Self-supervised Motion Learning from Static Images

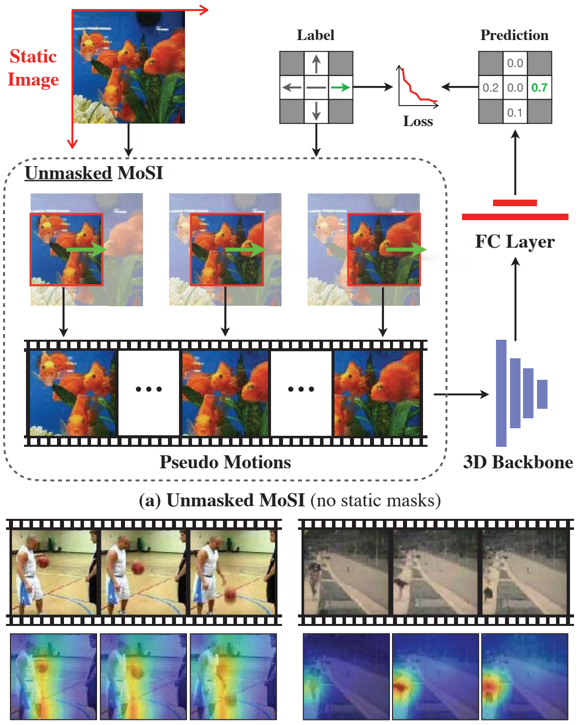
Ziyuan Huang[1,2], Shiwei Zhang[2], Jianwen Jiang[2], Mingqian Tang[2], Rong Jin[2], Marcelo H. Ang Jr[1]
[1] National University of Singapore, Singapore
[2] Alibaba Group, China
ziyuan.huang@u.nus.edu, mpeangh@nus.edu.sg
{zhangjin.zsw, jianwen.jjw, mingqian.tmq,jinrong.jr}@alibaba-inc.com

Abstract

*Motions are reflected in videos as the movement of pixels, and actions are essentially patterns of inconsistent motions between the foreground and the background. To well distinguish the actions, especially those with complicated spatio-temporal interactions, correctly locating the prominent motion areas is of crucial importance. However, most motion information in existing videos are difficult to label and training a model with good motion representations with supervision will thus require a large amount of human labour for annotation. In this paper, we address this problem by self-supervised learning. Specifically, we propose to learn **Mo**tion from **S**tatic **I**mages (MoSI). The model learns to encode motion information by classifying pseudo motions generated by MoSI. We furthermore introduce a static mask in pseudo motions to create local motion patterns, which forces the model to additionally locate notable motion areas for the correct classification. We demonstrate that MoSI can discover regions with large motion even without fine-tuning on the downstream datasets. As a result, the learned motion representations boost the performance of tasks requiring understanding of complex scenes and motions, i.e., action recognition. Extensive experiments show the consistent and transferable improvements achieved by MoSI. Codes will be soon released.*

1. Introduction

Understanding motion patterns is a key challenge in many video understanding problems such as action recognition [7], action localization [40] and action detection [55]. A suitable way to encode motions can significantly boost the performance in those tasks [41]. Early works represent motions using hand-crafted features [35, 46, 34] based on dense trajectories [45] and optical flow [2]. With the successful application of deep neural networks [14, 22, 18] and the construction of large scale image and video datasets [5, 19], endeavors have been made to design architectures to extract meaningful motion features [16, 7, 49, 47, 54, 41]. Despite their powerful capability of modeling dynamic variations between frames, the 3D convolutional models require a large amount of manually labeled videos to achieve a good generalized performance [13].



(a) **Unmasked MoSI** (no static masks)

(b) Dribble

(c) Cartwheel

Figure 1: (a) **Unmasked MoSI** constructs image sequences with pseudo motions from static images. The model is trained to encode motions by predicting the direction and speed of the pseudo motions. For simplicity, the speed granularity here is set as $K = 1$ (see Sec. 3.1). (b), (c) Grad-CAM [38] visualizations on HMDB51 videos for the conv5 pre-trained by our MoSI on ImageNet [5], where the model locates prominent motions even without fine-tuning on the downstream dataset (See more in Sec. 4.1).

Recently, self-supervised learning has emerged as a powerful technique for training the model without labeled data in both image and video paradigm [51, 11, 20, 6, 33]. These methods learn visual representations by exploiting inherent structures of the unlabeled images or videos, for instance, by predicting the correct order of spatial or temporal sequences [51, 6, 9, 24] or by predicting partial contents [11, 12, 33]. Because videos naturally have an extra axis of time compared to images, some methods manipulate the temporal dimension and predict the playback speeds [1, 52]. Although some of the efforts were able to capture the motion information implicitly, almost none of them aims to model motion information of videos explicitly in a self-supervised fashion.

In this work, we seek to train the video model to directly distinguish different motion patterns. The objective is for the model to encode meaningful motion information, so that prominent motions can be discovered and attended to during fine-tuning. Since directly generating predefined motion patterns from a video set may be difficult, we leverage static images for motion generation. Formally, we propose a learning framework that learns motions directly from images (MoSI). Its general structure is shown in Fig. 1. Given the desired direction and the speed of the motions, MoSI generates pseudo motions from static images. By correctly classifying the direction and speed of the movement in the image sequence, models trained with MoSI is able to well encode motion patterns. Furthermore, a static mask (Fig. 3) is applied to the pseudo motion sequences. This produces inconsistent motions between the masked area and the unmasked one, which guides the network to focus on the inconsistent local motions. We term the one with and without static masks as MoSI and unmasked MoSI respectively. Conceptually, the idea of masked MoSI is closely related to attention learning, where the network learns to attend more to the moving areas in videos explicitly created by pseudo motion. Different from the attention mechanism [15, 25, 26], where attention is generated by carefully designed architectures, the attention learned by MoSI is achieved by purely altering the training data.

