SST-VLM: Sparse Sampling-Twice Inspired Video-Language Model

Yizhao Gao[1,2] and Zhiwu Lu[1,2](✉)

[1] Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
{gaoyizhao,luzhiwu}@ruc.edu.cn

[2] Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing, China

Abstract. Most existing video-language modeling methods densely sample dozens (or even hundreds) of video clips from each raw video to learn the video representation for text-to-video retrieval. This paradigm requires high computational overload. Therefore, sparse sampling-based methods are proposed recently, which only sample a handful of video clips with short time duration from each raw video. However, they still struggle to learn a reliable video embedding with fragmented clips per raw video. To overcome this challenge, we present a novel video-language model called SST-VLM inspired by a Sparse Sampling-Twice (SST) strategy, where each raw video is represented with only two holistic video clips (each has a few frames, but throughout the entire video). For training our SST-VLM, we propose a new Dual Cross-modal MoCo (Dual X-MoCo) algorithm, which includes two cross-modal MoCo modules to respectively model the two clip-text pairs (for each video-text input). In addition to the classic cross-modal contrastive objective, we devise a clip-level alignment objective to obtain more consistent retrieval performance by aligning the prediction distributions of the two video clips (based on the negative queues of MoCo). Extensive experiments show that our SST-VLM achieves new state-of-the-art in text-to-video retrieval.

1 Introduction

Video-language modeling has drawn great attention in recent years, because it is applicable to a wide variety of practical downstream tasks, including text-to-video retrieval [1–4], video captioning [1,5,6], and video question answering [7–9]. In this paper, we focus on text-to-video retrieval, and hopefully our work can bring some inspirations to other video-language tasks. Since raw videos consist of a series of image frames, processing these frames acquires tremendous computation cost and resource consumption. Therefore, how to efficiently and effectively utilize/integrate the video frames to obtain informative video representation has become a great challenge in text-to-video retrieval.

Existing approaches [10–19] address this challenge mainly by encoding each raw video with multiple sampled video clips. Most of them [10–14,16] sample video clips with short time duration (e.g., 1 s for each clip) from the raw video. Since such local clips can hardly represent the holistic content of the raw

L. Wang et al. (Eds.): ACCV 2022, LNCS 13844, pp. 537–553, 2023.
https://doi.org/10.1007/978-3-031-26316-3_32

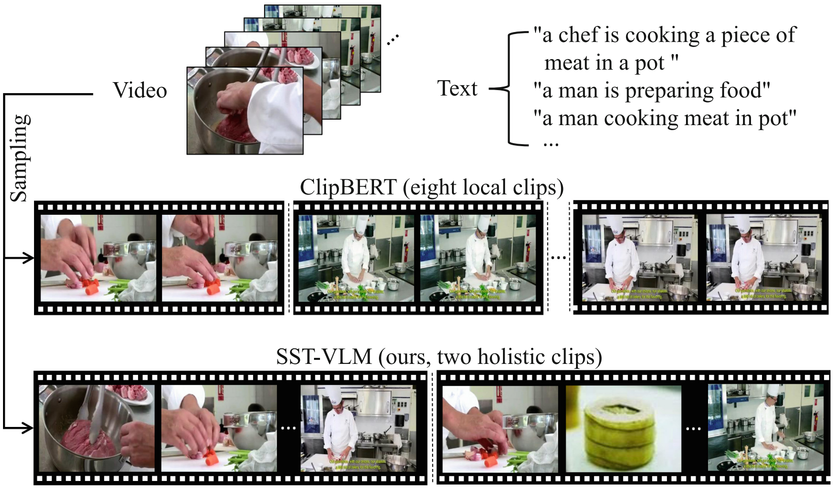
Video
Text
"a chef is cooking a piece of meat in a pot "
"a man is preparing food"
"a man cooking meat in pot"
...
Sampling
ClipBERT (eight local clips)
SST-VLM (ours, two holistic clips)

Fig. 1. Comparison among different sparse sampling strategies for text-to-video retrieval. We draw one video-text pair from the original dataset (1st row). Note that the video content of 'meat' (that appears in several captions) fails to be sampled in ClipBERT [10] (2nd row), but it is correctly sampled in our SST-VLM (3rd row).

video, these methods often sample them densely (i.e., sample a large number of local clips per raw video). Unlike these dense sampling methods, ClipBERT [10] applies a sparse sampling strategy to each raw video (i.e., only a handful of local clips are sampled), which has been reported to be effective. However, it still has limitations: the sampled local clips with short time duration are separately matched with the query text to obtain the clip-level predictions (before aggregated into the final video-level prediction), and thus the video content of some important concepts may be ignored by such sparse sampling strategy (see Fig. 1). Therefore, matching sampled local clips to the whole video description is not reliable for video-language modeling.

To overcome these limitations, in this work, we propose a new video sampling strategy named '**Sparse Sampling-Twice (SST)**' for text-to-video retrieval, which sparsely and holistically samples two video clips from each raw video. Our sampling strategy has two key characteristics: (1) *Sparse Random Sampling* – we first subdivide a raw video into a handful of equal segments and then randomly sample a single frame from each segment, resulting in a holistic video clip. (2) *Sampling-Twice* – since sampling only one holistic clip may ignore some key information of the raw video and make the video-text prediction unreliable, we propose to sample two holistic clips by imposing the same sparse random sampling strategy twice on each raw video. Note that we can easily sample more clips per raw video, but in this work, we focus on sampling-twice due to the GPU resource restriction. The detailed comparison between the sparse sampling strategies used in ClipBERT [10] and our SST strategy is shown in Fig. 1.

Inspired by our SST, we present a novel video-language model termed SST-VLM for text-to-video retrieval (see Fig. 2). To train our model, we propose a new

Dual Cross-modal MoCo (**Dual X-MoCo**) algorithm, which includes two cross-modal MoCo [20] modules to respectively model the two clip-text pairs for each video-text input. For the video clip, we employ a 2D image encoder (i.e., ViT-base [21]) to embed the sampled frames and obtain the video embedding by a Transformer [22] module. For the text description, we employ a text encoder (i.e., BERT-base [23]) to obtain its embedding. Note that making retrieval prediction with only one sparsely sampled clip is not reliable and the model's performance varies significantly across different sampled clips per raw video. Therefore, in addition to the classic cross-modal contrastive objective, we devise a new clip-level alignment objective to obtain more consistent retrieval performance based on the two video clips sampled by SST (per raw video). Concretely, in each training step, the retrieval distributions of the two video clips are aligned by minimizing the Kullback-Leibler (KL) divergence between them. Since the retrieval distributions have actually been computed during obtaining the cross-modal contrastive loss, our alignment objective almost requires no extra computation cost. Overall, our clip-level alignment objective enables our SST-VLM to achieve more consistent performance in text-to-video retrieval. Importantly, we find that it is even effective without using more frames per raw video (see the ablation study in Table 1).

