TCVM: Temporal Contrasting Video Montage Framework for Self-supervised Video Representation Learning

Fengrui Tian[†1], Jiawei Fan[†2], Xie Yu[2], Shaoyi Du[1(✉)], Meina Song[2], and Yu Zhao[3]

[1] Xi'an Jiaotong University, Xi'an, China
[2] Beijing University of Posts and Telecommunications, Beijing, China
[3] Harbin Institute of Technology, Harbin, China
tianfr@stu.xjtu.edu.cn, {jwfan, yuxie130, mnsong}@bupt.edu.cn
dushaoyi@gmail.com, zhaoyu_zzz96@163.com

Abstract. Extracting appropriate temporal differences and ignoring irrelevant backgrounds are two important perspectives on preserving sufficient motion information in video representation, such as driver behavior monitoring and driver fatigue detection. In this paper, we propose a unified contrastive learning framework called Temporal Contrasting Video Montage (TCVM) to learn action-specific motion patterns, which can be implemented in a plug-and-play way. On the one hand, Temporal Contrasting (TC) module is designed to guarantee appropriate temporal difference between frames. It utilizes high-level feature space to capture raveled temporal information. On the other hand, Video Montage (VM) module is devised for alleviating the effect from video background. It demonstrates similar temporal motion variances in different positive samples by implicitly mixing up the backgrounds of different videos. Experimental results show that our TCVM reaches promising performances on both large action recognition dataset (i.e. Something-Somethingv2) and small datasets (i.e. UCF101 and HMDB51).

1 Introduction

Video representation learning is a fundamental task in computer vision, which promotes the performance of related downstream tasks, e.g., action recognition[45, 32, 12], video retrieval[47], and temporal action detection[33]. Especially, in auto-piloting[4] or co-piloting[31] areas, video representation learning is significant for driver fatigue detection and driver abnormal behavior detection. Unlike image-related tasks, video representation learning needs neural networks to capture both spatial and temporal features, which makes the problem more

[1] †: Equal Contribution.
[2] This work was done when Fengrui Tian and Jiawei Fan were interns at Megvii Research.

complicated and challenging. Moreover, compared with image labeling, video labeling is relatively cumbersome and time-consuming work. Many researchers have turned to self-supervised learning methods recently.

There are several mainstream directions in self-supervised video representation learning, including pretext task designing and contrastive learning. Some studies focus on designing video-related pretext tasks, such as solving video jigsaw puzzles[1, 24, 28], predicting video clip order[47] and predicting video speed[3, 43, 23]. Besides, contrastive representation learning shows great potential in computer vision tasks. Some researcher turn to design effective contrastive learning frameworks, e.g. MoCo[20], BYOL[16], SimCLR[7]. VideoMoCo[35] and Feichtenhofer *et al.*[13] try to apply these contrastive learning methods to video representation learning.

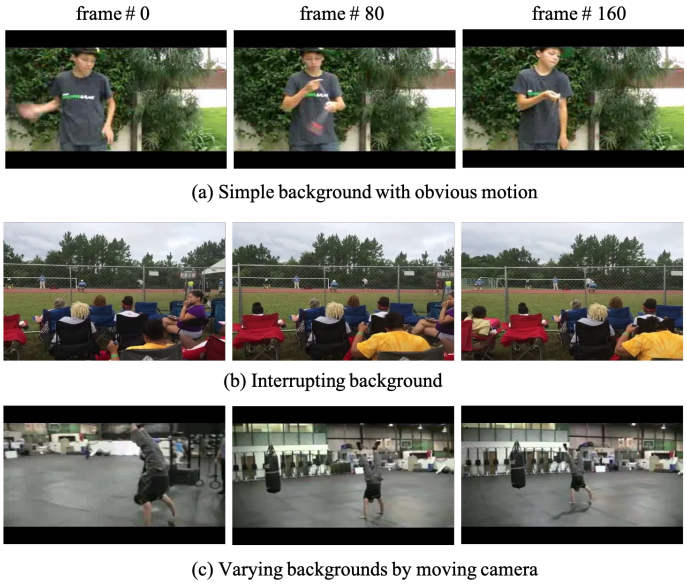

Fig. 1: The ability of extracting motion information in video stream is severely constrained by the complex combinations of views, backgrounds, and motions.

Although previous studies have gained great achievements, there still exist two main problems in video contrastive learning. **i) Extracting motion information.** On the one hand, in a specific video clip, frames with high similarity easily contributes to similar representations. On the other hand, some videos contains interrupting background and camera movements (shown in Fig. 1(b) and Fig. 1(c)), which affects the localization to action subject. The above aspects make it difficult to extract motion variances. **ii) Unavoidable scene-bias.** Actions and backgrounds are mutually relevant. For example, playing football

often happens in playground, and swimming pool obviously implies swimming. The apparently distinguishable backgrounds provide a shortcut for discriminating positive and negative samples, which drastically affects the performance of contrastive learning.

In order to tackle the above problems and capture more action-specific motion patterns in video representation learning, we propose a unified framework called Temporal Contrasting Video Montage (TCVM), which contains two modules Temporal Contrasting (TC) and Video Montage (VM). The TC module utilizes high-level feature differences among neighboring frames to model temporal information, and simultaneously retains necessary spatial features in videos, which does not need to pre-process dataset in advance or to add extra datas in other modalities. The VM module mixes up all background information of different videos within a batch to decrease the differences of background bias between positive and negative samples, which improves the performance of contrastive learning. Overall, the proposed TCVM framework can be efficiently inserted into 2D or 3D CNN models with ease.