To the best of our knowledge, this is the first time that static images are used as the data source for pre-training video models. Using MoSI, we are able to exploit large-scale image datasets such as ImageNet [5] to train video models. Although images contain less information about dynamics that are intrinsic in videos, the representations learned with MoSI can be as powerful as those learned using videos in terms of motion understanding. Extensive empirical studies with HMDB51 and UCF101 further demonstrate the effectiveness of MoSI. Compared with other previously published works, we show that the proposed MoSI reaches new state-of-the-art results for learning video representations using RGB modality.

2. Related Work

Motion learning by architectures. Motion information are crucial for understanding videos. There are mainly two popular architectures that are frequently used for extracting video features, respectively two-stream networks [41, 8, 48, 37, 4] and 3D convolutional networks [3, 13, 44, 36, 43]. Two-stream networks extracts motions representations explicitly from optical flows, while 3D structures apply convolutions on the temporal dimension [36, 44] or space-time cubics [3, 43, 13] to extract motion cues implicitly. Besides these two architectures, different motion encodings are proposed to better handle motions in videos [26, 16, 47]. Compared to complicated structure designs that aim at better representing motions, our MoSI can take any video models as the backbone. For simplicity, structures proposed in [44] are adopted for our experiments.

Self-supervised image representation learning. Self-supervised learning is proven to be a powerful tool for learning representations that are useful to down-stream tasks without requiring labeled data. Using image as data sources, there are patch-based approaches [6, 30, 29, 31] that are inspired by natural language processing methods [28], and image-level pretext tasks, such as image inpainting [33], image colorization [53], motion segment prediction [32] and predicting image rotations [10]. The most similar to our work is [32], where labels are generated by grouping pixels that share the same movement together in videos. There are two crucial differences: (a) the aim of [32] is to learn pixels that belong to the same object by motion segmentation, while our MoSI is proposed to learn motion cues for understanding videos; (b) [32] exploits videos to learn image representations, while MoSI takes advantage of images to learn video representations.

Self-supervised video representation learning. With an extra time dimension, videos provides rich static and dynamic information, and there is thus an abundant supply of various supervision signals. A natural way is to extend patch-based context prediction to spatio-temporal scenarios, such as spatio-temporal puzzles [20], video cloze procedure [27] and frame/clip order prediction [24, 51, 9]. Besides the extension of image based supervisions, recent works propose to learn representations by predicting future frames [11, 12]. In addition, supervision signals can be generated by purely manipulating the time axis. Representative works include speed up prediction [1] and play back rate prediction [52]. All previous video representation learning methods exploit videos as the data source. Hence, the motion patterns have not yet been able to be explicitly learned due to the difficulty of generating predefined motion patterns from videos. In this work, we take images as our data source, and generate deterministic motion patterns for directly learning motion representations.

3. Motion Learning from Static Images

The goal of MoSI is to learn motion representations. Because directly generating predefined motions from videos could be difficult, MoSI exploit images to generate samples for motion learning. Specifically, MoSI generates pseudo motions with different speeds and directions. To correctly predict the motion pattern, the 3D video backbone is required to distinguish different motion patterns. In addition, to mimic the actions in actual videos, where there exist inconsistent motions between the foreground and the background, we apply a static mask to the generated pseudo motions. In this way, the network is additionally required to locate prominent motion areas and attend less to the background. In short, there are two core components in the proposed MoSI, respectively the pseudo motions and the static masks, which will be discussed in Sec. 2 and Sec. 3.2 respectively. In the following sections, we refer to the framework as MoSI and unmasked MoSI respectively for the variants with and without static masks.

3.1. Pseudo Motions

The first component is the pseudo motions. The generation process is visualized in Fig. 1. Given the motion label (x, y) sampled from the label pool $\mathbb{L}$, MoSI crops an continuous sequence of images $\mathbf{u} \in \mathbb{R}^{N \times L \times L}$ from the input image (which we term as source image). N and L are selected in accordance to the number of frames and crop size in the downstream task. The generated pseudo motion sequence is then used as the input to the video backbone for motion classification.

Label pool. The motion patterns generated by MoSI consists of two axes, respectively a horizontal axis and a vertical axis. The positive direction for them are respectively toward right and down, as in Fig. 1. For each axis, there are $C = 2 \times K + 1$ speeds, where K denotes the granularity of the speeds in one direction (*e.g.*, the positive direction on the horizontal axis). This corresponds to the motion speed set $\mathbb{S} = \{-K, ..., -1, 0, 1, ...K\}$, where negative values indicate moving in the negative direction of the corresponding axis. K is set to be larger than 1, since we want the network to learn not only the existence but also the magnitude of motions. For simplicity, we decouple the motions for two axis, which means for each label, a non-zero speed only exists on one axis. Therefore, the total size of the label pool is $C_T = 2 \times C - 1 = 4 \times K + 1$ with K labels for each direction and 1 label denoting static sequence. The label pool can be expressed as follows for each label index i:

$$\mathbb{L} = \{i : (x, y) | x \in \mathbb{S}, y \in \mathbb{S}, xy = 0\} \ . \tag{1}$$

Note: It is crucial to generate motions for both axes, because the motion patterns in videos can be both horizontal and vertical. See Sec. 4.1 for the empirical results.