Our main contributions are three-fold: (1) We present a novel video-language contrastive learning framework termed SST-VLM for text-to-video retrieval, which is inspired by the 'Sparse Sampling-Twice (SST)' strategy. Different from ClipBERT [10], our SST sparsely and holistically samples two video clips from each raw video. (2) We propose a new Dual X-MoCo algorithm for training our SST-VLM. It is seamlessly integrated with the SST strategy so that our model can achieve more stable as well as better performance. (3) Extensive experiments show that our SST-VLM achieves new state-of-the-art in text-to-video retrieval.

2 Related Work

Text-to-Video Retrieval. Text-to-video retrieval has recently become a popular video-language modeling task. Classic approaches [24,24–30] pre-extract the video features using expert models, including those trained on other tasks such as object recognition, and action classification. They also pre-extract the text features using pre-trained language models [23,31]. The major drawback of this paradigm is the lack of cross-modal interaction during feature pre-extraction. To tackle this problem, a number of works [10–14,16,18,32] have proposed to train video-language models without using pre-extracted features. Most of them [11–14,16] embed each raw video with densely sampled video clips. Different from such costly dense sampling, ClipBERT [10] introduces a sparse sampling strategy, which samples a handful of video clips with short-time duration to learn the video representation. In this work, instead of sampling locally multiple times (like ClipBERT), our SST-VLM proposes a Sparse Sampling-Twice strategy which sparsely and holistically samples two video clips from each raw video. Importantly, we choose to align the clip-level prediction distributions in the retrieval task to obtain more reliable video embeddings.

Contrastive Learning. Contrastive learning has achieved great success in visual recognition [20,33–39]. The infoNCE loss [33] has been widely used for contrastive learning, where a large number of negative samples are proven to be crucial for better performance. There are two ways of collecting negative samples: (1) SimCLR [37] utilizes the augmented view of each sample to be the positive sample and all other samples in the current batch to be negative ones. (2) MoCo [20] and its variant [40] introduce a momentum mechanism to maintain a large negative queue. Since MoCo can decouple the number of negative samples from the batch size, MoCo-based models are applicable to the setting with a small total batch size (less GPUs are needed for training). In this work, we thus choose to employ MoCo for video-language modeling. Interestingly, we have also explored BYOL [39] and SimSiam [41] in text-to-video retrieval, but found that they fail without using negative samples.

Note that contrastive learning has already been applied to text-to-video retrieval in the latest works [28,30]. Concretely, TACo [28] adopts token-aware cascade contrastive learning (enhanced with hard negative mining), and HiT [30] introduces cross-modal MoCo based on both feature-level and semantic-level matching. They both utilize pre-extracted features as their inputs, leading to suboptimal results. Different from them, in this work, we focus on learning better video/text embeddings directly from *raw* videos/texts by proposing a new Dual X-MoCo algorithm, which includes two cross-modal MoCo modules with both cross-modal contrastive loss and clip-level alignment loss.

3 Methodology

3.1 Framework Overview

The flowchart of our SST-VLM model for text-to-video retrieval is illustrated in Fig. 2. Concretely, we are given a training set of N video-text pairs $\mathcal{D} = \{V_i, T_i\}_{i=1}^{N}$, where each video V_i has S_i frames of resolution $H \times W$ and each text T_i is represented by the natural language in English. Our model aims to learn a video encoder f_{θ_v} and a text encoder f_{θ_t} to project each video and its paired text into the joint embedding space so that they can be aligned with each other. The video and text encoders of our model are presented in Sect. 3.2, and our proposed Dual X-MoCo algorithm for model training is given in Sect. 3.3.

3.2 Video and Text Encoders

Video Encoder. Given a raw video V_i, we randomly and sparsely sample $N_c = 2$ video clips $\{c_{i,r}\}_{r=1}^{N_c}$, with s ($s < S_i$) frames per video clip (see Sect. 3.3 for more details). For each sampled video clip $c_{i,r}$, we first extract the visual embeddings $F_{i,r}^{v} \in \mathbb{R}^{s \times D_v}$ of all frames through an pre-trained image encoder f^v (e.g., the ViT-base model [21]) with an output dimension D_v:

$$F_{i,r}^{v}[k] = f^{v}(c_{i,r}[k]), k = 1, \cdots, s, \tag{1}$$

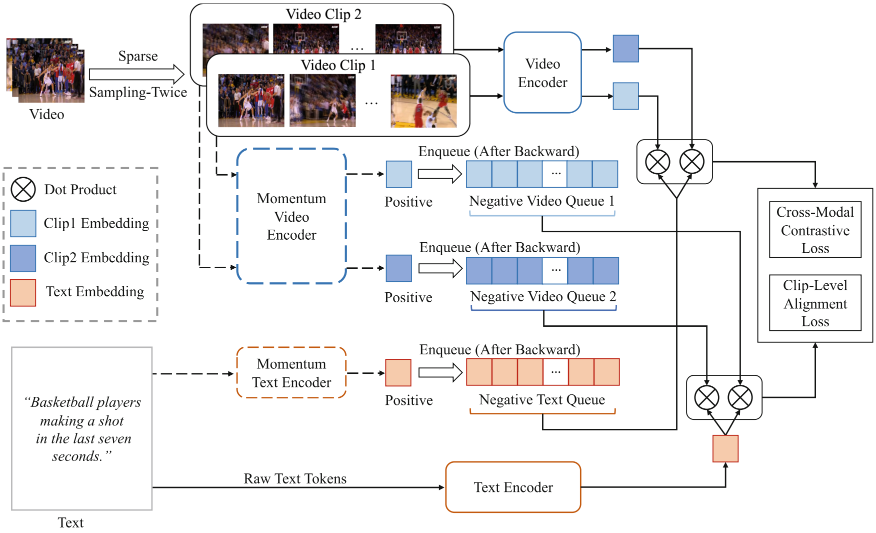

Fig. 2. Schematic illustration of our SST-VLM model. We sparsely and randomly sample two video clips from each raw video. Inspired by this 'Sparse Sampling-Twice' strategy, we devise a Dual X-MoCo algorithm to train our model. The video encoder consists of a ViT-base model followed by a Transformer module, and the text encoder is a BERT-base model. The momentum video/text encoders marked with dotted lines are initialized by the video/text encoders and updated by Eqs. (8) and (9).