The contributions are summarized as follows.

- We propose a plug-and-play framework called Temporal Contrasting Video Montage (TCVM) for video contrastive learning, which can improve the ability of extracting motion information and alleviate negative scene bias.
- We propose the Video Montage module that implicitly mixes up different video backgrounds for erasing the irrelevant background noises. To demonstrate motion features, we present the Temporal Contrasting module that models frame-wise foreground variances in videos.
- Experimental results show that the proposed method outperforms significant margin on Something-Somethingv2 action classification task and achieves a significant improvement on UCF101 and HMDB51.

2 Related works

Video representation learning from pretext tasks. Learning video representations from pretext tasks aims to design a task that generates pseudo video labels. There are some methods that extend successful pretext tasks from image domain to video domain, such as solving video jigsaw puzzle[1, 24, 28], identifying video clips rotation[26] and video colorization[42]. However, these methods cannot capture strong spatio-temporal video representations. By the nature of temporal consistency in videos, researchers design different pretext tasks such as identifying video clip order[14, 41], sorting video frames[30], and predicting video clip order[47]. Besides, many recent studies focus on predicting video speeds [3, 43, 23] or relative playback speed of the same video[6]. However, the representations learned by these methods are similar to the designed tasks and often irrelevant to downstream tasks[34].

Video contrastive learning. Contrastive learning-based methods aim to build representations by maximizing the similarity of the same instance with different

views (positive pairs) and minimizing similarity of different instances (negative pairs). Recently, contrastive representation learning shows great potential in computer vision tasks. MoCo[20] presents a contrastive learning framework that uses a momentum encoder to build dynamic negative pairs. To explore the ability in video representation learning, some methods research on designing basic contrastive learning frameworks. [35] and [13] try to extend successful image contrastive learning methods to videos. CVRL[37] indicates the importance of temporal consistency in different views of videos. TCLR[10] tries to model video representation by local-local and global-local contrastive pairs. Due to the unique temporal features in video domain compared to images, Some researchers start to design contrastive pairs for learning more temporal representations. For example, DPC[17] and Mem-DPC[18] try to learn temporal representations by contrasting the predicted frame features.

Motion Learning in Videos The motion information in video plays an important role in video representation learning. But existing methods often pay insufficient attention to temporal features. Ding *et al.*[11] presents a self-supervised contrastive learning method that merges different foregrounds and backgrounds in videos. Choi *et al.*[9] propose to mask actors with a human detector and further present a novel adversarial loss. On the other hand, many researchers focus on multi-view video contrastive learning methods[2, 40, 36]. For example, contrastive learning with the optical flow is widely researched in recent years and achieves impressive results[19, 46]. Besides, Huang *et al.*[22] presents a video representation learning that naturally uses different types of frames in compressed video streams to decouple the motion and context information. The above-mentioned methods try to learn motions by prepossessing the input videos or adding another modality. Our paper focuses on modeling motion features extracted from the encoders.

3 Method

Given only RGB video clips, the proposed method builds the video representations in a contrastive learing way. Like all contrastive learning methods, our approach has an encoder that maps the input videos to latent representations in a unit sphere, and supervises it by the contrastive loss to identify the representations from the same or different video clips. Unlike other video contrastive learning methods, the proposed method employs modeling the high-level video representations and adopts recognizing similar actions in different videos for learning motion representations.

3.1 Overview of Temporal Contrasting Video Montage Framework

The overview of the proposed method is shown in Fig. 2. First, the Video Montage module distributes clips with the same motion patterns to different videos. In this way, it generates a video with several clips that share similar motion

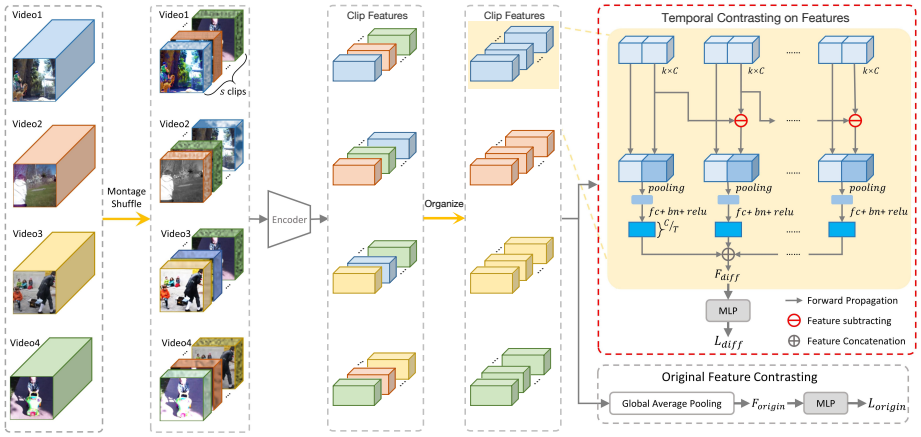

Fig. 2: **The overview of the proposed method.** First, the input videos are re-organized for sharing similar motion clips in different videos by the proposed Video Montage framework. Second, the generated videos are fed into the backbone encoder to extract frame-wise features of each clip. Third, the clip features are introduced into Temporal Contrasting module and Original Feature Contrasting module for video contrastive learning.

