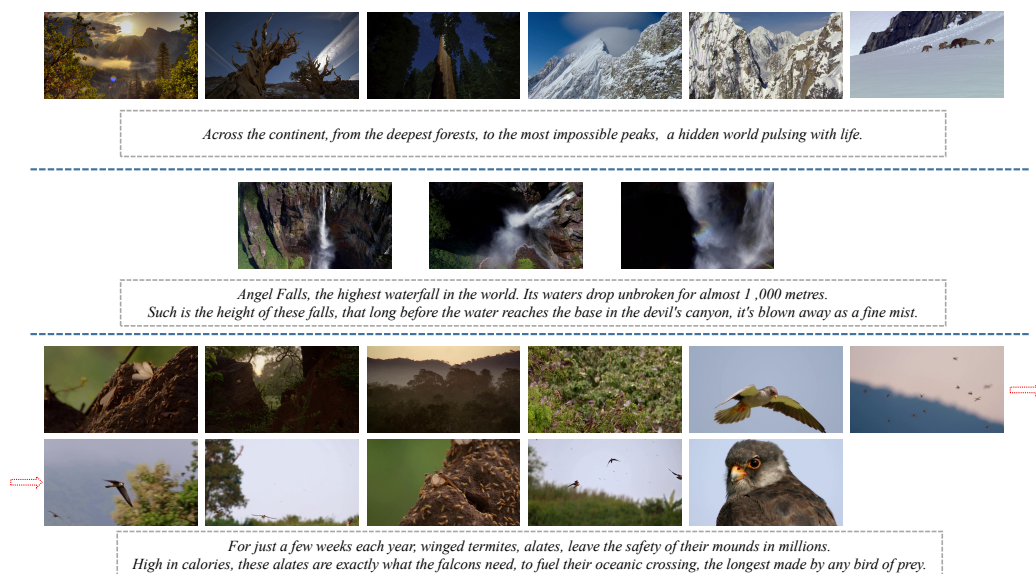


Figure 8: Samples of script-video pairs in our constructed VSPD dataset. Each row shows the video and its corresponding text script. Each picture denotes a shot in the video and the shots are arranged in chronological order from left to right in each row. The red arrow denotes the two shots are contiguous.

original video. We provide some samples of our constructed VSPD dataset. The samples of video shots in our VSPD dataset are shown in Figure 7. We can observe that the shots in our VSPD dataset are highly varied due to the large diversity of the themes of our collected documentaries. Further, we also provide the samples of script-video pairs in our VSPD dataset, which are shown in Figure 8. It can be clearly seen that the videos are highly consistent with the corresponding text scripts in semantics. More importantly, the order of shots in the videos is also basically consistent with that of the words in the text scripts. Our VSPD dataset is available at the anonymous GitHub link<sup>1</sup>, where we can download the whole set of complete videos instead of only pictures.

<sup>1</sup><https://github.com/RATVDemo/RATV>



Figure 9: Qualitative results for video montage generation on the VSPD dataset. Every image here denotes a shot, and the image in red box means that the shot falls in the ground-truth video. Each row (except the last two rows) shows the shots in the video generated by a compared method. These videos are all generated according to the text script in the last row. Note that the number of shots of some generated videos is less than that of the other videos due to the earlier termination by threshold.

## D USER STUDY ON PERFORMANCE EVALUATION

Specifically, with each method, we randomly select 50 text scripts from the test set and generate videos according to these scripts. We then invite volunteers to score all generated videos (i.e., 400 videos in total for all methods) according to the semantic consistency and temporal coherence, which are shortened as ‘Semantic’ and ‘Coherence’, respectively. We recruited volunteers from our university and finally selected 6 male volunteers and 4 female volunteers for user study, which include 3 undergraduate students, 3 master students and 4 Ph.D students. Among 10 volunteers, 8 of them are inexperienced for video creation/editing and the other 2 volunteers have created video montages by themselves. Importantly, all of them are active users of video sharing websites (e.g., YouTube and TikTok), which ensures the results of user study are convincing. In this paper, ten independent volunteers are asked to score the videos from 1 to 5 (higher is better), and the chosen text scripts for video montage generation are different for different volunteers. Table 2 of the main paper shows the user study results averaged over all ten volunteers.