where $c_{i,r}[k]$ denotes the k-th frame of the video clip $c_{i,r}$ and $F^v_{i,r}[k]$ denotes the k-th row of $F^v_{i,r}$. We then utilize a Transformer [22] module f_{att} to capture the temporal correlation across all frame embeddings:

$$\widehat{F}^v_{i,r} = f_{att}(F^v_{i,r}[1], F^v_{i,r}[2], \cdots, F^v_{i,r}[s]), \tag{2}$$

where $\widehat{F}^v_{i,r} \in \mathbb{R}^{s \times D_v}$ are the embeddings output by the Transformer. We finally obtain the embedding $F^c_{i,r} \in \mathbb{R}^D$ of video clip $c_{i,r}$ by averaging all frame embeddings, which is followed by a linear projection layer:

$$F^c_{i,r} = \text{Linear}(\text{Avg}(\widehat{F}^v_{i,r}[1], \widehat{F}^v_{i,r}[2], \cdots, \widehat{F}^v_{i,r}[s])), \tag{3}$$

where $\widehat{F}^v_{i,r}[k]$ is the k-th row of $\widehat{F}^v_{i,r}$ and D is the common dimension for video and text embeddings. Linear$(\cdot)$ denotes a linear projection layer and Avg$(\cdot)$ denotes the average pooling function that computes the mean of input embeddings. Overall, our video encoder f_{θ_v} (with parameters θ_v) encodes video clips by Eqs. (1)–(3).

Text Encoder. For each text T_i, we first tokenize it into a sequence of tokens $[t^i_1, t^i_2, \cdots, t^i_{l_i}]$, where l_i denotes the length of T_i. A pre-trained language model f^t (e.g., the BERT-base model [23]) is then used to map the sequence of tokens to a D_t-dimensional embedding $F^t_i \in \mathbb{R}^{D_t}$:

$$F^t_i = f^t(t^i_1, t^i_2, \cdots, t^i_{l_i}). \tag{4}$$

Similar to Eq. (3), we obtain the final text embedding by:

$$F_i^t = \text{Linear}(F_i^t), \tag{5}$$

where Linear$(\cdot)$ is a linear layer with the output dimension D. Overall, our text encoder f_{θ_t} (with parameters θ_t) encodes raw texts by Eqs. (4) and (5).

3.3 Dual X-MoCo

Sparse Sampling-Twice. In this work, to overcome the limitations of ClipBERT [10] (see Fig. 1), we propose to sparsely and holistically sample two video clips from each raw video, noted as the 'Sparse Sampling-Twice' strategy. Specifically, given a raw video V_i with S_i frames from a mini-batch $\mathcal{B} = \{V_i, T_i\}_{i=1}^B$, we first subdivide it into s equal segments, where $s = 4$ in our implementation. We then randomly sample one frame from each segment to form the first video clip $c_{i,1}$. The second video clip $c_{i,2}$ is obtained by the same random sampling strategy. Finally, we feed the two sparsely sampled video clips $c_{i,1}, c_{i,2}$ and paired text T_i separately into the video encoder and text encoder to obtain video clip embeddings $F_{i,1}^c$, $F_{i,2}^c$ and text embedding F_i^t:

$$F_{i,1}^c = f_{\theta_v}(c_{i,1}), F_{i,2}^c = f_{\theta_v}(c_{i,2}), F_i^t = f_{\theta_t}(T_i). \tag{6}$$

Cross-Modal Contrastive Loss. As shown in Fig. 2, our Dual X-MoCo includes two cross-modal MoCo [20] modules to respectively model the sampled two clip-text pairs for each video-text input. Concretely, in a mini-batch $\mathcal{B} = \{V_i, T_i\}_{i=1}^B$, we first sample two video clips $c_{i,1}$ & $c_{i,2}$ from each video V_i by our SST strategy. Further, we encode the video clips $c_{i,1}$, $c_{i,2}$ and paired text T_i by Eq. (6). After that, we obtain the key embeddings $K_{i,1}^c$, $K_{i,2}^c$, K_i^t of $c_{i,1}$, $c_{i,2}$, T_i by the momentum encoders $f_{\theta_v^m}$ and $f_{\theta_t^m}$:

$$K_{i,1}^c = f_{\theta_v^m}(c_{i,1}), K_{i,2}^c = f_{\theta_v^m}(c_{i,2}), K_i^t = f_{\theta_t^m}(T_i), \tag{7}$$

where $f_{\theta_v^m}$ (with parameters θ_v^m) is initialized by the video encoder f_{θ_v} and $f_{\theta_t^m}$ (with parameters θ_t^m) is initialized by the text encoder f_{θ_t}. During training, the parameters of the momentum encoders $f_{\theta_v^m}$ and $f_{\theta_t^m}$ are updated by the momentum-update rule as follows:

$$\theta_v^m = m \cdot \theta_v^m + (1 - m) \cdot \theta_v, \tag{8}$$

$$\theta_t^m = m \cdot \theta_t^m + (1 - m) \cdot \theta_t, \tag{9}$$

where m is a momentum coefficient. After loss calculation, with the earliest B momentum embeddings popped out, we push $\{K_{i,1}^c\}_{i=1}^B$, $\{K_{i,2}^c\}_{i=1}^B$, and $\{K_i^t\}_{i=1}^B$ respectively into queues Q_1^c, Q_2^c, and Q^t, where $Q_1^c = \{n_{j,1}^c\}_{j=1}^{N_q}$, $Q_2^c = \{n_{j,2}^c\}_{j=1}^{N_q}$, and $Q^t = \{n_j^t\}_{j=1}^{N_q}$. Note that the queue size N_q is decoupled from B.

As a result, for each query text embedding F_i^t, we have positive video clip embeddings $K_{i,1}^c$, $K_{i,2}^c$ and negative video clip embeddings $n_{j,1}^c \in Q_1^c$, $n_{j,2}^c \in Q_2^c$.