representation with other clips in other videos. Second, the generated videos are introduced to the backbone encoder to extract the frame-wise spatio-temporal features, before the representations are re-organized. Third, to model the motion information in videos, the extracted features are introduced into the Temporal Contrasting (TC) module for erasing background bias in each clip and building the temporal representations. The video representations from TC module are supervised by L_{diff}. On the other hand, to ensure that the model learns necessary scene features in videos, Original Feature Contrasting (OFC) module projects the extracted features to a lower dimension and supervises it by the contrastive loss L_{origin}. The goal of contrastive learning is to pull together positive samples and push away negative samples. The loss function is defined as the following equations.

$$L_{origin} = \sum_{j \in P_o(i)} -log \frac{exp(cos(z_i, z_j)/\tau)}{\sum_{k \in A_o} exp(cos(z_i, z_k)/\tau)}, \tag{1}$$

$$L_{diff} = \sum_{j \in P_d(i)} -log \frac{exp(cos(z_i, z_j)/\tau)}{\sum_{k \in A_d(i)} exp(cos(z_i, z_k)/\tau)}, \tag{2}$$

where z_i denotes the i^{th} clip representation and $P_o(i)$ and $P_d(i)$ contain all positive sample representations of ith clip in OFC and TC module, respectively. $A_o(i)$ and $A_d(i)$ include the representations of negative samples of ith clip in OFC and TC module, separately. Cosine similarity is applied to calculate the distance between different samples. Following [27], we use the summation over

positive samples located outside of the *log*. τ indicates a temperature parameter to control the smoothness of the distance. In this module, the final loss L is defined as Equation (3).

$$L = \alpha L_{origin} + \beta L_{diff}, \tag{3}$$

where α and β are all set to 0.5 in this study. The model is trained in an end-to-end manner.

3.2 Video Montage Module

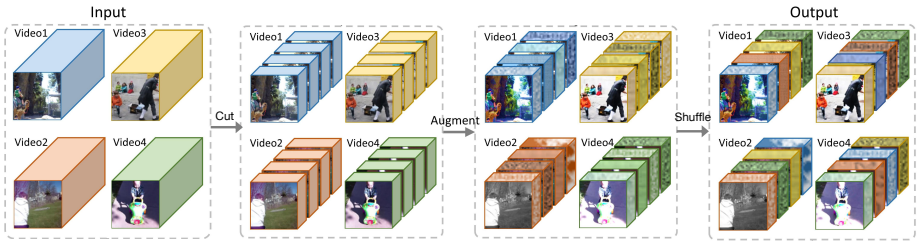

Fig. 3: **Video Montage module.** First, the input video is segmented into s clips uniformly. Next, each clip is added augmentation independently. Last, the clips are shuffle cross all batches.

The Philosophy of Video Montage Video Montage is especially designed for video contrastive learning, which aims to alleviate negative scene-bias. Different from adding static frames, Video Montage mixes up the background information in positive-negative sample sets within a batch, which can decrease the dissimilarity resulting from background. Video Montage mixes up the background information by model itself, rather than introducing additional operation, in which the background information is fused after random shuffling.

Video Montage Operation To imitate recognizing the same action in different videos by human eyes, the VM framework is proposed as shown in Fig. 3. There are three steps in VM framework: cut, augment, and shuffle.

Cut. First, s clips with the same motion are created by segmenting videos uniformly. For example, the input video consists of 16 frames and suppose $s = 2$. The input video is cut into 2 clips uniformly with 8 frames in each clip. In other words, the first clip consists of the front 1-8 frames, the second clip consists of the next 8-16 frames.

Augment. Each clip is augmented independently as shown in Fig. 3. The clips from the same video are considered as positive pairs. Adding independent augmentation could create dissimilar positive clip pairs and hence boosts the network to learn the action representation with less background bias. In this way, the same motion clips with different background information are created as positive pairs.

Shuffle. The same motion clips are distributed in current mini-batch. The new s clips in the same batch are concatenated in the temporal dimension to generate the new videos. The junction of different clips is considered as an augmentation at temporal dimension. It forces the network to separate two adjacent clips by inner temporal similarity.

3.3 Temporal Contrasting Module

Above-mentioned contrastive learning framework could learn strong representations of videos. However, as described in Sec. 1, the network may take a shortcut for discriminating different video clips because scenes in different videos vary greatly. To address this issue, the TC module is proposed. Suppose $X = \{x_1, x_2, ..., x_T\}$ as the feature tensor of a video clip extracted from the encoder. $x_i \in \mathbb{R}^{C\times H\times W}$ is the ith frame feature. T is the number of video frames. In TC module, the feature of ith frame x_i is used to model the differences with its next frame.

$$d_i = \begin{cases} x_i^j - x_{i+1}^j & j \leq k \times C, \\ x_i^j & j > k \times C, \end{cases} \tag{4}$$

where d_i denotes the feature differences at the i^{th} frame. $k \in [0, 1]$ controls the proportion of channels that calculates the current frame feature differences with the next frame. Then, the feature d_i is transformed by one spatial pooling layer and one fully connected layer with the batch normalization and the ReLU function for encoding the i^{th} frame features. The difference information between i and $i + 1$ frames represents the changing information cross temporal dimension. Besides, the original feature represents scene information in videos. The proportion of the channels that model differences in each frame is important in TC, and it is set to $k = 0.5$ in this study. We will discuss the k values in ablation study.