## E ADDITIONAL QUALITATIVE RESULTS

In this section, we provide more qualitative results for video montage generation with text scripts on our constructed VSPD dataset. Specifically, we compare our proposed RATV with VSE [Frome et al. \(2013\)](#), VSE++ [Faghri et al. \(2018\)](#), MIL-NCE [Miech et al. \(2020\)](#), Frozen-in-Time [Bain et al. \(2021\)](#), CLIP [Radford et al. \(2021\)](#), Write-A-Video [Wang et al. \(2019\)](#) and Transcript-to-Video [Xiong et al. \(2021\)](#). Note that Write-A-Video and Transcript-to-Video are the state-of-the-art methods for video montage generation, and the other methods belong to existing cross-modal retrieval methods that can be directly employed as baseline methods for video montage generation. The quality results are shown in Figure 9. We can observe that: (1) VSE, VSE++ and MIL-NCE can not retrieve the shots that express the given text script precisely. (2) Frozen-in-Time and CLIP-A can find the shots that are more relevant to the text scripts, which contain the 2 shots from the ground-truth video. However, they make mistakes in retrieving the word ‘dolphins’, resulting in the ‘whale’ and ‘penguin’ in the videos (e.g., the last but one and the last but two shots in the video generated by Frozen-in-Time, and the fifth shot in the video generated by CLIP). (3) Based on CLIP, Write-A-



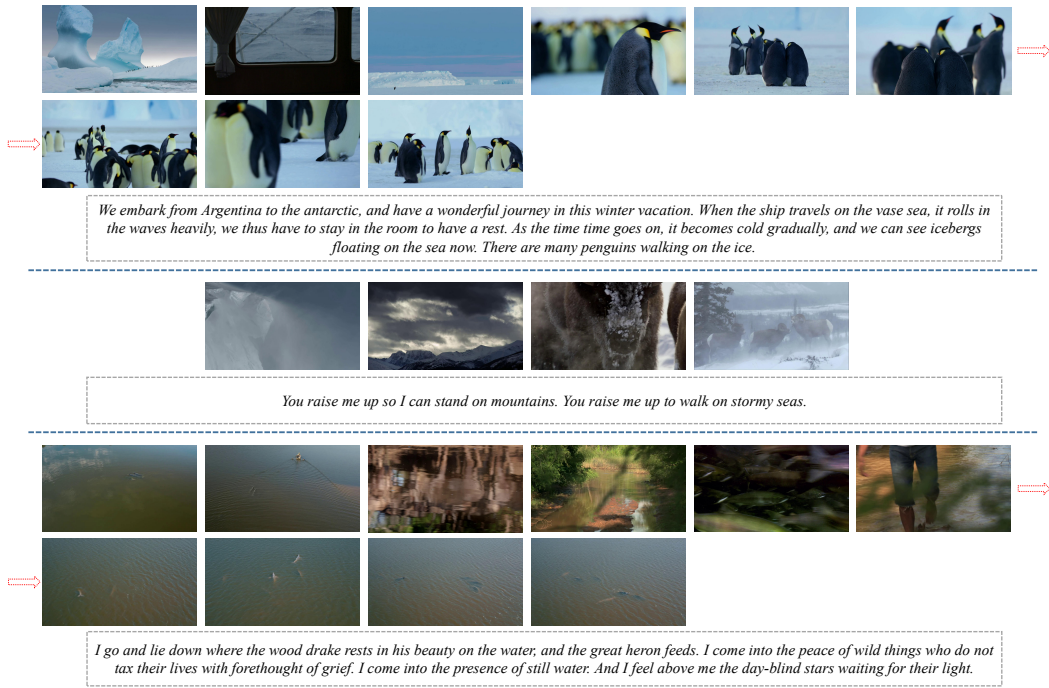


Figure 10: Samples of videos generated by our RATV with wild text scripts. Each row shows the generated video and its corresponding text script. Each picture denotes a shot in the video and the shots are arranged in chronological order from left to right in each row. The red arrow denotes the two shots are contiguous.

Video and Transcript-to-Video can both retrieve the shots that are well aligned with the text script, but they can not arrange the retrieved shots well. For example, Write-A-Video reverses the two shots from the ground-truth video and places the shot of penguin after that of dolphin (i.e., the fourth and the fifth shots). The possible reason for this is that the pre-defined rules used in Write-A-Video ignore the context of the input text script, which thus leads to misunderstanding the content of the shots and arranging them in an unreasonable order. Similarly, Transcript-to-Video also suffers from this problem. (4) Our RATV can generate the video that precisely expresses the whole text script. Furthermore, our model considers the semantic alignment and temporal coherence in the meantime during video montage generation. As a result, the video generated by our RATV does not contain any shots irrelevant to the text script, and the retrieved shots are also arranged in a reasonable order. Importantly, our model retrieve all shots (including the shot missed by the other competitors) from the ground-truth video. The corresponding generated video in Figure 9 and more videos generated by our RATV are available at the anonymous GitHub link (RATVDemo).

