Motion Sensitive Contrastive Learning for Self-supervised Video Representation

Jingcheng Ni[1,2], Nan Zhou[1,2], Jie Qin[3]*, Qian Wu[4], Junqi Liu[4], Boxun Li[4], and Di Huang[1,2]*

[1] State Key Laboratory of Software Development Environment, Beihang University, Beijing, China
[2] School of Computer Science and Engineering, Beihang University, Beijing, China
[3] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
[4] MEGVII Technology

Abstract. Contrastive learning has shown great potential in video representation learning. However, existing approaches fail to sufficiently exploit short-term motion dynamics, which are crucial to various downstream video understanding tasks. In this paper, we propose Motion Sensitive Contrastive Learning (MSCL) that injects the motion information captured by optical flows into RGB frames to strengthen feature learning. To achieve this, in addition to clip-level global contrastive learning, we develop Local Motion Contrastive Learning (LMCL) with frame-level contrastive objectives across the two modalities. Moreover, we introduce Flow Rotation Augmentation (FRA) to generate extra motion-shuffled negative samples and Motion Differential Sampling (MDS) to accurately screen training samples. Extensive experiments on standard benchmarks validate the effectiveness of the proposed method. With the commonly-used 3D ResNet-18 as the backbone, we achieve the top-1 accuracies of 91.5% on UCF101 and 50.3% on Something-Something v2 for video classification, and a 65.6% Top-1 Recall on UCF101 for video retrieval, notably improving the state of the art.

Keywords: Video Representation Learning, Self-supervised Learning, Local Motion Contrastive Learning, Motion Differential Sampling

1 Introduction

Video understanding has become a necessity in the past decade due to the rapid and massive growth of data. In this challenging task, video representation is the most fundamental and important issue and has received consistently increasing attention. In the literature, many efforts have been made along with the release of several large-scale benchmarks, such as Kinetics [25] and YouTube-8M [1], where representations are learned in a supervised manner from manually annotated samples. Unfortunately, building such databases inevitably incurs enormous human and time cost.

* Corresponding authors.

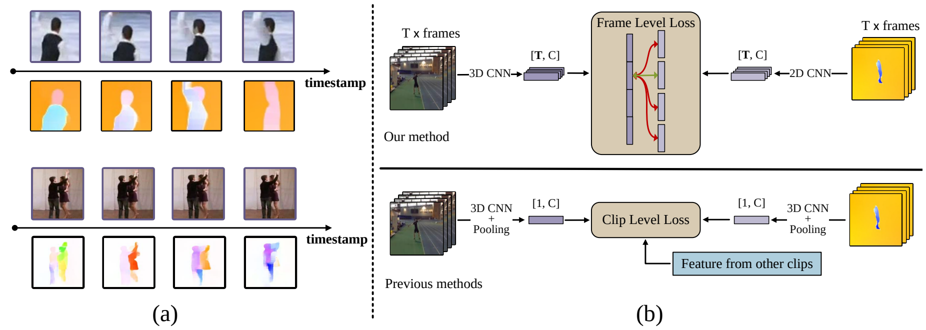


Fig. 1: (a) The RGB frames are not sensitive enough to short-term motion changes, while the optical flows are able to capture subtle motion dynamics between frames, where the changes of motion vectors (in different colors) are clearly observed from the flow maps. (b) Comparison between our method and previous ones using optical flows. Existing works generate clip-level features with temporal pooling, while we focus on more fine-grained frame-level features.

Self-supervised learning has recently emerged as a promising alternative in visual representation. Different from the case on images that only considers spatial variations, that on videos puts more emphasis in temporal characteristics. A number of studies on videos have shown huge potential to learn general features by making use of a tremendous amount of unlabeled data available on the Internet, facilitating diverse downstream applications, including action recognition, action detection, video retrieval, *etc*.

Among current self-supervised video representation learning methods, contrastive learning based ones [12,36] have delivered a great success. They treat the clips from the same video as positive pairs while the ones from different videos as negative pairs and apply the InfoNCE loss [31] to train the model, which is expected to distinguish the clips of a given video from the ones of others. However, clip-level contrastive learning is relatively coarse and primarily benefits global (*a.k.a.* long-term) features [49] without meticulously capturing local (*a.k.a.* short-term) dynamics, thus limiting the performance, in particular in fine-grained scenarios.

More recently, some attempts have compensated this by designing and conducting contrastive learning across video clips with additional views, *e.g.* global *vs.* local [10] and long *vs.* short [4]. Although local temporal modeling is enhanced to some extent with decent improvements reported, they still suffer from two major downsides. On the one hand, for the continuity and redundancy of video data, it is really difficult to handle the discrepancy between the frames within a small time slot, *e.g.* at adjacent timestamps, without high-level supervision, making their representations not sufficiently powerful. On the other hand, existing global-local or long-short contrastive learning requires repetitive temporal interval sampling, leading to multiple forward processes, for a single video, which is both time- and memory-consuming.

In this paper, we propose a novel self-supervised contrastive based approach for video representation learning, namely Motion Sensitive Contrast Learning (MSCL). To overcome the shortcomings aforementioned, besides encoding the global motion from RGB frames, it also fully exploits local temporal clues by introducing optical flows since they prove sensitive to very short-term dynamics, as illustrated in Fig. 1 (a). To fulfill this, we propose Local Motion Contrastive Learning (LMCL) that directly leverages optical flows as the supervisory signal for frame-level local dynamics learning. Specifically, LMCL matches cross-modality (RGB *vs.* optical flows) features at the same timestamp so that subtle motions are modeled. Meanwhile, to restrict the temporal receptive field of flow features in frame-level contrast, different from previous works [37,16,51] on clip-level contrast, we adopt a lightweight 2D CNN as the encoder without temporal message passing, as illustrated in Fig. 1 (b) and elaborated in Section 2.2. In this way, those features capturing local dynamics can be efficiently obtained from the 2D flow encoder, bypassing the cumbersome phase of extra local interval sampling required in [4,10]. In addition, we present two practical strategies to further facilitate LMCL. First, as LMCL introduces frame-level contrast, clips with limited motions tend to bring negative effects to the learning process. To solve this problem, we design Motion Differential Sampling (MDS) to enhance the sampling probability of clips with large motion differential. Second, Flow Rotation Augmentation (FRA) takes rotated flows with different motion vectors as extra negative samples, thereby highlighting motion information on local features.