We thus define the text-to-video contrastive loss L_{t2v} as an InfoNCE-based loss:

$$P_{i,1} = e^{F_i^t \cdot K_{i,1}^c/\tau}, \quad P_{i,2} = e^{F_i^t \cdot K_{i,2}^c/\tau}, \tag{10}$$

$$N_{i,1} = \sum_{j=1}^{N_q} e^{F_i^t \cdot n_{j,1}^c/\tau}, \quad N_{i,2} = \sum_{j=1}^{N_q} e^{F_i^t \cdot n_{j,2}^c/\tau}, \tag{11}$$

$$L_{t2v} = -\frac{1}{B}\sum_{i=1}^{B}(\log\frac{P_{i,1}}{P_{i,1}+N_{i,1}} + \log\frac{P_{i,2}}{P_{i,2}+N_{i,2}}), \tag{12}$$

where τ is the temperature, $P_{i,1}$ (or $P_{i,2}$) is the similarity score between positive clip embedding $K_{i,1}^c$ (or $K_{i,2}^c$) and query text embedding F_i^t. Additionally, $N_{i,1}$ (or $N_{i,2}$) is the summed similarity score between negative clip embeddings $n_{j,1}^c$ (or $n_{j,2}^c$) and query text embedding F_i^t. Similarly, for query video clips with embeddings $F_{i,1}^c$, $F_{i,2}^c$ computed by Eq. (6) and their positive/negative text embeddings K_i^t, n_j^t, we define the video-to-text contrastive loss L_{v2t} as follows:

$$\widehat{P}_{i,1} = e^{F_{i,1}^c \cdot K_i^t/\tau}, \quad \widehat{P}_{i,2} = e^{F_{i,2}^c \cdot K_i^t/\tau}, \tag{13}$$

$$\widehat{N}_{i,1} = \sum_{j=1}^{N_q} e^{F_{i,1}^c \cdot n_j^t/\tau}, \quad \widehat{N}_{i,2} = \sum_{j=1}^{N_q} e^{F_{i,2}^c \cdot n_j^t/\tau}, \tag{14}$$

$$L_{v2t} = -\frac{1}{B}\sum_{i=1}^{B}(\log\frac{\widehat{P}_{i,1}}{\widehat{P}_{i,1}+\widehat{N}_{i,1}} + \log\frac{\widehat{P}_{i,2}}{\widehat{P}_{i,2}+\widehat{N}_{i,2}}). \tag{15}$$

The total cross-modal contrastive loss L_{cl} is given by:

$$L_{cl} = L_{t2v} + L_{v2t}. \tag{16}$$

Clip-Level Alignment Loss. Note that different video clips sampled from the same video should correspond to the same ground-truth text. Therefore, a good model should have consistent retrieval performance over different video clips sampled from each video. To this end, in addition to the cross-modal contrastive loss, we propose to enhance the performance consistency of our SST-VLM model by minimizing a clip-level alignment loss defined with the Symmetrical Kullback-Leibler (Sym-KL) divergence.

Given a video V_i with two video clips $c_{i,r}$ ($r = 1, 2$) and its paired text T_i, the video-to-text retrieval probability distribution of the query video clip is denoted as $\widehat{U}_{i,r} = [\hat{u}_{i,r}^0, \cdots, \hat{u}_{i,r}^{N_q}] \in \mathbb{R}^{N_q+1}$. Similarly, the text-to-video retrieval probability distribution of the query text is $U_{i,r} = [u_{i,r}^0, \cdots, u_{i,r}^{N_q}] \in \mathbb{R}^{N_q+1}$. Concretely, the j-th element ($j = 0, \cdots, N_q$) of $\widehat{U}_{i,r}$ or $U_{i,r}$ is defined as:

$$\hat{u}_{i,r}^j = \frac{\widehat{P}_{i,r}}{\widehat{P}_{i,r}+\widehat{N}_{i,r}}, \quad u_{i,r}^j = \frac{P_{i,r}}{P_{i,r}+N_{i,r}}, (j = 0), \tag{17}$$

$$\hat{u}_{i,r}^j = \frac{e^{F_{i,r}^c \cdot n_j^t/\tau}}{\widehat{P}_{i,r}+\widehat{N}_{i,r}}, \quad u_{i,r}^j = \frac{e^{F_i^t \cdot n_{j,r}^c/\tau}}{P_{i,r}+N_{i,r}}, (j > 0), \tag{18}$$

where $\widehat{P}_{i,r}$, $\widehat{N}_{i,r}$, $P_{i,r}$, $N_{i,r}$ have been computed in Eqs. (13), (14), (10) and (11). We then define the clip-level alignment loss L_{al} with the Sym-KL divergence:

$$L_{\hat{u}} = \frac{1}{B}\sum_{i=1}^{B}\sum_{j=0}^{N_q}(\hat{u}_{i,1}^{j}\log\frac{\hat{u}_{i,1}^{j}}{\hat{u}_{i,2}^{j}} + \hat{u}_{i,2}^{j}\log\frac{\hat{u}_{i,2}^{j}}{\hat{u}_{i,1}^{j}}), \tag{19}$$

$$L_{u} = \frac{1}{B}\sum_{i=1}^{B}\sum_{j=0}^{N_q}(u_{i,1}^{j}\log\frac{u_{i,1}^{j}}{u_{i,2}^{j}} + u_{i,2}^{j}\log\frac{u_{i,2}^{j}}{u_{i,1}^{j}}), \tag{20}$$

$$L_{al} = L_{\hat{u}} + L_{u}. \tag{21}$$

Total Loss. Our SST-VLM model is trained by minimizing both the cross-modal contrastive loss and clip-level alignment loss. We thus have the total loss:

$$L_{total} = L_{cl} + \lambda * L_{al}, \tag{22}$$

where λ is the weight hyper-parameter.

3.4 Model Pre-training

Note that our model can be readily applied to the image-text retrieval task when the temporal Transformer module is removed. Therefore, similar to ClipBERT [10], our model (excluding the temporal Transformer module) is pre-trained on a widely-used image-text dataset (with overall 5.3M image-text pairs), which consists of CC3M [42], VisGenome [43], SBU [44], COCO [45], and Flickr30k [46]. In this work, we do not pre-train our model on a large-scale external video-text dataset like HowTo100M [12] due to the limited computation resources. Although only pre-trained on an image-text dataset rather than a large-scale video-text dataset, our model still achieves new state-of-the-art on several benchmark datasets for text-to-video retrieval (see Table 3).

4 Experiments

4.1 Datasets and Settings

Datasets. We evaluate our SST-VLM on three benchmarks: (1) **MSR-VTT** [1] contains 10k videos with 200k descriptions. We first follow recent works [18,24,47], using the 1k-A split of 9k training videos and 1k test videos. Further, we also adopt the split in [10,28] (called 7k-1k split in our work), having 7k training and 1k test videos. (2) **MSVD** [2] consists of 80k English descriptions for 1,970 videos from YouTube, and each video has around 40 captions. As in [18,29,47], we use the standard split: 1,200 videos for training, 100 videos for validation, and 670 ones for test. (3) **VATEX** [3] includes 25,991 videos for training, 3000 videos for validation, and 6000 ones for test. Since the original test set is private, we follow [29,48] to randomly split the original val set into two equal parts with 1500 videos for validation and the other 1500 videos for test.

Table 1. Ablation study for our SST-VLM model. Text-to-video retrieval results are reported on the MSR-VTT 1K-A test set.