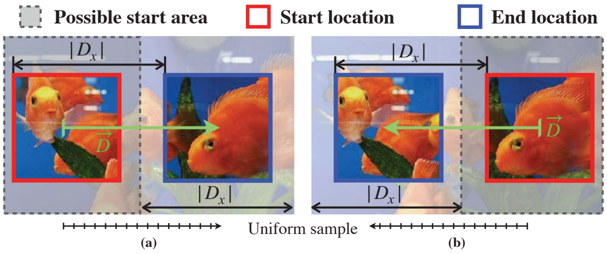


Figure 2: **Exemplar pseudo motion generation processes.** (a) and (b) generates motions respectively in the positive and negative direction of the horizontal axis. The start image is sampled randomly from the *possible start area* and the position of the end image is calculated using the distance D_x and D_y. The *possible start area* is determined so that the end image is located within the source image. N frames are uniformly sampled between the start and the end positions from the source image.

Pseudo motion generation. To generate the samples with different speeds, we define the moving distance from the start to the end of the pseudo motion sequences. For source image with the size of $H \times W$, the distance $\vec{D} = (D_x, D_y)$ for the pseudo motion of speed $(x, y) \in \mathbb{L}$ is defined as:

$$\begin{cases} D_x = \dfrac{(W - L)x}{K}, \text{ if } x \neq 0 \text{ else } D_x = 0 \\ D_y = \dfrac{(H - L)y}{K}, \text{ if } y \neq 0 \text{ else } D_y = 0 \end{cases} . \tag{2}$$

Note that the value of D_x and D_y could be negative, which denotes moving in the negative direction of an axis.

The start location $\overrightarrow{l_{start}} = (x_{start}, y_{start})$ is randomly sampled from a certain area which ensures the end location $\overrightarrow{l_{end}} = \overrightarrow{l_{start}} + \vec{D}$ is located completely within the source image, as demonstrated in Fig. 2. For example, if $D_x > 0$, the distance between the right border of both the start image and the source image should be at least $|D_x|$. For label $(x, y) = (0, 0)$, where the sampled image sequence is static on both axis, the start location is selected from the whole image with uniform distribution. N images are then sampled with uniform gaps from the source image between the start position $\overrightarrow{l_{start}}$ and the end position $\overrightarrow{l_{end}}$.

Classification. The generated image sequence $\mathbf{u}$ is then fed into a 3D backbone network and a linear classifier. Following [1, 10], we employ the same-batch training technique, where each batch contains all transformed image sequences generated from the same source image. This means for each source image, C_T image sequences of pseudo motions are generated and included in the same mini-batch. This is found to be significantly effective for reducing the artificial cues. The model is trained by cross entropy loss.

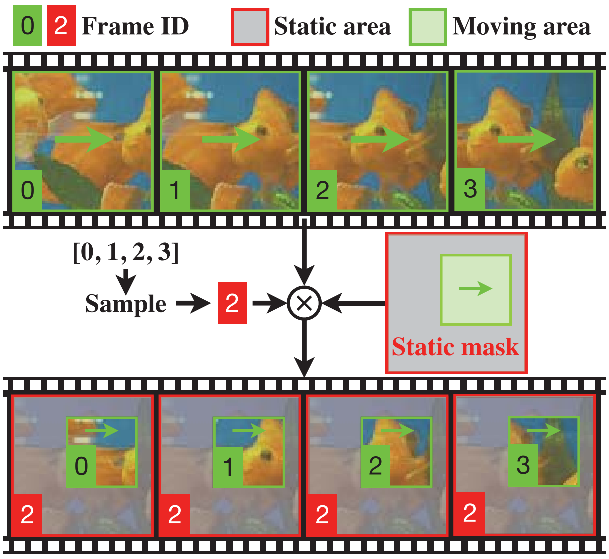


Figure 3: **Exemplar static mask applied on the sampled image sequence.** One of the images in the generated pseudo motions is selected to replace the contents of each image in the static area, while the contents in the moving area are not altered. In this case, the third image (ID=2) in the sequence is selected as the background. The green arrows indicate the moving direction of the bounding boxes in the source image. Essentially, the static mask is a filter that only allows contents in a certain area to be kept the same as the input. The unmasked variant (upper sequence) represents global motion. The static masks creates local motion patterns that are inconsistent with the background.

3.2. Static Masks

By correctly classifying pseudo motions with different directions and magnitudes, the model is able to recognize different motion patterns. However, since for most videos, actions occur in a constrained area rather than all the spatial locations, one is expected to recognize not only global motion patterns, but also inconsistent motions between the foreground and the background. Another drawback for the model to understand global motion is that the model will possibly focus on just several pixels, as all the motion patterns (speed and direction) in the image sequences are essentially the same. This creates an obvious artificial cue [6, 11, 50] that hinders the true capability of the model to understand motions. To this end, we introduce static masks as the second core component of the proposed MoSI.