We summarize our main contributions as follows:

- We propose LMCL, taking advantage of optical flows to underline subtle motions for frame-level contrastive learning, which substantially strengthens self-supervised video representations.
- We present MDS and FRA to optimize temporal interval sampling and optical flow augmentation respectively, both of which further facilitate LMCL.
- We achieve competitive results on several standard benchmarks, *i.e.*, UCF101, HMDB51, and Something-Something v2, in video classification and retrieval.

2 Related Work

2.1 Self-supervised Video Representation Learning

Various self-supervised video representation learning methods have been devised to take advantage of unlabeled video data on the Internet. These methods learn to accomplish various human-designed pre-tasks, including frame sorting [27], pace prediction [45], speed prediction [5,19], and spatio-temporal jigsaw solving [20,2]. More recent works have been inspired by the success of contrastive learning in the image domain such as [17,8,14,6], which can be viewed as instance discrimination tasks [50]. Since videos contain extra attributes that contribute to distinguishing instances, [36,12] take time-shift as the invariant attribute and [18,7] regard speed as the variant attribute. More generally, multiple attributes are explored jointly by the combination of temporal transforms in [22,33]. In [34], transforms are performed in the feature space to reduce memory consumption.

2.2 Optical Flows in Video Understanding

Temporal information is of high importance in video understanding. Optical flows, corresponding to motion vectors across frames, have shown potential in modeling dynamics in the two-stream structure [39]. However, [38,23] indicate that motion information does not work as expected. For example, in supervised action recognition, the shuffling operation at input stage has much more impact on RGB sequences than flow ones. The motivation behind is on the property of appearance-invariance [38], where the motion foreground is described with a low variety. Some self-supervised learning methods can also be seen as taking advantage of this property. COCLR [16] highlights the prior that videos belonging to the same class have similar flow patterns but significant variations in the RGB space, and MoDist [51] distills the flow information to make RGB features focus more on motion foreground. Unlike these methods, we make use of the motion information itself as guidance to improve the ability of modeling local dynamics in RGB features.

2.3 Fine-grained Temporal Features

Learning local dynamics is an important topic for video understanding. Some works attempt to improve modeling through dedicatedly designed structures [47,24] or explicit constraints [48]. But in the self-supervised learning domain, existing methods do not pay enough attention to fine-grained temporal features. For instance, [12,36] can be viewed as learning slow features [49], which are more relevant to scene information. To encode more temporal clues in self-supervised video representation learning, LSFD [4] introduces feature contrast between long-term and grouped short-term features, and TCLR [10] directly compares features with different time-spans. Although these works do make improvements, they require extra sampled clips in the forward process for short views and cannot take advantage of local motion information in flows.

3 Method

In this section, we introduce the proposed self-supervised framework for video representation learning in detail. We begin by introducing the global and local feature extraction pipelines for the two modalities (*i.e.*, RGB and optical flow), respectively. Subsequently, we present the commonly-used global contrastive losses and the proposed Local Motion Contrastive Learning (LMCL), including sample pair construction and augmentation. Finally, we introduce the motion differential sampling policy, which provides meaningful samples for learning more effective local temporal features. An overview of our proposed framework is illustrated in Figure 2.

3.1 Global and Local Feature Extraction

Given a sequence of video frames, we first extract optical flow images from pairs of frames with stride s. We learn both clip-level global and frame-level local