L_{cl}	L_{al}	Frames	R@1 ↑	R@5 ↑	R@10 ↑
×	×	4	31.4	59.5	69.8
×	×	8	31.3	59.3	70.1
✓	×	4 + 4	32.1	61.5	71.8
✓	✓	4 + 4	**33.4**	**62.5**	**73.5**

Evaluation Metrics. We evaluate the text-to-video retrieval performance with the widely-used evaluation metrics in information retrieval, including Recall at K (shortened as R@K with K = 1, 5, 10) and Median Rank (shortened as MedR). R@K refers to the percentage of queries that are correctly retrieved in the top-K results. MedR measures the median rank of correct answers in the retrieved ranking list, where lower score indicates better performance.

Implementation Details. We adopt ViT-base [21] as the frame feature extractor of our video encoder and BERT-base [23] as the text encoder. For visual augmentation at the training stage, we apply random-crop, gray-scaling, horizontal-flip, and color-jitter to the input video frames that are resized to 384×384 (but only frame-resizing and center-crop are deployed at the evaluation stage). Due to the computation constraint, we empirically set the hyperparameters as: $\tau = 1$, $\lambda = 0.1$, and the initial learning rate is 5e−5. We only update the last 8 layers of the video and text encoders (but the other layers are frozen) during training. It takes about 2 h per epoch to train our model on MSR-VTT with 8 T V100 GPUs. In addition, different from the SST strategy (4 frames per clip) used for model training, we sample two video clips with 8 frames from each video V_i at the evaluation stage. With two clip embeddings $F^c_{i,1}$, $F^c_{i,2}$ obtained by Eq. (6), we have the final embedding of V_i for evaluation by averaging $F^c_{i,1}$ and $F^c_{i,2}$.

4.2 Ablation Study

Contributions of Contrastive and Alignment Losses. We analyze the contributions of cross-modal contrastive loss L_{cl} and clip-level alignment loss L_{al} used in our SST-VLM. The obtained ablative results are shown in Table 1. The baseline model (1st row) is formed with a single cross-modal MoCo framework where only one video clip (with $frames = 4$) is sampled for each video. Based on our SST strategy (with $frames = 4 + 4$), we obtain another baseline method by removing the clip-level alignment loss L_{al} from our Dual X-MoCo (3rd row). To further demonstrate the effectiveness of our full model, we train a baseline model (2nd row) based on single cross-modal MoCo with $frames = 8$. We can observe that: (1) Sampling one video clip with 4 or 8 frames leads to comparable performance (2nd row vs. 1st row). (2) With our SST strategy (4 + 4 frames per video), our model achieves significant improvements (3rd row vs. 1st/2nd row). This suggests that our SST-VLM model is even effective without using more frames per raw video. (3) The clip-level alignment loss can further improve

Table 2. Comparative results obtained by different alignment methods used in Eq. (21). Text-to-video retrieval results are reported on the MSR-VTT 1K-A test set.

Alignment method	R@1 ↑	R@5 ↑	R@10 ↑
NC	32.8	62.0	71.7
L2	32.4	62.2	71.5
Asym-KL	33.2	62.2	72.1
Sym-KL (ours)	**33.4**	**62.5**	**73.5**

the performance of our SST-VLM model, yielding around 1% improvement on all R@K (K = 1, 5, 10) results over our SST-VLM model with only contrastive loss (4th row vs. 3rd row).

Comparison to Alternative Alignment Methods. We further analyze the impact of alternative methods used for our clip-level alignment loss in Table 2. Note that the alignment loss L_{al} in Eq. (16) is defined with the Symmetric Kullback-Leiber (Sym-KL) divergence. This Sym-KL distance can be replaced by the negative cosine similarity (NC), L2 distance, or Asymmetric KL (Asym-KL) divergence. Concretely, the alternative alignment losses are defined by:

$$L_{al}^{NC} = -\frac{1}{B}\sum_{i=1}^{B}\left(\frac{\widehat{U}_{i,1}\cdot\widehat{U}_{i,2}}{||\widehat{U}_{i,1}||_2||\widehat{U}_{i,2}||_2} + \frac{U_{i,1}\cdot U_{i,2}}{||U_{i,1}||_2||U_{i,2}||_2}\right), \tag{23}$$

$$L_{al}^{L2} = \frac{1}{B}\sum_{i=1}^{B}(||\widehat{U}_{i,1} - \widehat{U}_{i,2}||_2 + ||U_{i,1} - U_{i,2}||_2), \tag{24}$$

$$L_{al}^{Asym} = \frac{1}{B}\sum_{i=1}^{B}\sum_{j=0}^{N_q}(\hat{u}_{i,1}^{j}\log\frac{\hat{u}_{i,1}^{j}}{\hat{u}_{i,2}^{j}} + u_{i,1}^{j}\log\frac{u_{i,1}^{j}}{u_{i,2}^{j}}), \tag{25}$$

where $\widehat{U}_{i,r}$, $U_{i,r}$, $\hat{u}_{i,r}$, $u_{i,r}$ are defined in Eqs. (17) and (18). We can find that SST-VLM with NC, L2 or Asym-KL leads to slightly lower performance on R@1 and R@5, and nearly 2% performance degradation on R@10, as compared with our SST-VLM using Sym-KL. We thus choose Sym-KL in this work.

4.3 Comparative Results

Table 3 shows the comparative results for text-to-video retrieval on MSR-VTT. We compare our SST-VLM with a wide range of representative/state-of-the-art methods including those [27–29] pre-trained on HowTo100M and those [10,18] pre-trained on image-text datasets. For extensive comparison, we also include methods [24,30,47] that utilize pre-extracted expert features. Although our SST-VLM is pre-trained on the smallest dataset with only 5.3M image-text pairs, it still achieves the best performance under both 7k-1k and 1k-A splits, demonstrating the effectiveness of our Dual X-MoCo for video-language modeling. Concretely, under the 7k-1k split, our SST-VLM outperforms the second best competitor by 5.0% on R@1, 5.9% on R@5, and 4.2% on R@10. It also leads to the

Table 3. Comparison to the state-of-the-art results for text-to-video retrieval on the MSR-VTT test set. **w/o PE**: methods trained without using multi-modal pre-extracted features. **VLM PT**: datasets for pre-training visual-language models. **VL Pairs**: the number of visual-language pairs in the pre-training datasets.