Static masks divide the spatial location into two groups, respectively masked area and unmasked area, as in Fig. 3. The masked area is regarded as the background and the motions within this area is thus removed, by setting the content of this area in all images in $\mathbf{u}$ to the q-th image $\mathbf{u}_q$. On the other hand, the original contents (*i.e.*, the motions) are retained in the unmasked area of the image sequence $\mathbf{u}$. For simplicity, the unmasked area is by default a square area within the image, with the size of $L_m \times L_m$. Formally, given the masked pixels $\mathbf{m}$, the content of the p-th image is determined by:

$$\tilde{\mathbf{u}}_p = M(\mathbf{u}_p, \mathbf{m}) = \begin{cases} \mathbf{u}_q, & \text{if } (a,b) \in \mathbf{m} \\ \mathbf{u}_p, & \text{if } (a,b) \notin \mathbf{m} \end{cases}, \tag{3}$$

where (a, b) is the spatial location of a certain pixel, and q is the randomly selected static image.

By applying the static mask, the background area of the image sequence becomes static and the foreground is moving according to the label. To perform correct classification, the model is now required not only to recognize motion patterns, but also to spot where the motion is happening. This benefits a lot for downstream tasks such as action recognition, as the model is equipped with knowledge on where to focus even before fine-tuning on the downstream datasets.

3.3. Instantiation

Data preparations. One advantage of the proposed MoSI is that it can train video models on both video datasets and image datasets. This allows for exploiting a large amount of existing image-based datasets. For video and image datasets, the only difference is that the source images need to be first sampled from the videos in the video datasets, while for image datasets, no frame-sampling step is required. Specifically, for video datasets, one frame out of each video is randomly sampled as the source image. Using the same-batch training technique, each image generates C_T samples with different labels. We alter the sampled source frame index for different epochs for a larger variety of visual contents. After obtaining the source images, we resize the source image so that the length of the short side is L_s. An $L_s \times L_s$ square area is randomly cropped from the resized image, which means $H = W = L_s$ in Eq. 2. This ensures that the motion magnitude for the same speed on both axis are the same.

Augmentations. It is shown in previous self-supervised approaches that the model tend to learn some artificial cues or trivial solutions [6, 11, 50] that disrupts the learning of the designed objectives. In our case, we have introduced the static mask to avoid one possible trivial solution, where the model only need to recognize motions in an extremely small area for the prediction of the correct motion class. Based on that, we further randomize the location and the size of the unmasked area. In addition, we randomize the selection of the background frames in the MoSI. In Sec. 4.1, we closely investigate the effect of mask sizes and demonstrate the benefit of our randomization.

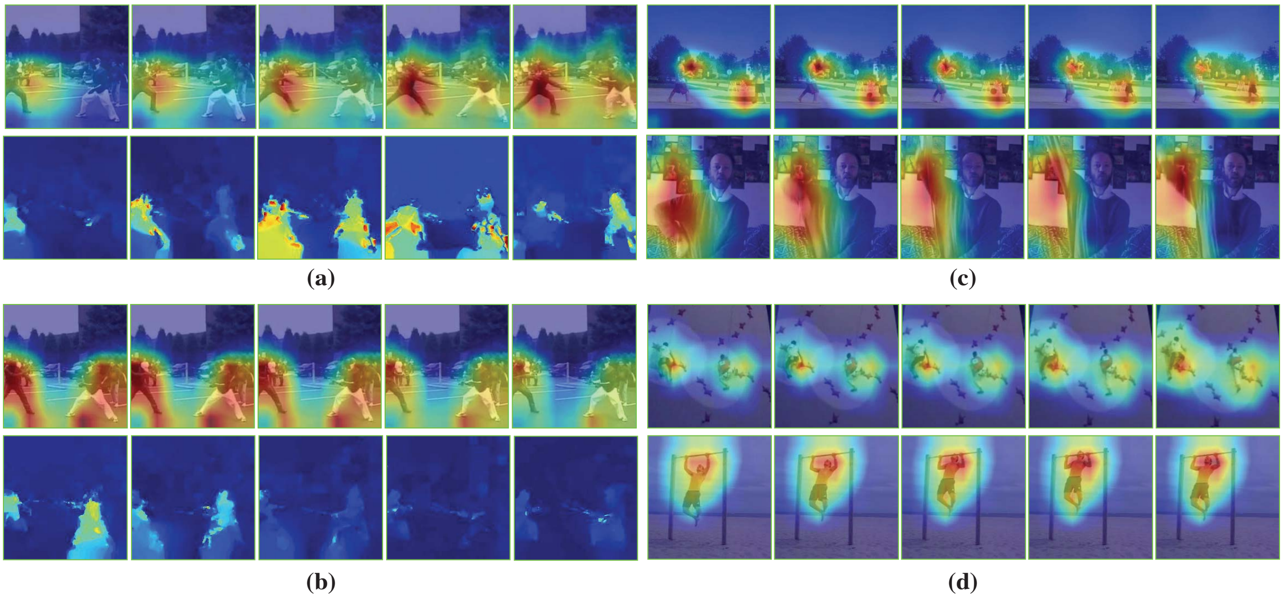


Figure 4: **Grad-CAM visualization** [38] of the conv-5 layer on HMDB51 over the models trained by MoSI on ImageNet ((a), (c)) and HMDB51 ((b),(d)) **without fine-tuning**. Red and blue areas denotes respectively highly and less activated areas. Although no semantic meaning has been taught through MoSI, models trained by MoSI already possess the ability to locate prominent areas according to the motions. In (a) and (b), we additionally compare with the value of optical flow calculated with $\sqrt{x^2+y^2}$. It can be observed that most highly activated areas corresponds to regions where motions are significant.