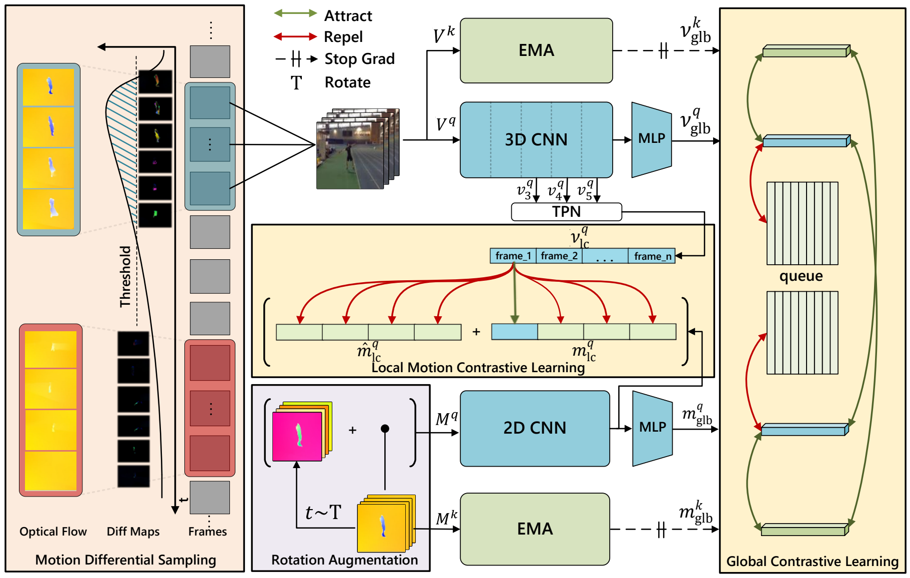


Fig. 2: Overview of the proposed Motion Sensitive Contrast Learning (MSCL) framework. Clips with large motion differentials are sampled from the video and random augmentations are employed for generating queries $\{V^q, M^q\}$ and keys $\{V^k, M^k\}$ w.r.t. the RGB and flow modalities. Then, global features $\{v^q_{\mathrm{glb}}, v^k_{\mathrm{glb}}, m^q_{\mathrm{glb}}, m^k_{\mathrm{glb}}\}$ are extracted for global contrastive learning, where EMA indicates the momentum key encoder in [17]. In order to inject motion dynamics from flow to RGB features, Local Motion Contrastive Learning (LMCL) is conducted based on local features $\{v^q_{\mathrm{lc}}, m^q_{\mathrm{lc}}\}$. To further enforce the model to focus on the motion information, rotation augmentation is applied on the flow inputs and the corresponding local features $\hat{m}^q_{\mathrm{lc}}$ are used to enhance negative samples in LMCL.

features in terms of both RGB and optical flow modalities. Specifically, we adopt the spirit of contrastive learning to build self-supervised features, and randomly sample video clips $\{V^q, V^k\}$ with the corresponding optical flows $\{M^q, M^k\}$ as the queries and keys. As we follow the symmetric structure of MoCo [17] where queries and keys are encoded in a similar way, we only elaborate how to extract features for queries for brevity.

In the RGB pathway, a 3D CNN is employed as the feature encoder, where the output features of different stages (*i.e.*, conv3, conv4 and conv5 layers) are denoted as $\{v^q_3, v^q_4, v^q_5\}$, respectively. We extract video-level global features based on the output of the last stage (*i.e.*, v^q_5). More concretely, we apply spatio-temporal pooling on v^q_5, followed by a 2-layer MLP as the projection head. For frame-level local features, we use a 3-layer Temporal Pyramid Network (TPN) [54] to merge multi-level features. Specifically, we compute the global features

${}^{q}_{\mathrm{glb}}$ and the local features ${}^{q}_{\mathrm{lc}}$ in the RGB pathway as follows:

$$
\begin{aligned}
{}^{q}_{\mathrm{glb}} &= \mathrm{MLP}(\mathrm{STPool}(\ {}^{q}_{5}));\\
{}^{q}_{\mathrm{lc}} &= \mathrm{SPool}(\hat{}^{q}_{3});\ \ \hat{}^{q}_{3}; \hat{}^{q}_{4}; \hat{}^{q}_{5} = \mathrm{TPN}(\ {}^{q}_{3};\ {}^{q}_{4};\ {}^{q}_{5});
\end{aligned}
\tag{1}
$$

where STPool( ) and SPool( ) indicate spatio-temporal pooling and spatial pooling, respectively. In practice, we only use the first-stage output $\hat{}^{q}_{3}$ from TPN, due to two reasons. First, the receptive fields of temporal features in later stages of 3D CNNs (especially those with temporal down-sampling, *e.g.*, S3D [52] and R(2+1)D [43]) are too large to compute the frame-level contrastive loss. Second, TPN makes local features benefit more from multi-level information conveyed in the RGB image.

For the flow pathway, we use a 2D CNN as the feature encoder to extract optical flow features. This design pushes the feature only containing the temporal information in the single flow which benefit LMCL in Section 3.3, and avoid the temporal position leakage [21] by zero padding in 3D CNN. Since the variability of flows is less than that of RGB frames, we decrease the number of channels to 1/8 of that of the RGB counterpart, as suggested in [11,51]. Different from the RGB pathway, there exists no temporal down-sampling (*i.e.*, temporal receptive field is not enlarged), so we directly apply the output of the last stage m^{q}_{5} to extract both global and local features as follows:

$$
\begin{aligned}
m^{q}_{\mathrm{glb}} &= \mathrm{MLP}(\mathrm{STPool}(m^{q}_{5}));\\
m^{q}_{\mathrm{lc}} &= \mathrm{SPool}(m^{q}_{5}):
\end{aligned}
\tag{2}
$$

Similarly, we can obtain the global features of the keys V^{k} and M^{k} w.r.t. RGB and flow as ${}^{k}_{\mathrm{glb}}$ and m^{k}_{glb}, respectively. Note that there is no need to extract local features for the keys as the local contrastive learning is applied on the query features only (see Sec. 3.3).

3.2 Global Contrastive Learning

In this section, we introduce the global contrastive learning based on clip-level features. We follow the basic pipeline of MoCo [17] for both pathways, where the momentum encoder is employed for key inputs, and the memory bank is used for saving negative clips. There are two types of global contrastive losses in our method. The first intra-modality loss is applied on the features from either the RGB or the flow modality to make them discriminative in their own domains. Specifically, we adopt the widely-used InfoNCE [31] loss for global contrastive learning, which is defined as:

$$
\begin{aligned}
L_{\mathrm{RGB}} &= \quad log \frac{h(\ {}^{q}_{\mathrm{glb}};\ {}^{k}_{\mathrm{glb}})}{h(\ {}^{q}_{\mathrm{glb}};\ {}^{k}_{\mathrm{glb}}) + \sum_{i=1}^{N} h(\ {}^{q}_{\mathrm{glb}};\ \bar{}^{k}_{i,\mathrm{glb}})};\\
L_{\mathrm{Flow}} &= \quad log \frac{h(m^{q}_{\mathrm{glb}}; m^{k}_{\mathrm{glb}})}{h(m^{q}_{\mathrm{glb}}; m^{k}_{\mathrm{glb}}) + \sum_{i=1}^{N} h(m^{q}_{\mathrm{glb}}; \bar{m}^{k}_{i,\mathrm{glb}})};
\end{aligned}
\tag{3}
$$