Another challenge is how to fuse the feature difference information for the final video representation. Temporal pooling operation is widely used in recent self-supervised video representation learning methods[14, 35, 1]. This operation may not be appropriate for fusing the feature difference information. First, the feature difference information is well-ordered in temporal dimension. Videos are generated from the first frame to the last frame. The feature difference information between two ordered frames represents the changing information in the video. However, the pooling operation is irrelevant in time. If we flip frame-wise features in temporal dimension, the final video representation ought to be

different. Unfortunately, the representation is the same with non-flipped features by using the temporal pooling. Second, pooling operation would destroy the difference information between two neighboring frames. Since the difference information is calculated by the subtraction of two neighboring frames, the addition among temporal dimension equals to adding the first and the last frame features and omits the information in other frames. To model the difference information at every frame, one fully connected layer f is used to project the i^{th} frame difference information d_i from C dimensions to C/T dimensions. The final difference feature F_{diff} is conducted by the following equation.

$$F_{diff} = concat\{f(d_1), f(d_2), ..., f(d_T)\}. \tag{5}$$

The dimensions of the final difference features equals to C. In this way, the final representation contains the motion difference information among all temporal frames.

The network extracts the features of the generated videos without any down-sampling operation at the temporal dimension. Then the frame-wise features are cut to k segments uniformly representing the feature of s video clips. Clips' features from the same videos are positive pairs. Besides, the features from different videos are negative pairs. Last, the network will learn from the features of each clip by using the TC module and the OFC module.

4 Results

4.1 Implementation Details

Backbone selections To have an apple-to-apple comparison, we follow the common practices[13, 37] and choose the Slow path in SlowFast[12] as the R3D50 backbone. We also use the TSM50[32] model as a strong backbone to test the potential of the proposed method. To use our framework, all the down-sampling operations in temporal dimension are removed.

Self-supervised pre-training We conduct experiments on Kinetics-400[5] (K400) dataset. K400 is a large scale action recognition dataset. It contains about 240k videos in training dataset and 20k validation videos. We use the training videos without any labels for pre-training our models. 32 frames sampled from each video with the temporal stride 2 are segmented into $s = 4$ clips in our experiments. Each clip consists of 8 frames. The image size in each frame is set as 112×112. We follow [8] and use the random grayscale, random color jitter, random gaussian blur, random horizontal flip for augmentation. Temperature parameter τ is set to 0.07 following [20]. The model is trained for 200 epochs. SGD is used as our optimizer with the momentum of 0.9. Batch size is set to 4 per GPU and we use 8 NVIDIA 2080ti GPUs in self-supervised pre-training. The learning rate is set as 0.1 decaying as $0.1\times$ at 50, 100, and 150 epochs.

Method	Year	Pretrain	Network	Input Size	Params	Epoch	Top1	Top5
Modist[46]	2021	K400	R3D50	16×224×224	31.8M×2	600	54.9	
RSPNet[6]	2021	K400	R3D18	16×224×224	33.2M	50	44.0	
RSPNet[6]	2021	K400	S3D-G	16×224×224	11.6M	50	55.0	
MoCo[13]	2021	K400	R3D50	8×224×224	31.8M×2	200	54.4	
BYOL[13]	2021	K400	R3D50	8×224×224	31.8M×2	200	55.8	
SwAV[13]	2021	K400	R3D50	8×224×224	31.8M×2	200	51.7	
SimCLR[13]	2021	K400	R3D50	8×224×224	31.8M×2	200	52	
MoCo$^{\diamondsuit}$[8]	2020	K400	R3D50	8×224×224	31.8M×2	200	54.6	84.3
Random	2022		R3D50	8×224×224	31.8M		54.2	82.9
Supervised	2022	K400	R3D50	8×224×224	31.8M		55.9	84.0
Ours	2022	K400	R3D50	8×224×224	31.8M	200	55.8	84.6
Ours†	2022	K400	R3D50	8×224×224	31.8M	200	**59.3**	**87.2**
Random	2022		TSM50	8×224×224	24.3M		56.7	84.0
Random‡	2022		TSM50	8×224×224	24.3M		57.6	84.8
Supervised	2022	K400	TSM50	8×224×224	24.3M		58.5	85.2
Supervised‡	2022	K400	TSM50	8×224×224	24.3M		60.8	86.5
Ours	2022	K400	TSM50	8×224×224	24.3M	200	58.5	85.2
Ours‡	2022	K400	TSM50	8×224×224	24.3M	200	**60.7**	**86.6**

Table 1: Action recognition results on SSv2. Epoch denotes the pretraining epoch. $\diamondsuit$ denotes the MoCo result from our implementation. † denotes the results from 10×3 view evaluation following the common practice [12, 37, 13]. ‡ denotes the results from 2×3 view evaluation following [32].

Downstream action classification. We found an interesting phenomenon that different activity datasets are related to motion information at different levels[21]. As shown in Table 3b, It remains a relatively high action recognition accuracy on K400 dataset by randomly selecting one frame in each video to train and test the network. In contrast, the action recognition performance decreases dramatically on Something-Somethingv2 (SSv2) with the same training strategy. Fig. 4 could explain these experimental results. Some classes in K400 dataset are more related to their static scene information. In contrast to SSv2 dataset, videos share similar appearances and backgrounds. Temporal information among different frames plays a more important role in the action classification task. Considering this observation, it is more appropriate to use SSv2 rather than K400 for evaluating the video representation ability of the proposed method.