Method	w/o PE	VLM PT	VL Pairs	R@1 ↑	R@5 ↑	R@10 ↑	MedR ↓
7k-1k Split							
JSFusion [11]	✓	–	–	10.2	31.2	43.2	13.0
HT MIL-NCE [12]	✓	HowTo100M	>100M	14.9	40.2	52.8	9.0
ActBERT [13]	✓	HowTo100M	>100M	16.3	42.8	56.9	10.0
HERO [14]	✓	HowTo100M	>100M	16.8	43.4	57.7	–
VidTranslate [16]	✓	HowTo100M	>100M	14.7	–	52.8	–
NoiseEstimation [26]		HowTo100M	>100M	17.4	41.6	53.6	8.0
UniVL [27]		HowTo100M	>100M	21.2	49.6	63.1	6.0
ClipBERT [10]	✓	COCO, VisGenome	5.6M	22.0	46.8	59.9	6.0
TACo [28]		HowTo100M	>100M	24.8	52.1	64.5	5.0
SST-VLM (Ours)	✓	CC3M, Others	5.3M	**29.8**	**58.0**	**68.7**	**3.0**
1k-A Split							
CE [47]		–	–	20.9	48.8	62.4	6.0
AVLnet [25]		HowTo100M	>100M	27.1	55.6	66.6	4.0
MMT [24]		HowTo100M	>100M	26.6	57.1	69.6	4.0
Support Set [29]		HowTo100M	>100M	30.1	58.5	69.3	3.0
HiT [30]		HowTo100M	>100M	30.7	60.9	73.2	**2.6**
TACo [28]		HowTo100M	>100M	28.4	57.8	71.2	4.0
Frozen in Time [18]	✓	CC3M, WebVid-2M	5.5M	31.0	59.5	70.5	3.0
SST-VLM (ours)	✓	CC3M, Others	5.3M	**33.4**	**62.5**	**73.5**	3.0

Table 4. Comparison to the state-of-the-arts on MSVD for text-to-video retrieval.

Method	R@1 ↑	R@5 ↑	R@10 ↑	MedR ↓
VSE [49]	12.3	30.1	42.3	14.0
VSE++ [50]	15.4	39.6	53.0	9.0
Multi. Cues [51]	20.3	47.8	61.1	6.0
CE [47]	19.8	49.0	63.8	6.0
Support Set [29]	28.4	60.0	72.9	4.0
Frozen in Time [18]	33.7	64.7	76.3	3.0
SST-VLM (ours)	**36.2**	**66.4**	**76.9**	**2.0**

best MedR = 3.0. Moreover, under the 1k-A split (with more training data than the 7k-1k split), our SST-VLM outperforms the latest state-of-the-arts [18,30] by 2.4% on R@1 and 1.6% on R@5. In particular, as compared with HiT [30], our SST-VLM achieves better results on R@1 and R@5, and obtains competitive results on R@10 and MedR. This is still impressive and remarkable, given that HiT not only is pre-trained on the much larger dataset HowTo100M but also utilizes numerous pre-extracted expert features.

Table 4 shows the comparative results on MSVD. Our SST-VLM outperforms all competitors, especially yielding 2.5% margin on R@1 against the latest stat-of-the-art [18]. The results on VATEX in Table 5 further demonstrate that our

Table 5. Comparison to the state-of-the-arts on VATEX for text-to-video retrieval.

Method	R@1 ↑	R@5 ↑	R@10 ↑	MedR ↓
VSE [49]	28.0	64.3	76.9	3.0
VSE++ [50]	33.7	70.1	81.0	2.0
Dual [51]	31.1	67.4	78.9	3.0
HGR [47]	35.1	73.5	83.5	2.0
HANet [19]	36.4	74.1	84.1	2.0
Support Set [29]	45.9	82.4	90.4	**1.0**
SST-VLM (ours)	**53.4**	**85.3**	**92.0**	**1.0**

Table 6. Comparison to the state-of-the-arts on MSR-VTT (1k-A split) for video-to-text retrieval.

Method	R@1 ↑	R@5 ↑	R@10 ↑	MedR ↓
CE [47]	20.9	48.8	62.4	6.0
AVLnet [25]	28.5	54.6	65.2	4.0
MMT [51]	28.0	57.5	69.7	3.7
Support Set [29]	28.5	58.6	71.6	**3.0**
SST-VLM (ours)	**33.2**	**61.2**	**72.0**	**3.0**

Table 7. Comparison to the state-of-the-arts on MSVD for video-to-text retrieval.

Method	R@1 ↑	R@5 ↑	R@10 ↑	MedR ↓
VSE++ [50]	21.2	43.4	52.2	9.0
Multi. Cues [51]	31.5	51.0	61.5	5.0
Support Set [29]	34.7	59.9	70.0	3.0
SST-VLM (ours)	**47.3**	**72.0**	**78.0**	**2.0**

SST-VLM achieves 7.5% improvement on R@1, 2.9% improvement on R@5, and 1.6% improvement on R@10 over the second best competitor [29].

For comprehensive comparison, we also provide video-to-text retrieval results on the MSR-VTT 1k-A split in Table 6, in addition to the text-to-video retrieval results. Our SST-VLM outperforms the latest state-of-the-art (i.e., Support Set [29] pre-trained on Howto100M [12]) by 4.7% on R@1, 2.6% on R@5, and 0.4% on R@10. It also achieves the best MedR = 3.0. Moreover, we present the results for video-to-text retrieval on the MSVD [2] test set in Table 7. Our SST-VLM outperforms the second best competitor [29] by 12.6% on R@1, 12.1% on R@5, and 8.0% on R@10. It also leads to the best MedR = 2.0. These results indicate that our SST-VLM is effective for video-language modeling on both video-to-text and text-to-video retrieval tasks.

4.4 Visualization Results

Retrieval Rank Distribution. To show the stability of our SST-VLM, we visualize the text-to-video retrieval rank results on MSR-VTT 1k-A test set in

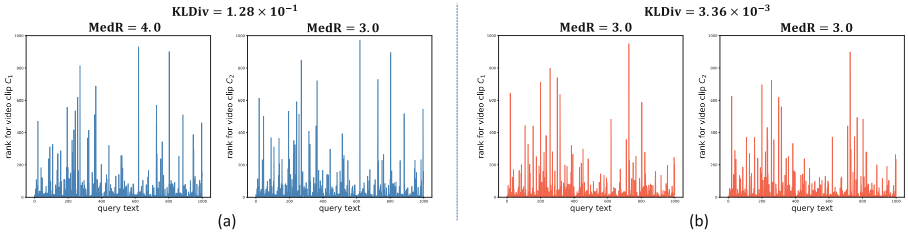

Fig. 3. Visualization results of text-to-video retrieval on the MSR-VTT 1k-A test set. (a) The results by our SST-VLM without using the clip-level alignment loss (but with cross-modal contrastive loss); (b) The results by our full SST-VLM.