where $h(x;y) = \exp(x^T y{=}kxkkyk\ )$ is the distance between two feature vectors x and y; ${}^{-k}_{i,\mathrm{glb}}$ and $\bar{m}^k_{i,\mathrm{glb}}$ represent the i-th global features of the two modalities in the memory bank with size N, respectively; is the temperature parameter. The second inter-modality loss is applied across the two modalities, with the aim of making the RGB features focus more on motion foreground areas [51], which contribute to the subsequent local contrastive learning. Concretely, the loss term is formulated as follows:

$$\begin{aligned} L_{\mathrm{RF}} = \ & (log \frac{h(\ ^q_{\mathrm{glb}}; m^k_{\mathrm{glb}})}{h(\ ^q_{\mathrm{glb}}; m^k_{\mathrm{glb}}) + \sum_{i=1}^{N} h(\ ^q_{\mathrm{glb}}; \bar{m}^k_{i,\mathrm{glb}})} + \\ & log \frac{h(m^q_{\mathrm{glb}};\ ^k_{\mathrm{glb}})}{h(m^q_{\mathrm{glb}};\ ^k_{\mathrm{glb}}) + \sum_{i=1}^{N} h(m^q_{\mathrm{glb}};\ ^{-k}_{i,\mathrm{glb}})}) ; \end{aligned} \tag{4}$$

and different from [51], the positive and negative keys come from the same modality, which we find in our experiments is more stable in early training. With the intra- and inter-modality losses, we can obtain discriminative global features for both modalities, and at the same time focus on motion foreground areas, which are essential for the subsequent local motion contrastive learning phase.

3.3 Local Motion Contrastive Learning

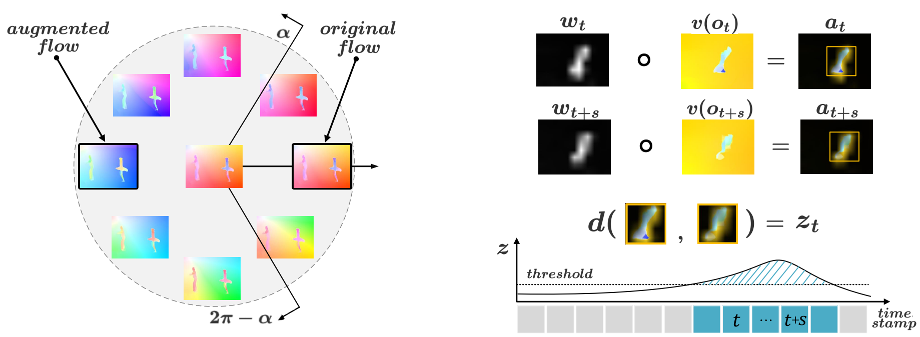
The features learned based on the global contrastive losses have difficulty in modeling local dynamics. To address this, we use optical flows to capture local motion information, and introduce frame-level contrastive losses for learning time-variant features. For the local features ${}^q_{\mathrm{lc}}$ and m^q_{lc}, we only take frame-level features at the same timestamp as positive pairs. Thus, the local motion contrastive loss can be formulated as:

$$L_{\mathrm{LMC}} = \sum_{i=1}^{T} log \frac{h(\ ^q_{\mathrm{lc}}(i); m^q_{\mathrm{lc}}(i))}{h(\ ^q_{\mathrm{lc}}(i); m^q_{\mathrm{lc}}(i)) + \sum_{j=1, j\neq i}^{T} h(\ ^q_{\mathrm{lc}}(i); m^q_{\mathrm{lc}}(j))} ; \tag{5}$$

where ${}^q_{\mathrm{lc}}(i)$ represents the frame-level features at timestamp i, and T is the total number of sampled video frames.

Flow Rotation Augmentation. To make this loss focus more on local motion information, we augment the optical flow with extra negative samples. Specifically, the flow is rotated with a specific angle that is randomly sampled from $[\ ;2\ \ ]$ and the corresponding local features after rotation are treated as negative samples. In this manner, the local motion contrastive loss turns into:

$$\begin{aligned} A(i) &= \sum_{j=1, j\neq i}^{T} h(\ ^q_{\mathrm{lc}}(i); m^q_{\mathrm{lc}}(j)) + \sum_{j=1}^{T} h(\ ^q_{\mathrm{lc}}(i); \hat{m}^q_{\mathrm{lc}}(j)) ; \\ L'_{\mathrm{LMC}} &= \sum_{i=1}^{T} log \frac{h(\ ^q_{\mathrm{lc}}(i); m^q_{\mathrm{lc}}(i))}{h(\ ^q_{\mathrm{lc}}(i); m^q_{\mathrm{lc}}(i)) + A(i)} ; \end{aligned} \tag{6}$$



(a) (b)

Fig. 3: (a) Illustration of Flow Rotation Augmentation (FRA). We randomly sample one angle from $[\alpha, 2\pi - \alpha]$ and rotate the motion vector. (b) Illustration of Motion Differential Sampling (MDS). The notations are consistent with those in Eq. (9). The foreground regions are highlighted for better viewing.

where $\hat{m}^q_{\mathrm{lc}}$ is the local query feature of the augmented flow. As shown in Figure 3 (a), the augmented flow shares the same outline but different motion vectors (reflected in different colors) with the original one. The augmentation strategy improves the ability of distinguishing local flows by motion vectors, which is in line with the goal of learning local dynamics.