4. Experiments

Datasets and backbone. For pre-training with MoSI, we employ three video datasets: UCF101 [42], HMDB51 [23], Kinetics [19], as well as the image dataset ImageNet [5]. For evaluation of the learned representation, we use UCF101 and HMDB51. We use R(2+1)D [44] with 10 layers as well as R-2D3D with 18 layers as our backbone, following the configurations in [51, 11, 12, 52].

Self-supervised pre-training. For self-supervised pre-training, we set $L_s = 320$ and resize the source image to 320×320 by default. Image sequences of length 16 and size 112×112 with pseudo motions are generated from each source image and fed to the model. The number of speed on each axis is set to 5, which is the minimal number for each direction to have distinct speeds. The total size of our label pool is thus 9. The side length of the unmasked area in our static mask L_m is randomly sampled from $[0.3, 0.5] \times 112$.

Supervised action classification. During supervised training on UCF101 and HMDB51 for action classification, we train the network with a batch size of 128 samples per GPU for 8 GPUs using Adam with a base learning rate of 0.002 for 300 epochs. For evaluation, we follow the standard protocal [51, 44] using 10 clips to produce the final results and report the results on split 1 on both UCF101 and HMDB51. Further details on both self-supervised and supervised training can be referred to the supplemental material.

4.1. Understanding MoSI.

In this section, we investigate the models trained by MoSI. We use R-2D3D with 18 layers in this section with the same structure as in [11]. The datasets used for pre-training and fine-tuning are the same unless otherwise specified. For each ablation experiment, only the inspected factor is altered and the rest of the settings are kept according to the ones described before.

What has the network learned? We first establish some intuitive understanding of the method, by addressing the question of *what has the model learned through MoSI*. The Grad-CAM [38] visualization of the last layer in the pre-trained model is shown in Fig. 4. Note that no fine-tuning is performed at this stage. As can be seen, the model has learned to pick up salient motion regions in the videos. Especially compared to optical flow, the model trained by MoSI highlights the region where the values of the optical flow is large. Furthermore, despite the model is only given pseudo motions as training data, it is able to transfer the knowledge onto real videos with more complex spatio-temporal relations to discover locate areas with a large motion across different frames. In addition, the prior square motion area does not constrain the model to only look for square areas with motions. Surprisingly, given only prior knowledge of one possible moving region, the model learns to generalize to find multiple prominent motion areas.

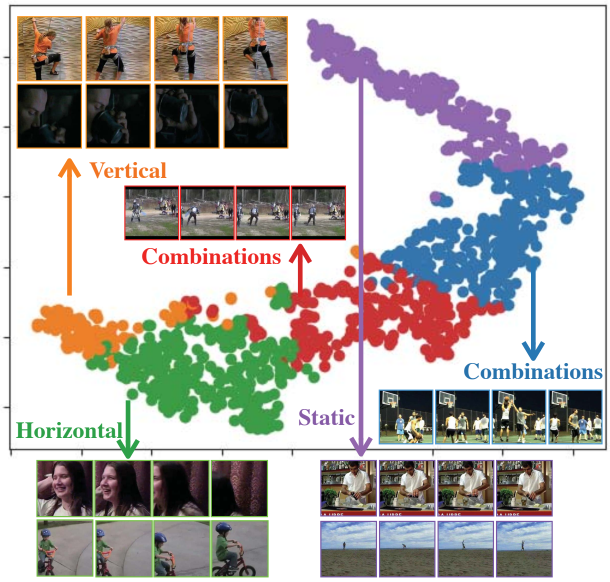


Figure 5: **T-SNE visualization of clustering results on HMDB51.** The representations are generated by unmasked MoSI for easy analysis, and without fine-tuning to directly evaluate the learned representation.

Dataset	Initialization	Label-X	Label-Y	Top1-Acc
UCF101	From scratch	-	-	64.5
	MoSI	✓	✓	**71.8** (+7.3)
		✓	×	71.6 (+7.1)
		×	✓	69.9 (+5.4)
HMDB51	From scratch	-	-	30.4
	MoSI	✓	✓	**47.0** (+16.6)
		✓	×	44.9 (+14.5)
		×	✓	43.1 (+12.7)

Table 1: **Baseline comparison** of models trained using MoSI. Models pretrained with MoSI achieves a notable improvement over the baselines that are trained from scratch.

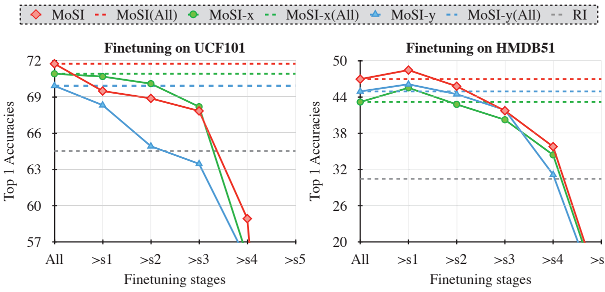


Figure 6: **Performance of the model trained by MoSI when the representation is frozen to a certain extent.** RI indicates training from random initialization, which is the baseline. The trend indicates the usefulness of the low-level feature learnt by MoSI. MoSI-x and -y indicates MoSI with label pool of respectively only horizontal and vertical labels.

We further cluster the learned representation on HMDB51, as in Fig. 5. Although video data are naturally heterogeneous and consist of complex combinations of motions, we can still observe that the movements of a large portions of pixels in three of the five clusters are easily observable, which are vertical, horizontal and static. The motions in the other two clusters are hard to be uniformly described as the multiple movements are present at the same time.