Something-Something v2[15] (SSv2) is a large and challenging video classification dataset. We fine-tune the pre-trained model on SSv2 dataset with a learning rate of 0.005. Following[47], the models are loaded with the weights from the pre-trained model except for TC module and the final fully-connected

Method	Year	Pretraining Dataset	Backbone	Top1	Top5	Top10
Huang[22]	2021	UCF101	C3D	41	41.7	57.4
PacePred[43]	2021	K400	C3D	31.9	49.7	59.2
PRP[48]	2020	UCF101	R3D	22.8	38.5	46.7
Mem-DPC[18]	2020	K400	R3D18	20.2	40.4	52.4
TCLR [10]	2021	K400	R3D18	56.2		
SpeedNet[3]	2020	K400	S3D-G	13.0	28.1	37.5
CoCLR[19]	2020	UCF101	S3D	53.3	69.4	76.6
CSJ[25]	2021	K400	R3D34	21.5	40.5	53.2
MoCo$^{\diamondsuit}$[20]	2020	K400	R3D50	45.0	61.4	70.0
Ours	2022	K400	R3D50	54.2	70.2	78.2
Ours	2022	K400	TSM50	**55.5**	**71.5**	**79.1**

(a) Action retrieval results on UCF101.

Method	Year	Pretraining Dataset	Backbone	Top1	Top5	Top10
Huang[22]	2021	UCF101	C3D	16.8	37.2	50.0
PacePred[43]	2021	K400	C3D	12.5	32.2	45.4
PRP[48]	2020	UCF101	R3D	8.2	25.3	36.2
Mem-DPC[18]	2020	K400	R3D18	7.7	25.7	40.6
BE[44]	2021	UCF101	R3D34	11.9	31.3	
TCLR [10]	2021	K400	R3D18	22.8		
CoCLR[19]	2020	UCF101	S3D	23.2	43.2	53.5
MoCo$^{\diamondsuit}$[20]	2020	K400	R3D50	20.7	41.3	55.5
Ours	2022	K400	R3D50	25.4	47.5	60.1
Ours	2022	K400	TSM50	**25.9**	**49.8**	**64.1**

(b) Action retrieval results on HMDB51.

Table 2: Action retrieval results on UCF101 and HMDB51. $\diamondsuit$ denotes the MoCo result from our implementation.

layer with randomly initialized. The video is sampled to 8 frames with 224 $\times$ 224 resolutions following [12].

Video Retrieval. The proposed method is also tested on UCF101[39] and HMDB51[29] datasets for video retrieval task. Because these datasets are relative small and easy to overfit, results may be quite different without some specific tricks in fine-tuning stage. In contrast, video retrieval task only uses the representations from the encoder, which could reveal the representation ability of the encoder more fairly by using different self-supervised methods. We only report their video re-

Method			Dataset		
TC	VM	OFC	UCF101	HMDB51	SSv2
✓			28.2	14.0	53.9
✓	✓		31.5	13.8	55.0
✓	✓	✓	56.1	25.6	55.8

(a) Ablation study on the proposed TC, VM, OFC.

Dataset	Frames	Top1
K400	1	59.22
K400	8	72.67
SSv2	1	18.42
SSv2	8	60.60

(b) Ablation study on temporal information of K400/SSv2.

k	D^2	UCF101	HMDB51	SSv2
0.25		52.8	23.2	56.4
1		56.4	28.0	55.8
0.5	✓	55.1	26.9	55.7
0.5		**57.3**	**25.9**	**58.6**

(c) Ablation study on TC module.

Operation	UCF101	HMDB51	SSv2
C-A	41.6	17.4	43.3
A-C-S	27.2	12.7	52.3
C-S-A	49.0	19.9	43.4
C-A-S	56.1	25.6	55.8

(d) Ablation study on VM operations.

LR Annealing		Video Sampler		# Frames		Dataset		
CosineLR	StepLR	TSN sampler	I3D sampler	16	32	UCF101	HMDB51	SSv2
✓			✓		✓	55.0	21.2	56.3
	✓	✓			✓	51.6	21.2	56.2
	✓		✓	✓		50.6	19.3	56.0
	✓		✓		✓	56.1	25.6	55.8

(e) Ablation study on the proposed LR annealing, video sampler, #Frames.

Table 3: Ablation studies on TCVM framework. D^2 denotes that we use feature subtraction again on the previous have-subtracted features. C, A and S separately denote *cut, argument* and *shuffle* operation.

trieval results in this paper. After training on K400 dataset, the model is fixed as a feature encoder to test video retrieval tasks.