The final learning objective is a linear combination of the global and local contrastive losses, simply weighted by a hyperparameter λ:

$$L = L_{\mathrm{RGB}} + L_{\mathrm{Flow}} + L_{\mathrm{RF}} + \lambda L'_{\mathrm{LMC}}. \tag{7}$$

3.4 Motion Differential Sampling

Temporal distinct features are expected to be learned after optimizing the above LMC loss. However, if the motion is similar across different timestamps, it is still difficult for the model to distinguish the corresponding local features. To address this issue, Motion Differential Sampling (MDS) is proposed to enhance the training procedure by choosing samples with larger motion differentials in the foreground. More concretely, suppose $o_t = \{x_t, y_t\}$ and $o_{t+s} = \{x_{t+s}, y_{t+s}\}$ are two adjacent flow maps, we calculate the weighted map w_t that coarsely locates the motion foreground at timestamp t as:

$$w_t = \mathrm{softmax}(\mathrm{up\_down}(\mathrm{Sobel}(o_t))), \tag{8}$$

where Sobel$(\cdot)$ is the Sobel operator [40] for motion boundary detection and up_down$(\cdot)$ is a coarsening operator implemented by downsampling and upsampling with stride r, which is set to 28 in practice. Then, the motion differentials in the foreground area can be defined as:

$$\begin{aligned} a_t &= v(o_t) \quad w_t;\ a_{t+s} = v(o_{t+s}) \quad w_{t+s}; \\ z_t &= \mathrm{sum}(d(a_t; a_{t+s}))\ ; \end{aligned} \tag{9}$$

where $v(\ )$ follows [3] to convert flows into RGB images, $d(\ )$ calculates the Euclidean distance along the channel dimension, s is the stride between frames,   is the Hadamard product, and sum( ) is the summation operation in the spatial domain. More concretely, at timestamp t, we first calculate the pixel-level distance map between two masked flow maps in the RGB space, and then the differential value z_t is the summation of the distance map. As the samples with larger motion differentials in the foreground contribute more to LMCL, we use the differential value as measurement. For one clip, we take the averaged frame-level differential values as the clip-level score. Finally, MDS is performed by choosing those clips, whose differential values are above the threshold, which is simply the median of the values of all candidate clips in the same video.

4 Experiments

4.1 Datasets

The UCF101 [41] and Kinetics400 (K400) [25] datasets are used for pre-training. To evaluate the action classification task, we conduct experiments on UCF101, HMDB51 [26], as well as Something-Something v2 (SSv2) [13]. As for the video retrieval task, we employ the UCF101 and HMDB51 datasets.

4.2 Experimental Settings

Backbones. For the RGB pathway, we follow the common practice in [4,10] and choose ResNet3D-18 (R3D-18) as the general backbone. We also use S3D [52] as the auxiliary backbone for apple-to-apple comparison with more counterparts. For the flow modality, thanks to its simple appearances, we always use ResNet-18 with 1/8 channels as the backbone.

Pre-training Details. The input clip contains 8 frames with the stride of 16 for K400 and 8 for UCF101, as the latter does not have sufficient frames for large strides. We use a 2-layer MLP like [9] for both pathways. For the RGB inputs, we follow the augmentation in [9] including random grayscale, color jitter, Gaussian blur and horizontal flip. For the motion inputs, we extract flows by RAFT [42] and visualize them to 3-channel images. For the consistency in motion information, we copy the flip transforms in RGB inputs to the corresponding motion ones and ignore other augmentations. During training,   is set to   =3 and the rotation angle keeps consistent in one clip. We set the memory size N to 65,536 and the temperature   is 0.07. The model is pre-trained with 200 epochs on the K400 training set and 600 epochs on the UCF101 training set (split-1), with a batch size of 128. The initial learning rate is 0.01 and decreased by the cosine schedule [30]. The optimizer is SGD with a momentum of 0.9 and a weight

Table 1: Ablation study on different designs of MSCL.

Contrastive Losses		LMCL Policies			Finetune		Retrival R@1	
L_{RF}	L_{LMC}	FA	MDS	3D	UCF101	SSV2	UCF101	HMDB51
					65.1	39.1	22.0	13.7
X					71.4	41.0	36.5	22.1
X				X	71.2	41.3	35.6	23.6
	X				70.5	40.3	31.2	19.6
X	X				75.6	42.2	45.7	26.7
X	X	X			76.8	42.5	47.3	27.9
X	X		X		76.4	42.6	46.5	27.2
X	X	X	X		77.3	42.9	48.0	28.1

decay of 1e 4. After pre-training, the RGB backbone is used as initialization parameters for other video tasks. In ablation study, we conduct all experiments on the subset of K400 with 80k videos like [53] and reduce the epochs to 100.

Downstream Tasks. We follow the evaluation protocol in [16], including two types of downstream tasks. (1) Action classification: we add a single layer for classification and then train the full model with both the linear probe and finetune policy. We evaluate the top-1 accuracy. (2) Action retrieval: the backbone is directly used for feature extraction and no further training is required. We take the representations of videos from the test set to query the k-nearest neighbours (k-NNs) in the training set and report Recall at k (R@k) for comparison.

4.3 Ablation Study

Motion Sensitive Contrastive Learning. We analyze how different designs contribute to MSCL, including the contrastive losses L_{RF} and L_{LMC} as well as the MDS and FRA strategies. When only L_{RF} is used, we add the experiment with the 3D backbone in the flow pathway, as there is no need to keep the flow features corresponding to very short-term motions without LMCL. The results are summarized in Table 1. The cross-modality contrastive loss L_{RF} and local motion contrastive loss L_{LMC} are complementary. As shown in [51], L_{RF} forces the model to focus on motion areas, which improves the appearance-invariant property (features are activated on motion areas regardless of appearances). Then, L_{LMC} enhances modeling local dynamics, which exhibits consistent performance gains. The comparison also demonstrates the potential of motion information in optical flows, which is not fully explored in recent self-supervised works. MDS and FRA boost the results independently and their combination leads to further improvement. Their contributions lie in different perspectives: MDS provides better training samples for LMCL and FRA aims at motion-related features.

Feature Extractor. Here, we first study the effect of the 2D backbone in the flow pathway. To achieve this, we exchange 2D ResNet-18 with 3D ResNet-18, whose temporal receptive field is enlarged by pooling and convolution. From the

Table 2: Ablation study on the feature extractor.