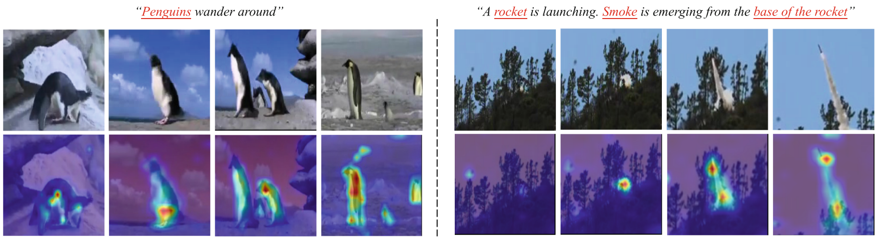
"Penguins wander around"
"A rocket is launching. Smoke is emerging from the base of the rocket"

Fig. 4. Attention visualization for our SST-VLM. The (attention) heatmaps are shown for two video-text pairs sampled from the MSR-VTT test set. Texts in *red* denote key objects for each video caption. (Color figure online)

Fig. 3. Note that we still sample two video clips (as in the training stage) from each raw video in the test set, resulting in two video clip sets $C_1 = \{c_{i,1}\}_{i=1}^{1,000}$ and $C_2 = \{c_{i,2}\}_{i=1}^{1,000}$. To show the effectiveness of our clip-level alignment loss L_{al} in enhancing stability, we evaluate two related models: model in Fig. 3(a) is trained without L_{al} (but with cross-modal contrastive loss L_{cl}), while model in Fig. 3(b) is exactly our full SST-VLM model. For each model, we visualize the retrieval rank (range from 1 to 1,000) distribution between video clip set C_1 (or C_2) and the same set of text queries. In addition, we report the MedR results for each distribution and KL divergence (KLDiv) between distributions of differently sampled video clips for each model. We find that: (1) The MedR results in Fig. 3(b) are equal to 3.0 for both C_1 and C_2, while those in Fig. 3(a) are different (4.0 for C_1 and 3.0 for C_2). (2) The KL divergence in Fig. 3(b) is two orders of magnitude smaller than that in Fig. 3(a). Therefore, the clip-level alignment loss L_{al} indeed leads to more stable results.

Attention Visualization. To further show that our SST-VLM has learned to understand the semantic content in videos, we adopt a recent Transformer visualization method [52] to highlight the relevant regions of the input frames according to the input texts. In this work, different from the original visualization method that computes the gradients directly from the total loss backward, we compute the separate gradients of each input frame and visualize the attention

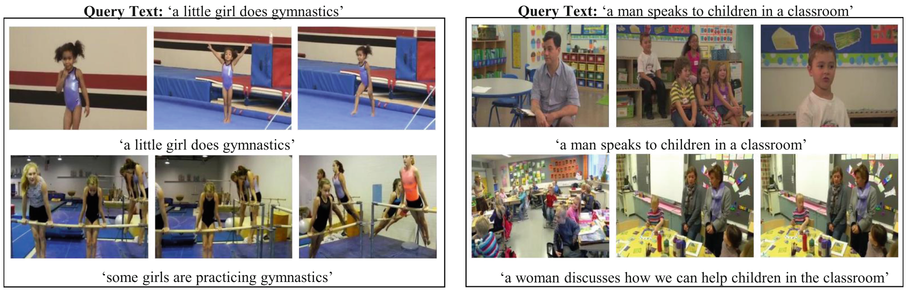
Query Text: 'a little girl does gymnastics'
'a little girl does gymnastics'
'some girls are practicing gymnastics'
Query Text: 'a man speaks to children in a classroom'
'a man speaks to children in a classroom'
'a woman discusses how we can help children in the classroom'

Fig. 5. Text-to-video retrieval examples obtained by our SST-VLM on the MSR-VTT 1k-A test set. For each query text, we visualize the top-2 retrieved videos (with 3 frames shown per video). We also present the *original paired text* under each video.

maps of all frames. Concretely, as shown in Fig. 4, we present two video-text pairs (and their visualization results) sampled from the MSR-VTT test set. The left part presents a 4-frame video clip with text 'Penguins wander around'. The attention visualization shows that penguins in all frames have actually been noticed by our model. Moreover, the right part presents a 4-frame video clip with a longer text 'A rocket is launching. Smoke is emerging from the base of the rocket'. The attention visualization is rather interesting: with the rocket launching, our model pays more attention to the rocket and its smoke. Overall, these visualization results indicate that our SST-VLM has actually learned to understand the semantic content in videos.

Text-to-Video Retrieval Examples. Figure 5 shows the text-to-video retrieval qualitative results obtained by our SST-VLM on the MSR-VTT 1k-A test set. We visualize the top-2 videos (with 3 frames show per video) for each query text. Concretely, the left part of Fig. 5 consists of a query text 'a little girl does gymnastics' and the retrieved top-2 videos (with their original paired texts) shown below, while the right part of Fig. 5 is organized similarly. For each query text, we have the following observations: (1) The ground-truth video is correctly retrieved at the first place. (2) The texts of the second retrieved videos are also similar to the query text, which means that the semantic content of these videos is still consistent with the query text. Overall, these qualitative results indicate that our SSL-VLM has indeed aligned the video and text embeddings well in the learned joint space (which is crucial for video-text retrieval).

5 Conclusion

In this paper, we propose a novel video-language model called SST-VLM inspired by the Sparse Sampling-Twice (SST) strategy that sparsely and holistically samples two video clips from each raw video. For training our SST-VLM, we devise a new Dual X-MoCo algorithm, which includes both cross-modal contrastive and clip-level alignment losses to enhance the performance stability of our model.

Extensive results on several benchmarks show that our SST-VLM achieves new state-of-the-art in text-to-video retrieval. The ablation study and attention visualization further demonstrate the effectiveness of our SST-VLM. In our ongoing research, we will apply our SST-VLM to other video-language understanding tasks such as video captioning and video question answering.