4.2 Compared with state-of-the-art methods

Table 1 shows the results of the proposed method on SSv2. Since few self-supervised learning methods focus on SSv2, we try our best to investigate all the methods and make comparisons as fair as possible. We also implement MoCo[8] by ourselves. We follow SlowFast[12] and TSM[32]'s evaluation settings to test backbone R3D50 and TSM's accuracy, which is 10×3 views and 2×3 views, respectively. 1-view test results are also provided in Table 1, which means that one video is only sampled into one clip for test. Our method's results are comparative with k400 supervised results in both TSM50 and R3D50 backbone, and the top-5 accuracy even outperforms k400 supervised model. As for TSM, top-1 accuracy can be improved from 57.6% to 60.6% by using the proposed method, which is only 0.1% lower than k400 supervised result. It is noticed that MoCo, BYOL, SwAV and SimCLR[13] uses teacher-student Network architecture. In contrast, the proposed method outperforms MoCo, SwAV, and SimCLR

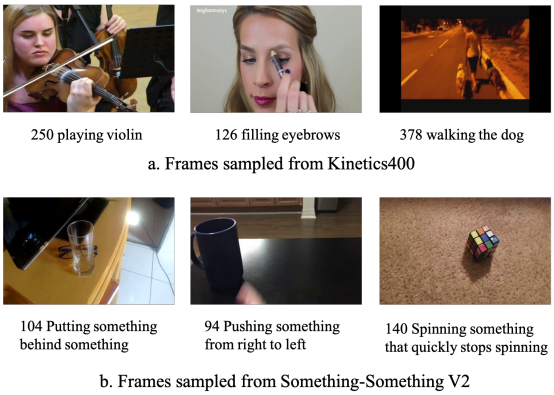

Fig. 4: The visualization of some frames from K400 and SSv2 datasets.

and reaches similar results with BYOL but only uses half of parameters in the pre-training step.

Table 2a and Table 2b present the video retrieval results on UCF101 and HMDB51 datasets. A series of state-of-the-art methods[22, 43, 48, 18, 44, 10, 19, 20, 25, 3] are also listed in the tables. For UCF101 retrieval, our method reaches 54.2% Top1 accuracy, 70.2% Top5 accuracy, and 78.2% Top 10 accuracy, respectively. Furthermore, our method achieves a higher performance by using TSM model. It is noticed that CSJ[25] presents a video jigsaw method to reason about the continuity in videos. Our method makes a great improvement compared with CSJ. Our method outperforms other methods significantly. For HMDB51 retrieval, the proposed method achieves 25.4% Top1, 47.5% Top5, and 60.1% Top10 accuracy by using R3D50 backbone and reaches 25.9% top1, 49.8% Top5, and 64.1% Top10 accuracy by using TSM model.

4.3 Ablation Study

Ablations on the entire framework. To demonstrate the combinations of the proposed TC and VM modules, Table 3a presents the ablation studies on the entire module. Without the proposed VM module, the performance of self-supervised video representation learning drops from 55.0% to 53.9% on the SSv2 action classification task. It also leads to a bad performance on UCF101 and HMDB51 video retrieval tasks. Interestingly, by adding the OFC module, it leaves the apparent performance gap on UCF101 and HMDB51 video retrieval task. But on SSv2 action classification task, the performance only raises from 55.0% to 55.8%. One of the reasons is that fine-tuning on the downstream task may learn non-linear relationship from the pre-trained models. It closes the gap between different pretrained models.

Ablations on the TC module. We conduct ablation study of proposed TC module, as shown in Table 3c. We first test the different proportions of k. $k = 0.25$

denotes that the network uses quarter of channels to model temporal differences among features. $k = 1$ means that all the channels are used for modeling relationships with neighboring frames. Besides, we also try to model high level temporal information. D^2 denotes that the frame subtraction operation performs again on the previous subtraction features. It can be seen that D^2 shows better performance on HMDB51 retrieval task. It means that modeling more complicated motion information is necessary for HMDB51 dataset. Compared with $k = 1$, only using half of channels for model temporal relationship with neighboring channels may bring necessary scene motion information. There is a clear performance gap between $k = 1$ and $k = 0.5$ on UCF101.

Ablations on the VM module. Table 3d presents the performance of two important operations in video montage: augmentation and shuffling. "aug before cut" denotes that augmentations of each clip are added before video shuffling. In this way, clips from the same videos shares the same augmentation. "aug after shuffle" denotes the augmentation is added after the pseudo videos are generated. In this way, clips in the same pseudo videos have the same augmentation. It can be seen that adding augmentation to each videos independently boosts the video understanding performances in each dataset. "wo shuffle" means that the shuffling operation in Fig. 3 is removed. The results show that the performance raises about 15%, 17%, 12% on UCF101 retrieval, HMDB51 retrieval and SSv2 classification tasks respectively by adding the shuffling performance.

Ablations on the training Strategy. To make a fair comparison, some other training strategies[8, 35] are also tested. In Table 3e, the "cosinelr" means we use the cosine learning rate decaying strategies following the common practices[8]. the "TSN sampler" means during the self-supervised training, the input videos are sampled by tsn sampler in [32]. And the "I3D sampler" denotes the input video sampling strategy that is used in the proposed method. The "16 frames" denotes the frame number of input videos in self-supervised training step drops from 32 to 16, which means each clip consists of 4 frames instead of 8.

4.4 Visualization Analysis

To further verify the effectiveness of the proposed method, class activation maps[38] (CAM) results are also presented on UCF101 compared with MoCo, as shown in Fig. 5. (a) shows that our method could localize motion areas more precisely in static background and discriminate action. (b) shows MoCo could easily be distracted by complex surroundings while our model still focuses on actors. (c) implies our model could correctly capture actors' motion clues even when the background is quickly changing.