Flow Backbone			Finetune		Retrieval R@1	
Arch	T-pad	TPN	UCF101	SSV2	UCF101	HMDB51
R3D-18	zero		71.3	41.3	35.6	23.6
R3D-18	reflect		70.7	40.8	33.9	22.1
ResNet-18	zero		72.3	40.8	38.6	23.9
ResNet-18	zero	X	75.6	42.2	45.7	26.7

results shown in Table 2, we can see that the 3D network delivers the worst performance, due to that zero padding incurs the leakage of the position information as [21] shows, which degrades LMCL. The result of a 3D network without reflect padding is also inferior, which verifies that detailed dynamics in the flow features encoded by 2D CNNs are more crucial to LMCL. We also present the effect of TPN in the RGB pathway. The model without TPN directly uses the conv3 output and adds one more fully-connected layer for consistency in depth. As Table 2 shows, TPN boosts the performance in terms of all the metrics. This can be attributed to the fact that multi-level local features improve the collaboration with global features (from conv5).

Interval Sampling. We study the effect of the proposed MDS by ablating the score function. In Table 3, we observe that when combining both the weight and differential maps, the result is significantly improved, much better than either of the single ones. This phenomenon shows that large motion differentials on foreground areas are beneficial for LMCL.

Flow Rotation. Table 4 shows how different rotation ranges influence video representation learning. From the table, we can see that larger ranges usually lead to better results, which is probably due to that augmented flows increase the difficulty of LMCL, making the features focus more on motion dynamics. It is also noteworthy that the performance decreases a bit when the rotation angle is very small. In this case, the augmented flow is similar to the original one, which confuses the contrastive learning procedure.

How LMCL Works? The model can learn to optimize L_{LMC} from two aspects: the motion vector itself and the deformation of the object. We show that LMCL indeed takes advantage of both factors. To verify this, we remove L_{RF} and conduct two kinds of experiments. First, we employ the original L_{LMC} without augmentation in Eq. (5), and take the motion boundary (extracted by the Sobel operator [40]) as input. In this manner, motion vectors are removed from the flow map. From Table 5, we can see the motion boundary obtains inferior results compared with the flow input, indicating both vector information and deformation are considered in LMCL. Second, we study the superiority of LMCL in learning motion information. To this end, we conduct experiments which directly learn to recognize these transforms. More concretely, we only use the augmented

Table 3: Ablation on sampling methods.

Score Function	Finetune UCF101	SSV2
-	75.6	42.2
sum(w_t)	75.7	42.2
sum($_t$)	76.1	42.3
sum($_t$ w_t)	76.8	42.5

Table 4: Ablation on rotation ranges.

Rotation Range	Finetune UCF101	SSV2
-	75.6	42.2
[=2; 3 =2]	76.0	42.3
[=3; 5 =3]	76.8	42.5
[=6; 11 =6]	76.5	42.4

Table 5: Ablation study on different paradigms for learning motion information.

Flow Input	L_{LMC}	Aug.	Finetune UCF101	SSV2	Retrival R@1 UCF101	HMDB51
flow	Eq.(5)	-	70.5	40.3	31.2	19.6
boundary	Eq.(5)	-	68.2	39.3	30.2	17.7
-	-	-	65.1	39.1	22.0	13.7
flow	Eq.(10)	shift	66.9	39.2	27.3	15.7
flow	Eq.(10)	rotate	66.7	39.4	27.2	14.8
flow	Eq.(10)	shift+rotate	66.9	39.5	27.2	15.7

flows as negative samples, *i.e.*, $A(i)$ in Eq. (6) becomes:

$$A(i) = \sum_{j=1}^{N} h({}^{q}_{\text{lc}}(i); \hat{m}^{q}_{j,\text{lc}}(i)); \tag{10}$$

where $\hat{m}^{q}_{j,\text{lc}}(i)$ indicates the local feature of the j-th augmented flow at timestamp i, and N is the number of augmentations, which is set to 3. Note that, different from Eq. (6), the augmented negative sample at each timestamp is independent. To take the deformation into consideration, we use shift with the same padding as the extra augmentation. Table 5 depicts the results, which, interestingly, show negligible improvement. This, once again, verifies the necessity and effectiveness of the proposed LMCL.

4.4 Comparison to State-of-the-Art Methods

Action Classification. We first evaluate our method on the action classification task. The results including the linear probe and fine-tune policy are shown in Table 6. In terms of the K400 pre-training setting, MSCL outperforms previous methods with the same R3D-18 backbone. For the major counterpart, MoDist [51], MSCL can achieve better top-1 accuracies on both datasets with only half of the epochs. We also notice that the results are still lower than those of the methods like CVRL or MoCo and the reason lies in that they use the more advanced R3D-50 backbone and more training epochs. When pre-training is applied on UCF101 only, MSCL can achieve better results. We also perform

Table 6: Action classification results on UCF101 and HMDB51. 'U+I' denotes the combination of UCF101 and ImageNet. Note that 'Sizes' refer to the test setting.

