Video Representation Learning by Recognizing Temporal Transformations

Simon Jenni[0000−0002−9472−0425], Givi Meishvili[0000−0002−0984−7078], and Paolo Favaro[0000−0003−3546−8247]

University of Bern, Switzerland
{simon.jenni,givi.meishvili,paolo.favaro}@inf.unibe.ch

Abstract. We introduce a novel self-supervised learning approach to learn representations of videos that are responsive to changes in the motion dynamics. Our representations can be learned from data without human annotation and provide a substantial boost to the training of neural networks on small labeled data sets for tasks such as action recognition, which require to accurately distinguish the motion of objects. We promote an accurate learning of motion without human annotation by training a neural network to discriminate a video sequence from its temporally transformed versions. To learn to distinguish non-trivial motions, the design of the transformations is based on two principles: 1) To define clusters of motions based on time warps of different magnitude; 2) To ensure that the discrimination is feasible only by observing and analyzing as many image frames as possible. Thus, we introduce the following transformations: forward-backward playback, random frame skipping, and uniform frame skipping. Our experiments show that networks trained with the proposed method yield representations with improved transfer performance for action recognition on UCF101 and HMDB51.

Keywords: Representation Learning, Video Analysis, Self-Supervised Learning, Unsupervised Learning, Time Dynamics, Action Recognition

1 Introduction

A fundamental goal in computer vision is to build representations of visual data that can be used towards tasks such as object classification, detection, segmentation, tracking, and action recognition [39, 11, 41, 26]. In the past decades, a lot of research has been focused on learning directly from single images and has done so with remarkable success [38, 17, 18]. Single images carry crucial information about a scene. However, when we observe a temporal sequence of image frames, *i.e.*, a video, it is possible to understand much more about the objects and the scene. In fact, by moving, objects reveal their shape (through a change in the occlusions), their behavior (how they move due to the laws of Physics or their inner mechanisms), and their interaction with other objects (how they deform, break, clamp etc.). However, learning such information is non trivial. Even when labels related to motion categories are available (such as in action recognition),

there is no guarantee that the trained model will learn the desired information, and will not instead simply focus on a single iconic frame and recognize a key pose or some notable features strongly correlated to the action [40].

To build representations of videos that capture more than the information contained in a single frame, we pose the task of learning an accurate model of motion as that of learning to distinguish an unprocessed video from a temporally-transformed one. Since similar frames are present in both the unprocessed and transformed sequence, the only piece of information that allows their discrimination is their temporal evolution. This idea has been exploited in the past [12, 28, 29, 33, 50] and is also related to work in time-series analysis, where dynamic time warping is used as a distance for temporal sequences [20].

In this paper, we analyze different temporal transformations and evaluate how learning to distinguish them yields a representation that is useful to classify videos into meaningful action categories. Our main finding is that the most effective temporal distortions are those that can be identified only by observing the largest number of frames. For instance, the case of substituting the second half of a video with its first half in reverse order, can be detected already by comparing just the 3 frames around the temporal symmetry. In contrast, distinguishing when a video is played backwards from when it is played forward [50] may require observing many frames. Thus, one can achieve powerful video representations by using as pseudo-task the classification of temporal distortions that differ in their long-range motion dynamics. Towards this goal, we investigate 4 different temporal transformations of a video, which are illustrated in Fig. 1:

1. **Speed**: Select a subset of frames with uniform sub-sampling (*i.e.*, with a fixed number of frames in between every pair of selected frames), while preserving the order in the original sequence;
2. **Random**: Select a random permutation of the frame sequence;
3. **Periodic**: Select a random subset of frames in their natural (forward) temporal order and then a random subset in the backward order;
4. **Warp**: Select a subset of frames with a random sub-sampling (*i.e.*, with a random number of frames in between every pair of selected frames), while preserving the natural (forward) order in the original sequence.

We use these transformations to verify and illustrate the hypothesis that learning to distinguish them from one another (and the original sequence) is useful to build a representation of videos for action recognition. For simplicity, we train a neural network that takes as input videos of the same duration and outputs two probabilities: One is about which one of the above temporal transformations the input sequence is likely to belong to and the second is about identifying the correct speed of the chosen **speed** transformation.