Baseline comparison. We then fine-tune the pre-trained model on the action classification task. As in Table 1, models trained with MoSI achieve notable improvements on both datasets. The improvement on HMDB51 reaches 16.5% when using both axes. The performances with only one axis are weaker compared to two axes, but they still outperform the baseline by a notable margin. On UCF101, the improvement of MoSI reaches 7.3%. However, the benefit of using two axes is smaller. This is partially because the motion cue can be of less importance in classifying UCF101 videos, where even using only one image could achieve satisfactory classification performance, as shown in [39].

Which parts of the representations learned using MoSI are the most useful? We further fine-tune the pre-trained weights with different stages of the learned representation frozen, as in Fig. 6. By gradually freezing the representations during fine-tuning, we observe only a small drop in the performance for the first three stages. On HMDB51, fixing one stage even improves the accuracy. This indicate that the models can learn a strong low-level representation from MoSI. Because MoSI focus less on the semantic understanding of the videos, only fine-tuning the linear layer does not have a high accuracy, which shows that the learned high-level representations are less discriminative. It is natural since the main objective of MoSI is for the network to attend to motions during fine-tuning. The only information that the model receives is different motion patterns, while to discriminate between actions, not only the motion pattern, but also the identity of the moving object need to be recognized. Nevertheless, fine-tuning the last stage on HMDB still gives an improvement of $\sim 5\%$ over its random initialized baseline. Comparing between MoSI with different label pools, we also observe a pattern similar to Table 1: On HMDB51, the models trained using only one axis consistently underperform the two-axes MoSI, while on UCF101, there is not a clear benefit of using two axes.

4.2. Ablation Studies

Effects of mask sizes on MoSI. We then investigate the effects of different mask sizes by altering the side length of the unmasked area L_m. The results are visualized in Fig. 7. In terms of the pre-training accuracy, we can see that the MoSI task is generally easy on UCF101, with the pre-training accuracy being 73% when L_m/L is 0.1 and reach-

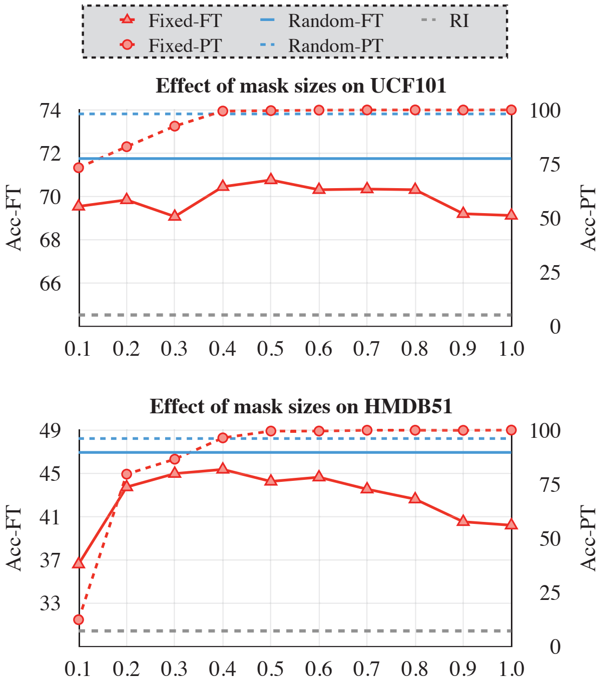


Figure 7: **Effect of mask sizes for MoSI.** The x-axis denotes the side length ratio L_m/L of the unmasked area (1.0 as no static masks). Static mask is useful especially when the unmasked area covers a relatively small region. Random mask sizes further improve the recognition performance.

ing 100% for 0.4. This is probably because of the numerous similar visual contents in UCF101 that cause the network to memorize the patterns. On the other hand, visual contents in HMDB51 have a larger diversity, thus the pre-training accuracy is lower compared to pre-training on UCF101 with the same parameters. For mask size ratio 0.1, the model can hardly learn to discriminate between different motions when the mask size $L_m = 0.1_s$. This demonstrates that the proposed MoSI is not a trivial task that can be learned easily. Comparing different mask sizes, the validation accuracy during pre-training generally improves with the increase of the unmasked area. However, the high performance in our pretext task does not always mean a higher accuracy on the downstream task, which is also observed in [21]. Therefore, a suitable mask size is crucial to ensure a high quality of the representations. We then randomize the unmasked area within the range of $L_m = [0.3, 0.5] \times L_s$ and observe an improvement upon the variants with fixed mask sizes.

Effects of the speed granularity and the number of frames on MoSI. As in Table 2a and 2b, compared to the default setting, reducing the number of class to 3 hurt the performance in that the model is not able to distinguish different motion patterns, which shows the importance of the speed granularity. On the other hand, further increasing the granularity on top of 5 does not have a visible improvement as well. This means it is sufficient for the model to possess the basic ability to distinguish different speeds. For different frames, we fix the frame-wise distances for each label and alter the sizes of source images so that the only factor changed is the number of frames. Table 2c and 2d show that using 16 frames to pre-train MoSI achieves the best performance. One possible reason is that the downstream task also uses 16 frames for fine-tuning.