Method	Network	Year	Dataset	Sizes		Epochs	UCF101	HMDB51
Playback [56]	R18	2020	UCF101	16	112	300	69.0/ -	33.7/ -
CoCLR [16]	S3D	2020	UCF101	32	224	-	81.4/70.2	52.1/39.1
MFO [35]	R18	2021	UCF101	16	112	300	76.2/ -	44.1/ -
TCLR [10]	R18	2021	UCF101	16	112	400	82.4/ -	52.9/ -
SeCo [55]	R50	2021	U+I	-		-	88.2/ -	55.5/ -
MCL [28]	S3D	2021	U+I	16	224	-	90.5/79.8	63.5/ -
Ours	R18	-	UCF101	8	112	400	82.1/72.5	53.7/39.9
Ours	R18	-	UCF101	16	112	400	86.7/77.1	58.9/45.3
TCLR [10]	R18	2021	K400	16	112	100	84.1/ -	53.6/ -
VideoMoCo[32]	R18	2021	K400	32	112	200	74.1/ -	43.6/ -
MFO [35]	R18	2021	K400	16	112	100	79.1/63.2	47.6/33.4
LSFD [4]	R18	2021	K400	16	112	500	77.2/-	53.7/-
ASCNet [18]	R18	2021	K400	16	112	200	80.5/-	52.3/-
MCN [29]	R18	2021	K400	32	128	500	89.7/73.1	59.3/42.9
TE [22]	R18	2021	K400	16	128	200	87.1/-	63.6/-
MoDist [51]	R18	2021	K400	32	112	800	91.3/90.4	62.1/57.5
CVRL [36]	R50	2021	K400	32	256	800	92.2/89.2	66.7/57.3
MoCo [12]	R50	2021	K400	8	256	200	91.0/ -	- / -
Ours	R18	-	K400	16	112	200	90.7/86.1	62.3/55.6
Ours	R18	-	K400	16	112	400	91.5/88.7	62.8/56.5

Table 7: Action classification results on SSv2. † denotes our reproduced result.

Method	Network	Year	Dataset	Size		Epochs	Top-1
RSPNet [7]	R18	2021	K400	16	112	50	44.0
MoDist [51]†	R18	2021	K400	16	112	200	49.1
Ours	R18	-	K400	16	112	200	50.3

evaluation on the SSv2 dataset in Table 7 and reproduce MoDist [51] under the same training setting, where the results show MSCL outperforms others with a 50.3% top-1 accuracy.

Video Retrieval. Similar to the action classification task above, we validate our method with two pre-training datasets and the results are shown in Table 8. On the K400 dataset, our method outperforms the recent state-of-the-art ones with R@1 of 63.7 and 32.6, respectively. On the UCF101 dataset, our method performs better on UCF101 but slightly worse on HMDB51 when compared with the advanced counterpart TE [22]. It is worth noting that MCL [28] applies extra MoCo [17] pre-training on ImageNet, which takes advantage of more training

Table 8: Video retrieval performance on UCF101 and HMDB51. † indicates additional pretraining on ImageNet is applied.

Method	Network	Year	Dataset	UCF101 R@1	UCF101 R@5	UCF101 R@10	HMDB51 R@1	HMDB51 R@5	HMDB51 R@10
MemDPC [15]	R18	2020	UCF101	20.2	40.4	52.4	7.7	25.7	40.6
CoCLR [16]	S3D	2020	UCF101	53.3	69.4	82.0	23.2	43.2	53.5
BE [46]	R18	2021	UCF101	11.9	31.3	44.5	-	-	-
TCLR [10]	R18	2021	UCF101	56.2	72.2	79.0	22.8	45.4	57.8
MFO [35]	R18	2021	UCF101	39.6	57.6	69.2	18.8	39.2	51.0
MCN [29]	R18	2021	UCF101	53.8	70.2	78.3	24.1	46.8	59.7
TE [22]	R18	2021	UCF101	63.6	79.0	84.8	32.2	61.3	71.6
MCL [28]†	S3D	2021	UCF101	67.0	80.8	86.3	26.7	52.5	67.0
Ours	S3D	-	UCF101	63.2	78.7	83.9	25.8	52.1	66.5
Ours	R18	-	UCF101	65.6	80.3	86.1	28.9	56.2	68.3
SpeedNet [5]	S3D	2020	K400	13.0	28.1	37.5	-	-	-
STS [44]	R18	2021	K400	38.3	59.9	68.9	18.0	37.2	50.7
MFO [35]	R18	2021	K400	41.5	60.6	71.2	20.7	40.8	55.2
LSFD [4]	R18	2021	K400	44.9	64.0	73.2	26.7	54.7	66.4
Ours	R18	-	K400	63.7	79.1	84.0	32.6	58.5	70.5

data. These results also demonstrate the effectiveness of MSCL on the retrieval task.

5 Conclusion

In this work, we propose a self-supervised learning framework, namely MSCL, to build motion sensitive video representations. We perform clip-level contrastive learning with intra-modality and inter-modality losses as well as frame-level contrastive learning LMCL to inject motion dynamics from optical flows into RGB frames. Moreover, FRA and MDS are developed to further enhance the contrastive learning procedure by providing better motion-related features and training samples, respectively. Extensive experiments on standard benchmarks show that our MSCL leads to significant performance gains over state-of-the-art methods in terms of two downstream tasks.

Acknowledgment. This work is partly supported by the National Natural Science Foundation of China (62022011), the Research Program of State Key Laboratory of Software Development Environment (SKLSDE-2021ZX-04), and the Fundamental Research Funds for the Central Universities.