## F QUALITATIVE RESULTS FOR WILD TEXT SCRIPTS

Although the VSPD dataset contains the videos from documentaries, the goal of our proposed RATV is not to generate documentaries only. In contrast, it devotes to generating videos with various themes which mainly depend on the input text scripts. To demonstrate this, we consider the shots in the test set of VSPD as candidates (799 shots in total), and feed wild text scripts into our RATV to generate videos with different themes, which are shown in Figure 10. We find that our RATV can generate travel vlog with narrated text scripts (first row), MV for music with lyrics (second row), and background video with poem (last row), all of which are well aligned with the input text scripts. Note that some shots in these videos can not express some words precisely (e.g., the last two shots in the second video and the fifth shot in the last video) because there are no proper shots in the candidate set, which can be alleviated by adopting larger shot gallery during generation. In addition to these three themes, users can also adopt our RATV to generate videos on other diverse themes as they desired in practice. The corresponding generated videos in Figure 10 are available at the anonymous GitHub link (RATVDemo).

Table 4: Results for different length distributions of positive and negative samples.

Method	IoU $\uparrow$	UMS $\downarrow$	SMS $\uparrow$	CS $\uparrow$
Pos More	0.093	4.741	0.051	0.079
Neg More	0.109	4.513	0.056	0.131
RATV	<b>0.144</b>	<b>3.393</b>	<b>0.090</b>	<b>0.685</b>

Table 5: Length distributions of positive and negative samples under different settings.

Method	Positive	Negative
Pos More	$3.06 \pm 2.24$	$5.43 \pm 2.70$
Neg More	$2.57 \pm 1.97$	$6.84 \pm 2.28$
RATV	$2.57 \pm 1.97$	$5.43 \pm 2.70$

## G USER STUDY ON DATASET CONSTRUCTION

To show the impact of subjectivity of our VSPD dataset, we conduct the user study to investigate whether people can achieve the goal of video montage generation task well on the VSPD dataset. Note that it is very time-consuming for volunteers to select desired shots according to the give texts from all candidates in the test set of VSPD dataset, which consists of about 800 shots. Therefore, to make the user study more feasible, we randomly select 50 video-script pairs from test set to construct a subset for user study, which consists of 179 shots. We then invite 5 volunteers to manually select shots from candidates according to given texts, and arrange selected shots. Specifically, we shuffle the 50 texts and then averagely divide these texts into 5 parts. Each volunteer is given 10 texts in one part, and is asked to select shots from candidates according to each text and arrange the retrieved shots to form a consecutive video. The resultant IoU and SMS is  $0.728$  and  $0.558$ , respectively. Note that the volunteers are not the experts in video editing. Therefore, these two results indicate even inexperienced people that lack professional expertise can find the ground-truth shots from the candidates and arrange the retrieved shots in the correct order well. The results of this user study thus demonstrate that our proposed VSPD dataset is beneficial to evaluating a model’s ability of text-shots alignment and shot sequence arrangement. Furthermore, the videos in VSPD dataset are continue clips of documentaries, which are well edited by the experts that have professional expertise. Therefore, the impact of subjectivity for VSPD dataset is relatively minor.

## H IMPACT OF DIFFERENCE IN LENGTH DISTRIBUTION

To investigate how the difference in the length distributions affects the training, we conduct two experiments below: (1) Firstly, we modify the constraint of the random number  $L_0$  to  $0 \leq L_0 < n/2$ , so that the positive samples contain more shots (denoted as ‘Pos More’). (2) Secondly, we modify the constraint to the length of negative samples (denoted as ‘Neg More’). If the length of the selected unpaired shot sequence is less than (or equal to) 5, we directly insert the whole sequence into the paired shot sequence in a random position. If the length of the unpaired sequence is greater than 5, we follow the statement in Section 4.2 in the main paper to obtain a sub-sequence of it but guarantee the sum of  $k + L_1$  is greater than 5. The obtained results for differencnt length distribution are shown in Table 4. Furthermore, the mean and standard deviation of each length distribution is also shown in Table 5. We can observe that both reducing and expanding the difference between the distributions of the lengths of sequences of positive and negative samples make the results worse, which indicates that the difference in the length distributions indeed affects the training and the intuitive setting of our RATV can achieve the best results. In addition, these results also demonstrate that our RATV can indeed learn the knowledge to generate video montage well from give texts, but not shortcut about the difference in the length distributions.