5 Conclusions

In this paper, we propose the Temporal Contrasting Video Montage framework for self-supervised video representation learning. First, the input videos are pro-

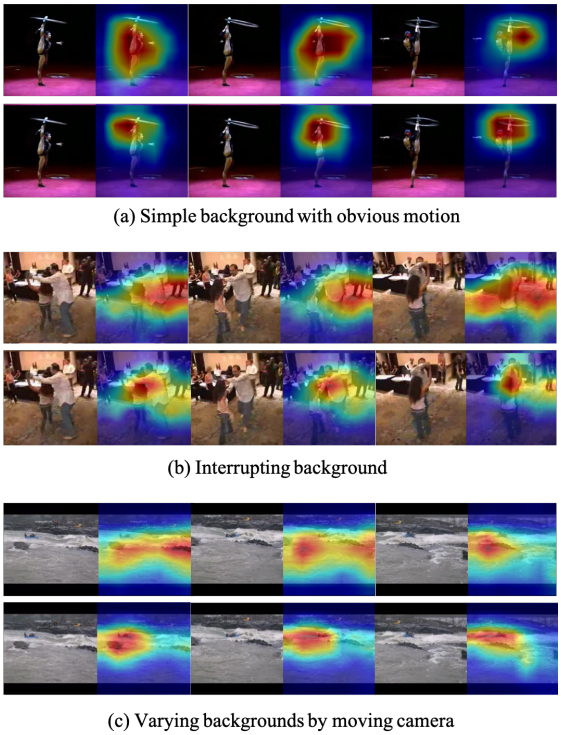

Fig. 5: CAM results on the UCF101 dataset. The first and second raws denote the visualization results of MoCo and the proposed method, respectively.

cessed by the VM module to generate video clips. Second, video clips are introduced to the encoder to extract features. Third, the proposed TC module encodes the high-level features by the frame-wise feature differences. Last, the features from the TC module are optimized in a contrastive learning strategy. Experimental results present that our methods outperforms other state-of-the-art methods for UCF101 and HMDB51 retrieval tasks and reaches the similar accuracy with supervised counterparts on the large scale action recognition dataset SSv2. The proposed method relies on a large scale of unlabelled videos for pretraining, which may limit its range of applications. In the future, we will try to train the proposed method on other datasets.

Acknowledgements This work was supported by the National Key Research and Development Program of China under Grant No. 2020AAA0108100, the National Natural Science Foundation of China under Grant No. 61971343, 62088102 and 62073257, and the Key Research and Development Program of Shaanxi Province of China under Grant No. 2022GY-076.

References

1. Ahsan, U., Madhok, R., Essa, I.: Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 179–189 (2019). https://doi.org/10.1109/WACV.2019.00025
2. Alwassel, H., Mahajan, D., Korbar, B., Torresani, L., Ghanem, B., Tran, D.: Self-supervised learning by cross-modal audio-video clustering. In: NeurIPS (2020)
3. Benaim, S., Ephrat, A., Lang, O., Mosseri, I., Freeman, W.T., Rubinstein, M., Irani, M., Dekel, T.: Speednet: Learning the speediness in videos. In: CVPR. pp. 9922–9931 (2020)
4. Biondi, F.N., Alvarez, I.J., Jeong, K.A.: Human–vehicle cooperation in automated driving: A multidisciplinary review and appraisal. International Journal of Human–Computer Interaction **35**, 932 – 946 (2019)
5. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR. pp. 6299–6308 (2017)
6. Chen, P., Huang, D., He, D., Long, X., Zeng, R., Wen, S., Tan, M., Gan, C.: Rspnet: Relative speed perception for unsupervised video representation learning. In: AAAI. vol. 1 (2021)
7. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML. pp. 1597–1607. PMLR (2020)
8. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
9. Choi, J., Gao, C., Messou, J.C., Huang, J.B.: Why can't i dance in the mall? learning to mitigate scene bias in action recognition. arXiv preprint arXiv:1912.05534 (2019)
10. Dave, I., Gupta, R., Rizve, M.N., Shah, M.: Tclr: Temporal contrastive learning for video representation. arXiv preprint arXiv:2101.07974 (2021)
11. Ding, S., Li, M., Yang, T., Qian, R., Xu, H., Chen, Q., Wang, J.: Motion-aware self-supervised video representation learning via foreground-background merging. arXiv preprint arXiv:2109.15130 (2021)
12. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: ICCV. pp. 6202–6211 (2019)
13. Feichtenhofer, C., Fan, H., Xiong, B., Girshick, R., He, K.: A large-scale study on unsupervised spatiotemporal representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3299–3309 (2021)
14. Fernando, B., Bilen, H., Gavves, E., Gould, S.: Self-supervised video representation learning with odd-one-out networks. In: CVPR. pp. 3636–3645 (2017)
15. Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al.: The" something something" video database for learning and evaluating visual common sense. In: Proceedings of the IEEE international conference on computer vision. pp. 5842–5850 (2017)
16. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., et al.: Bootstrap your own latent: A new approach to self-supervised learning. arXiv preprint arXiv:2006.07733 (2020)
17. Han, T., Xie, W., Zisserman, A.: Video representation learning by dense predictive coding. In: ICCV Workshops. pp. 0–0 (2019)