References

1. Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijayanarasimhan, S.: Youtube-8m: A large-scale video classification benchmark. arXiv preprint arXiv:1609.08675 (2016)
2. Ahsan, U., Madhok, R., Essa, I.: Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In: WACV. pp. 179–189. IEEE (2019)
3. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. IJCV **92**(1), 1–31 (2011)
4. Behrmann, N., Fayyaz, M., Gall, J., Noroozi, M.: Long short view feature decomposition via contrastive video representation learning. In: ICCV. pp. 9244–9253 (2021)
5. 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)
6. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. NeurIPS **33**, 9912–9924 (2020)
7. 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, p. 5 (2021)
8. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML. pp. 1597–1607. PMLR (2020)
9. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
10. Dave, I., Gupta, R., Rizve, M.N., Shah, M.: Tclr: Temporal contrastive learning for video representation. arXiv preprint arXiv:2101.07974 (2021)
11. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: ICCV. pp. 6202–6211 (2019)
12. Feichtenhofer, C., Fan, H., Xiong, B., Girshick, R., He, K.: A large-scale study on unsupervised spatiotemporal representation learning. In: CVPR. pp. 3299–3309 (2021)
13. 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: ICCV. pp. 5842–5850 (2017)
14. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. NeurIPS **33**, 21271–21284 (2020)
15. Han, T., Xie, W., Zisserman, A.: Memory-augmented dense predictive coding for video representation learning. In: ECCV. pp. 312–329. Springer (2020)
16. Han, T., Xie, W., Zisserman, A.: Self-supervised co-training for video representation learning. NeurIPS **33**, 5679–5690 (2020)
17. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR. pp. 9729–9738 (2020)
18. Huang, D., Wu, W., Hu, W., Liu, X., He, D., Wu, Z., Wu, X., Tan, M., Ding, E.: Ascnet: Self-supervised video representation learning with appearance-speed consistency. In: ICCV. pp. 8096–8105 (2021)

19. 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)
20. 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
21. Islam, M.A., Jia, S., Bruce, N.D.: How much position information do convolutional neural networks encode? arXiv preprint arXiv:2001.08248 (2020)
22. Jenni, S., Jin, H.: Time-equivariant contrastive video representation learning. In: ICCV. pp. 9970–9980 (2021)
23. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.J.: Towards understanding action recognition. In: ICCV. pp. 3192–3199 (2013)
24. Jiang, B., Wang, M., Gan, W., Wu, W., Yan, J.: Stm: Spatiotemporal and motion encoding for action recognition. In: ICCV. pp. 2000–2009 (2019)
25. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
26. 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)
27. Lee, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Unsupervised representation learning by sorting sequences. In: ICCV. pp. 667–676 (2017)
28. Li, R., Zhang, Y., Qiu, Z., Yao, T., Liu, D., Mei, T.: Motion-focused contrastive learning of video representations. In: ICCV. pp. 2105–2114 (2021)
29. Lin, Y., Guo, X., Lu, Y.: Self-supervised video representation learning with meta-contrastive network. In: ICCV. pp. 8239–8249 (2021)
30. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
31. Van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv e-prints pp. arXiv–1807 (2018)
32. 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)
33. Patrick, M., Asano, Y.M., Kuznetsova, P., Fong, R., Henriques, J.F., Zweig, G., Vedaldi, A.: On compositions of transformations in contrastive self-supervised learning. In: ICCV. pp. 9577–9587 (2021)
34. Patrick, M., Huang, P.Y., Misra, I., Metze, F., Vedaldi, A., Asano, Y.M., Henriques, J.F.: Space-time crop & attend: Improving cross-modal video representation learning. In: ICCV. pp. 10560–10572 (2021)
35. Qian, R., Li, Y., Liu, H., See, J., Ding, S., Liu, X., Li, D., Lin, W.: Enhancing self-supervised video representation learning via multi-level feature optimization. In: ICCV. pp. 7990–8001 (2021)
36. Qian, R., Meng, T., Gong, B., Yang, M.H., Wang, H., Belongie, S., Cui, Y.: Spatiotemporal contrastive video representation learning. In: CVPR. pp. 6964–6974 (2021)
37. Recasens, A., Luc, P., Alayrac, J.B., Wang, L., Strub, F., Tallec, C., Malinowski, M., Pătrăucean, V., Altché, F., Valko, M., et al.: Broaden your views for self-supervised video learning. In: ICCV. pp. 1255–1265 (2021)
38. Sevilla-Lara, L., Liao, Y., Güney, F., Jampani, V., Geiger, A., Black, M.J.: On the integration of optical flow and action recognition. In: GCPR. pp. 281–297. Springer (2018)
39. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. NeurIPS **27** (2014)

40. Sobel, I.: History and definition of the sobel operator. Retrieved from the World Wide Web **1505** (2014)
41. 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)
42. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: ECCV. pp. 402–419. Springer (2020)
43. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: CVPR. pp. 6450–6459 (2018)
44. Wang, J., Jiao, J., Bao, L., He, S., Liu, W., Liu, Y.H.: Self-supervised video representation learning by uncovering spatio-temporal statistics. IEEE TPAMI (2021)
45. Wang, J., Jiao, J., Liu, Y.H.: Self-supervised video representation learning by pace prediction. In: ECCV. pp. 504–521. Springer (2020)
46. 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)
47. Wang, L., Tong, Z., Ji, B., Wu, G.: Tdn: Temporal difference networks for efficient action recognition. In: CVPR. pp. 1895–1904 (2021)
48. Weng, J., Luo, D., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Jiang, X., Yuan, J.: Temporal distinct representation learning for action recognition. In: ECCV. pp. 363–378. Springer (2020)
49. Wiskott, L., Sejnowski, T.J.: Slow feature analysis: Unsupervised learning of invariances. Neural computation **14**(4), 715–770 (2002)
50. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: CVPR. pp. 3733–3742 (2018)
51. Xiao, F., Tighe, J., Modolo, D.: Modist: Motion distillation for self-supervised video representation learning. arXiv preprint arXiv:2106.09703 (2021)
52. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning for video understanding. arXiv preprint arXiv:1712.04851 **1**(2), 5 (2017)
53. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: ECCV. pp. 305–321 (2018)
54. Yang, C., Xu, Y., Shi, J., Dai, B., Zhou, B.: Temporal pyramid network for action recognition. In: CVPR. pp. 591–600 (2020)
55. Yao, T., Zhang, Y., Qiu, Z., Pan, Y., Mei, T.: Seco: Exploring sequence supervision for unsupervised representation learning. In: AAAI. vol. 2, p. 7 (2021)
56. 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)