References

1. Xu, J., Mei, T., Yao, T., Rui, Y.: MSR-VTT: a large video description dataset for bridging video and language. In: CVPR, pp. 5288–5296 (2016)
2. Chen, D., Dolan, W.B.: Collecting highly parallel data for paraphrase evaluation. In: ACL, pp. 190–200 (2011)
3. Wang, X., Wu, J., Chen, J., Li, L., Wang, Y., Wang, W.Y.: VaTeX: a large-scale, high-quality multilingual dataset for video-and-language research. In: ICCV, pp. 4580–4590 (2019)
4. Hendricks, A.L., Wang, O., Shechtman, E., Sivic, J., Darrell, T., Russell, B.: Localizing moments in video with natural language. In: ICCV, pp. 5804–5813 (2017)
5. Rohrbach, A., Rohrbach, M., Tandon, N., Schiele, B.: A dataset for movie description. In: CVPR, pp. 3202–3212 (2015)
6. Zhou, L., Xu, C., Corso, J.: Towards automatic learning of procedures from web instructional videos. AAA **I**, 7590–7598 (2018)
7. Antol, S., et al.: VQA: visual question answering. In: ICCV, pp. 2425–2433 (2015)
8. Lei, J., Yu, L., Bansal, M., Berg, T.L.: TVQA: Localized, compositional video question answering. In: EMNLP, pp. 1369–1379 (2018)
9. Jang, Y., Song, Y., Yu, Y., Kim, Y., Kim, G.: TGIF-QA: toward spatio-temporal reasoning in visual question answering. In: CVPR, 1359–1367 (2017)
10. Lei, J., Li, L., Zhou, L., Gan, Z., Berg, T.L., Bansal, M., Liu, J.: Less is more: ClipBERT for video-and-language learning via sparse sampling. CVPR, pp. 7331–7341 (2021)
11. Yu, Y., Kim, J., Kim, G.: A joint sequence fusion model for video question answering and retrieval. In: ECCV, pp. 487–503 (2018)
12. Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: HowTo100M: learning a text-video embedding by watching hundred million narrated video clips. In: ICCV, pp. 2630–2640 (2019)
13. Zhu, L., Yang, Y.: ActBERT: learning global-local video-text representations. In: CVPR, pp. 8743–8752 (2020)
14. Li, L., Chen, Y.C., Cheng, Y., Gan, Z., Yu, L., Liu, J.: HERO: hierarchical encoder for video+ language omni-representation pre-training. In: EMNLP, pp. 2046–2065 (2020)
15. Feng, Z., Zeng, Z., Guo, C., Li, Z.: Exploiting visual semantic reasoning for video-text retrieval. IJCA **I**, 1005–1011 (2020)
16. Korbar, B., Petroni, F., Girdhar, R., Torresani, L.: Video understanding as machine translation. arXiv preprint arXiv:2006.07203 (2020)
17. Li, Z., Guo, C., Yang, B., Feng, Z., Zhang, H.: A novel convolutional architecture for video-text retrieval. In: ICME, pp. 1–6 (2020)
18. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: a joint video and image encoder for end-to-end retrieval. arXiv preprint arXiv:2104.00650 (2021)
19. Wu, P., He, X., Tang, M., Lv, Y., Liu, J.: HANet: hierarchical alignment networks for video-text retrieval. In: ACM-MM, pp. 3518–3527 (2021)

20. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.B.: Momentum contrast for unsupervised visual representation learning. In: CVPR, pp. 9726–9735 (2020)
21. Dosovitskiy, A., et al.: An image is worth 16 × 16 words: transformers for image recognition at scale. In: ICLR (2021)
22. Vaswani, A., et al.: Attention is all you need. In: NeurIPS, pp. 5998–6008 (2017)
23. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT, pp. 4171–4186 (2019)
24. Gabeur, V., Sun, C., Alahari, K., Schmid, C.: Multi-modal transformer for video retrieval. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12349, pp. 214–229. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58548-8_13
25. Rouditchenko, A., et al.: AVLnet: learning audio-visual language representations from instructional videos. arXiv preprint arXiv:2006.09199 (2020)
26. Amrani, E., Ben-Ari, R., Rotman, D., Bronstein, A.: Noise estimation using density estimation for self-supervised multimodal learning. AAA **I**, 6644–6652 (2021)
27. Luo, H., et al.: UniVL: a unified video and language pre-training model for multimodal understanding and generation. arXiv preprint arXiv:2002.06353 (2020)
28. Yang, J., Bisk, Y., Gao, J.: TACo: token-aware cascade contrastive learning for video-text alignment. arXiv preprint arXiv:2108.09980 (2021)
29. Patrick, M., et al.: Support-set bottlenecks for video-text representation learning. In: ICLR (2021)
30. Liu, S., Fan, H., Qian, S., Chen, Y., Ding, W., Wang, Z.: HiT: hierarchical transformer with momentum contrast for video-text retrieval. arXiv preprint arXiv:2103.15049 (2021)
31. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
32. Wang, X., Zhu, L., Yang, Y.: T2VLAD: global-local sequence alignment for text-video retrieval. In: CVPR, pp. 5079–5088 (2021)
33. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
34. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: CVPR, pp. 3733–3742 (2018)
35. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12356, pp. 776–794. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58621-8_45
36. Khosla, P., et al.: Supervised contrastive learning. In: NeurIPS, pp. 18661–18673 (2020)
37. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E.: A simple framework for contrastive learning of visual representations. In: ICML, pp. 1597–1607 (2020)
38. Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.E.: Big self-supervised models are strong semi-supervised learners. In: NeurIPS, pp. 22243–22255 (2020)
39. Grill, J., et al.: Bootstrap your own latent - a new approach to self-supervised learning. In: NeurIPS, pp. 21271–21284 (2020)
40. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. arXiv preprint arXiv:2104.02057 (2021)
41. Chen, X., He, K.: Exploring simple siamese representation learning. In: CVPR, pp. 15750–15758 (2021)
42. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: a cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: ACL, pp. 2556–2565 (2018)

43. Krishna, R., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. IJCV, pp. 32–73 (2017)
44. Ordonez, V., Kulkarni, G., Berg, T.: Im2Text: describing images using 1 million captioned photographs. In: NeurIPS, pp. 1143–1151 (2011)
45. Chen, X., et al.: Microsoft COCO captions: data collection and evaluation server. arXiv preprint arXiv:1504.00325 (2015)
46. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: collecting region-to-phrase correspondences for richer image-to-sentence models. In: ICCV, pp. 2641–2649 (2015)
47. Liu, Y., Albanie, S., Nagrani, A., Zisserman, A.: Use what you have: video retrieval using representations from collaborative experts. In: BMVC, p. 279 (2019)
48. Chen, S., Zhao, Y., Jin, Q., Wu, Q.: Fine-grained video-text retrieval with hierarchical graph reasoning. In: CVPR, pp. 10635–10644 (2020)
49. Kiros, R., Salakhutdinov, R., Zemel, R.S.: Unifying visual-semantic embeddings with multimodal neural language models. arXiv preprint arXiv:1411.2539 (2014)
50. Faghri, F., Fleet, D.J., Kiros, J.R., Fidler, S.: VSE++: improving visual-semantic embeddings with hard negatives. In: BMVC, p. 12 (2018)
51. Mithun, N.C., Li, J., Metze, F., Roy-Chowdhury, A.K.: Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In: ICMR, pp. 19–27 (2018)
52. Chefer, H., Gur, S., Wolf, L.: Transformer interpretability beyond attention visualization. In: CVPR, pp. 782–791 (2021)