# Class	Acc-PT	Acc-FT
3	79.7	43.0
5	96.1	**47.0**
7	$\underline{96.3}$	$\underline{44.6}$
9	**96.6**	44.5

(a) **# Classes on HMDB51.**

# Class	Acc-PT	Acc-FT
3	81.4	70.1
5	98.2	71.8
7	$\underline{99.1}$	70.9
9	**99.4**	71.3

(b) **# Classes on UCF101.**

# Frames	Acc-PT	Acc-FT
8	93.7	$\underline{45.6}$
12	$\underline{95.0}$	44.9
16	**96.1**	**47.0**
24	94.5	44.6
32	91.7	45.5

(c) **# Frames on HMDB51.**

# Frames	Acc-PT	Acc-FT
8	98.1	71.7
12	98.3	71.0
16	98.2	**71.8**
24	98.1	71.1
32	**99.6**	70.0

(d) **# Frames on UCF101.**

Table 2: **Ablation studies** on action recognition task with models trained by MoSI. Acc-PT and -FT denotes pre-training and fine-tuning top-1 accuracies respectively. For # Classes, all parameters are kept the same except for the number of classes. For # Frames, only the source image size changes with the number of frames to keep the motion magnitude unchanged. Bold and underlined numbers denotes the best and the second-best performance.

# Samples	3(4%)	5(7%)	7(10%)	9(13%)	11(16%)	13(19%)	Full
BASELINE	4.5	6.7	8.1	10.4	11.7	14.8	30.4
MoSI	**8.1**	**12.5**	**17.4**	**22.2**	**24.1**	**25.4**	**46.9**
Diff	+3.6	+5.8	+9.3	+11.8	+12.4	+10.6	+16.5

Table 3: **Low-shot fine-tuning on HMDB51.** Top-1 accuracy is used for comparison with the baseline (trained from random initialization).

Few/low shot fine-tuning. We also evaluate MoSI under a few/low-shot setting, where we randomly sample 3, 5, 7, 9, 11 and 13 videos from each class of the split 1 training set of HMDB51 as the training set for fine-tuning. This corresponds to using only around 4% to 20% of the original dataset. The results are demonstrated in Table 3. Pre-trained using MoSI, the model is consistently better than the random initialized counterpart by a large margin. This shows the effectiveness of MoSI on few-shot video classification.

Training on ImageNet. Since MoSI is able to train video models from static images, we additionally use ImageNet as the data source for pre-training. We only use a small portion of the original ImageNet data because the motion

Dataset-PT	Acc-PT	Dataset-FT	Acc-FT
UCF101	98.2	UCF101	71.8
ImageNet-S5	83.1		71.1
ImageNet-S10	87.8		70.5
ImageNet-S20	96.9		71.2
ImageNet-S30	97.7		**71.9**
HMDB51	96.1	HMDB51	47.0
ImageNet-S5	83.1		47.3
ImageNet-S10	87.8		47.8
ImageNet-S20	96.9		**48.0**
ImageNet-S30	97.8		47.6

Table 4: **Pre-training models using MoSI on ImageNet.** S5, 10, 20, 30 denote randomly sample 5, 10, 20, 30 from each class respectively. Training video models on ImageNet with MoSI further increases the recognition accuracy on downstream datasets.

	Initialization		Supervised fine-tuning	
Method	Arch.	Dataset	UCF101	HMDB51
OPN [24]	VGG	UCF	59.6	23.8
DPC [11]	R-2D3D	K400	75.7	35.7
MemDPC [12]	R-2D3D	K400	78.1	41.2
3D-RotNet [17]	R3D	K400	62.9	33.7
ST-Puzzle [20]	R3D	K400	65.8	33.7
VCP [27]	C3D	UCF/HMDB	68.5	32.5
VCOP [51]	R(2+1)D	UCF	72.4	30.9
PRP [52]	R(2+1)D	K400	72.1	35.0
SpeedNet [1]	S3D-G	K400	81.1	48.8
MoSI (Ours)	R-2D3D	UCF/HMDB	71.8	47.0
MoSI (Ours)	R-2D3D	K400	70.7	48.6
MoSI (Ours)	R(2+1)D	UCF/HMDB	**82.8**	**51.8**

Table 5: **State-of-the-art comparisons.**

patterns can already be well learned on HMDB51 using only 5k videos. As in Table 4, we further validate that the performance in the pretext task largely depends on the number of data. Increasing the training data results in a higher validation accuracy before it saturates. In terms of the downstream task, the fine-tuning performance generally increases when the number of pre-training data increases before it saturates. Overall, the models pre-trained on ImageNet outperforms the ones trained on respective datasets.

4.3. Comparison with video-based methods

In Table 5, we demonstrate the performance comparison with the state-of-the-art video self-supervised training methods that only use RGB modality. Overall, MoSI performs competitively against other methods on both UCF101 and HMDB51. Compared to DPC [11] and MemDPC [12] with the same architecture MoSI achieves a much stronger performance on HMDB51. Note that DPC and MemDPC uses a 34-layer R-2D3D model with 224 × 224 as input, while MoSI uses an 18-layer R-2D3D with 112 × 112 as input. Using a stronger backbone R(2+1)D, we achieve the state-of-the-art performance on both UCF101 and HMDB51 datasets.