18. Han, T., Xie, W., Zisserman, A.: Memory-augmented dense predictive coding for video representation learning. In: ECCV. pp. 312–329 (2020)
19. Han, T., Xie, W., Zisserman, A.: Self-supervised co-training for video representation learning. In: NeurIPS (2020)
20. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR. pp. 9729–9738 (2020)
21. Huang, D.A., Ramanathan, V., Mahajan, D., Torresani, L., Paluri, M., Fei-Fei, L., Niebles, J.C.: What makes a video a video: Analyzing temporal information in video understanding models and datasets. In: CVPR. pp. 7366–7375 (2018)
22. Huang, L., Liu, Y., Wang, B., Pan, P., Xu, Y., Jin, R.: Self-supervised video representation learning by context and motion decoupling. In: CVPR. pp. 13886–13895 (2021)
23. Huang, Z., Zhang, S., Jiang, J., Tang, M., Jin, R., Ang, M.H.: Self-supervised motion learning from static images. In: CVPR. pp. 1276–1285 (2021)
24. Huo, Y., Ding, M., Lu, H., Huang, Z., Tang, M., Lu, Z., Xiang, T.: Self-Supervised Video Representation Learning with Constrained Spatiotemporal Jigsaw. In: IJCAI. pp. 751–757 (8 2021). https://doi.org/10.24963/ijcai.2021/104
25. Huo, Y., Ding, M., Lu, H., Huang, Z., Tang, M., Lu, Z., Xiang, T.: Self-supervised video representation learning with constrained spatiotemporal jigsaw. In: IJCAI. pp. 751–757 (8 2021). https://doi.org/10.24963/ijcai.2021/104
26. Jing, L., Yang, X., Liu, J., Tian, Y.: Self-supervised spatiotemporal feature learning via video rotation prediction. arXiv preprint arXiv:1811.11387 (2018)
27. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. In: NeurIPS. vol. 33, pp. 18661–18673 (2020)
28. Kim, D., Cho, D., Kweon, I.S.: Self-supervised video representation learning with space-time cubic puzzles. In: AAAI. vol. 33, pp. 8545–8552 (2019). https://doi.org/10.1609/aaai.v33i01.33018545
29. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: ICCV. pp. 2556–2563. IEEE (2011)
30. Lee, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Unsupervised representation learning by sorting sequences. In: ICCV. pp. 667–676 (2017)
31. Li, Y., Sun, D., Zhao, M., Chen, J., Liu, Z., Cheng, S., Chen, T.: Mpc-based switched driving model for human vehicle co-piloting considering human factors. Transportation Research Part C: Emerging Technologies **115**, 102612 (2020). https://doi.org/https://doi.org/10.1016/j.trc.2020.102612, https://www.sciencedirect.com/science/article/pii/S0968090X18308179
32. Lin, J., Gan, C., Han, S.: Tsm: Temporal shift module for efficient video understanding. In: ICCV. pp. 7083–7093 (2019)
33. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: Bsn: Boundary sensitive network for temporal action proposal generation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–19 (2018)
34. Misra, I., Maaten, L.v.d.: Self-supervised learning of pretext-invariant representations. In: CVPR. pp. 6707–6717 (2020)
35. Pan, T., Song, Y., Yang, T., Jiang, W., Liu, W.: Videomoco: Contrastive video representation learning with temporally adversarial examples. In: CVPR. pp. 11205–11214 (2021)
36. Patrick, M., Asano, Y.M., Huang, B., Misra, I., Metze, F., Henriques, J., Vedaldi, A.: Space-time crop & attend: Improving cross-modal video representation learning. In: ICCV (2021)

37. Qian, R., Meng, T., Gong, B., Yang, M.H., Wang, H., Belongie, S., Cui, Y.: Spatiotemporal contrastive video representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6964–6974 (2021)
38. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: ICCV. pp. 618–626 (2017)
39. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
40. Sun, C., Myers, A., Vondrick, C., Murphy, K., Schmid, C.: Videobert: A joint model for video and language representation learning. In: ICCV. pp. 7464–7473 (2019)
41. Suzuki, T., Itazuri, T., Hara, K., Kataoka, H.: Learning spatiotemporal 3d convolution with video order self-supervision. In: ECCV Workshops. pp. 0–0 (2018)
42. Vondrick, C., Shrivastava, A., Fathi, A., Guadarrama, S., Murphy, K.: Tracking emerges by colorizing videos. In: ECCV. pp. 391–408 (2018)
43. Wang, J., Jiao, J., Liu, Y.H.: Self-supervised video representation learning by pace prediction. In: ECCV. pp. 504–521. Springer (2020)
44. Wang, J., Gao, Y., Li, K., Lin, Y., Ma, A.J., Cheng, H., Peng, P., Huang, F., Ji, R., Sun, X.: Removing the background by adding the background: Towards background robust self-supervised video representation learning. In: CVPR. pp. 11804–11813 (2021)
45. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks for action recognition in videos. IEEE transactions on pattern analysis and machine intelligence **41**(11), 2740–2755 (2018)
46. Xiao, F., Tighe, J., Modolo, D.: Modist: Motion distillation for self-supervised video representation learning. arXiv preprint arXiv:2106.09703 (2021)
47. Xu, D., Xiao, J., Zhao, Z., Shao, J., Xie, D., Zhuang, Y.: Self-supervised spatiotemporal learning via video clip order prediction. In: CVPR. pp. 10334–10343 (2019)
48. Yao, Y., Liu, C., Luo, D., Zhou, Y., Ye, Q.: Video playback rate perception for self-supervised spatio-temporal representation learning. In: CVPR. pp. 6548–6557 (2020)