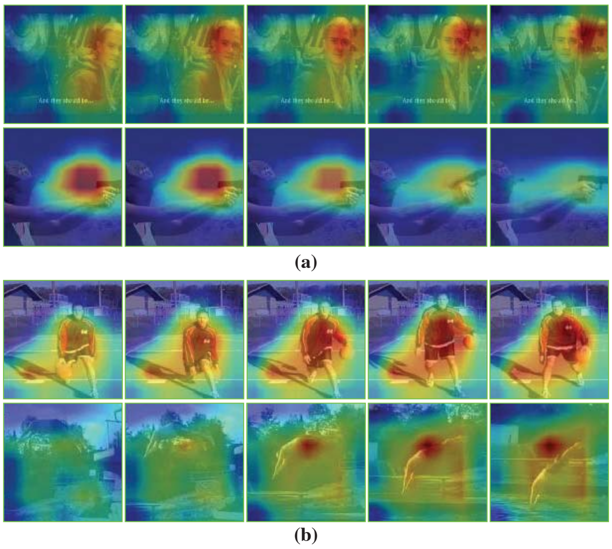


Figure 8: **Failure cases.** (a) Square activation patterns originated from the prior encoded by MoSI that the motion areas have a square shape. (b) The movement in the background causes confusion for the network. Examples show the model is confused by the movement of the shadow in the background and the scene respectively.

4.4. Discussions

Previous sections have shown that the proposed MoSI framework can train the model to focus on a local area with prominent motions. Despite that the models trained by MoSI achieves a satisfactory improvement on the action recognition task by learning to attend to motions, there are certain limitations. (A) Because of the square shape of the prior unmasked motion area in MoSI, in some cases, the model is somewhat biased toward attending to a square area as visualized in Fig. 8(a). (B) Because no semantic information is encoded during MoSI training (see Sec. 4.1), major movements in the background could also confuse the model, as in Fig. 8(b).

Conclusion. This work proposes MoSI, a simple framework for the video models to learn motion representations from images. It is shown that MoSI can discover and attend to prominent motions in videos, thus yielding a strong representation for the downstream action recognition task. We also demonstrate the possibility of using MoSI to train video models on image datasets. It is hoped that this research can inspire further study in understanding how motions can be encoded into video representations.

Acknowledgement. This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project A18A2b0046) and by Alibaba Group through Alibaba Research Intern Program.

References

[1] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *CVPR*, pages 9922–9931, 2020. 2, 3, 8

[2] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, pages 25–36. Springer, 2004. 1

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 2

[4] Nieves Crasto, Philippe Weinzaepfel, Karteek Alahari, and Cordelia Schmid. Mars: Motion-augmented rgb stream for action recognition. In *CVPR*, pages 7882–7891, 2019. 2

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 1, 2, 5

[6] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015. 2, 4

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, pages 6202–6211, 2019. 1

[8] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, pages 1933–1941, 2016. 2

[9] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, pages 3636–3645, 2017. 2

[10] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2, 3

[11] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *ICCV Workshops*, pages 0–0, 2019. 2, 4, 5, 8

[12] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. *arXiv preprint arXiv:2008.01065*, 2020. 2, 5, 8

[13] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, pages 6546–6555, 2018. 1, 2

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[15] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018. 2

[16] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *ICCV*, pages 2000–2009, 2019. 1, 2

[17] Longlong Jing and Yingli Tian. Self-supervised spatiotemporal feature learning by video geometric transformations. *arXiv preprint arXiv:1811.11387*, 2(7):8, 2018. 8

[18] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014. 1

[19] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1, 5

[20] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, volume 33, pages 8545–8552, 2019. 2, 8

[21] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1920–1929, 2019. 7

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 1

[23] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563. IEEE, 2011. 5

[24] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, pages 667–676, 2017. 2, 8

[25] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *CVPR*, pages 510–519, 2019. 2

[26] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *CVPR*, pages 909–918, 2020. 2

[27] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video cloze procedure for self-supervised spatio-temporal learning. *arXiv preprint arXiv:2001.00294*, 2020. 2, 8

[28] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 2

[29] T Nathan Mundhenk, Daniel Ho, and Barry Y Chen. Improvements to context based self-supervised learning. In *CVPR*, pages 9339–9348, 2018. 2

[30] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84. Springer, 2016. 2

[31] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *CVPR*, pages 9359–9367, 2018. 2

[32] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *CVPR*, pages 2701–2710, 2017. 2

[33] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016. 2

[34] Xiaojiang Peng, Limin Wang, Xingxing Wang, and Yu Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *Computer Vision and Image Understanding*, 150:109–125, 2016. 1

[35] Xiaojiang Peng, Changqing Zou, Yu Qiao, and Qiang Peng. Action recognition with stacked fisher vectors. In *ECCV*, pages 581–595. Springer, 2014. 1

[36] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, pages 5533–5541, 2017. 2

[37] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Xinmei Tian, and Tao Mei. Learning spatio-temporal representation with local and global diffusion. In *CVPR*, pages 12056–12065, 2019. 2

[38] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 1, 5

[39] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2020. 6

[40] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, pages 5734–5743, 2017. 1

[41] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, pages 568–576, 2014. 1, 2

[42] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5

[43] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *ICCV*, pages 5552–5561, 2019. 2

[44] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pages 6450–6459, 2018. 2, 5

[45] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176. IEEE, 2011. 1

[46] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558, 2013. 1

[47] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *CVPR*, pages 352–361, 2020. 1, 2

[48] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36. Springer, 2016. 2

[49] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 1

[50] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8052–8060, 2018. 4

[51] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *CVPR*, pages 10334–10343, 2019. 2, 5, 8

[52] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. Video playback rate perception for self-supervised spatio-temporal representation learning. In *CVPR*, pages 6548–6557, 2020. 2, 5, 8

[53] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, pages 649–666. Springer, 2016. 2

[54] Yue Zhao, Yuanjun Xiong, and Dahua Lin. Trajectory convolution for action recognition. In *NeurIPS*, pages 2204–2215, 2018. 1

[55] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *ICCV*, pages 2914–2923, 2017. 1