In the Experiments section, we transfer features of standard 3D-CNN architectures (C3D [44], 3D-ResNet [16], and R(2+1)D [45]) pre-trained through the above pseudo-task to standard action recognition data sets such as UCF101 and HMDB51, with improved performance compared to prior works. We also show that features learned through our proposed pseudo-task capture long-range motion better than features obtained through supervised learning. Our project

Fig. 1: **Learning from Temporal Transformations**. The frame number is indicated below each image. (a)-(d) `Speed` transformation by skipping: (a) 0 frames, (b) 1 frame, (c) 3 frames, and (d) 7 frames. (e) `Random`: frame permutation (no frame is skipped). (f) `Periodic`: forward-backward motion (at the selected speed). (g) `Warp`: variable frame skipping while preserving the order.

page `https://sjenni.github.io/temporal-ssl` provides code and additional experiments.

Our contributions can be summarized as follows: 1) We introduce a novel self-supervised learning task to learn video representations by distinguishing temporal transformations; 2) We study the discrimination of the following novel temporal transformations: **speed**, **periodic** and **warp**; 3) We show that our features are a better representation of motion than features obtained through supervised learning and achieve state of the art transfer learning performance on action recognition benchmarks.

2 Prior Work

Because of the lack of manual annotation, our method belongs to self-supervised learning. Self-supervised learning appeared in the machine learning literature

more than 2 decades ago [7, 2] and has been reformulated recently in the context of visual data with new insights that make it a promising method for representation learning [9]. This learning strategy is a recent variation on the unsupervised learning theme, which exploits labeling that comes for "free" with the data. Labels could be easily accessible and associated with a non-visual signal (for example, ego-motion [1], audio [35], text and so on), but also could be obtained from the structure of the data (*e.g.*, the location of tiles [9, 34], the color of an image [53, 54, 27]) or through transformations of the data [14, 21, 22]. Several works have adapted self-supervised feature learning methods from domains such as images or natural language to videos: Rotation prediction [23], Dense Predictive Coding [15], and [43] adapt the BERT language model [8] to sequences of frame feature vectors.

In the case of videos, we identify three groups of self-supervised approaches: 1) Methods that learn from videos to represent videos; 2) Methods that learn from videos to represent images; 3) Methods that learn from videos and auxiliary signals to represent both videos and the auxiliary signals (*e.g.*, audio).

Temporal ordering methods. Prior work has explored the temporal ordering of the video frames as a supervisory signal. For example, Misra *et al.* [33] showed that learning to distinguish a real triplet of frames from a shuffled one yields a representation with temporally varying information (*e.g.*, human pose). This idea has been extended to longer sequences for posture and behavior analysis by using Long Short-Term Memories [5]. The above approaches classify the correctness of a temporal order directly from one sequence. An alternative is to feed several sequences, some of which are modified, and ask the network to tell them apart [12]. Other recent work predicts the permutation of a sequence of frames [28] or both the spatial and temporal ordering of frame patches [6, 24]. Another recent work focuses on solely predicting the arrow of time in videos [50]. Three concurrent publications also exploit the playback speed as a self-supervision signal [10, 4, 52]. In contrast, our work studies a wider range of temporal transformations. Moreover, we show empirically that the temporal statistics extent (in frames) captured by our features correlates to the transfer learning performance in action recognition.

Methods based on visual tracking. The method of Wang and Gupta [48] builds a metric to define similarity between patches. Three patches are used as input, where two patches are matched via tracking in a video and the third one is arbitrarily chosen. Tracking can also be directly solved during training, as shown in [46], where color is used as a supervisory signal. By solving the task of coloring a grey-scale video (in a coherent manner across time), one can automatically learn how to track objects. Visual correspondences can also be learned by exploiting cycle-consistency in time [49] or by jointly performing region-level localization and fine-grained matching [29]. However, although trained on videos, these methods have not been used to build video representations or evaluated on action recognition.

Methods based on auxiliary signals. Supervision can also come from additional signals recorded with images. For example, videos come also with audio.

The fact that the sounds are synchronized with the motion of objects in a video, already provides a weak supervision signal: One knows the set of possible sounds of visible objects, but not precisely their correspondence. Owens *et al.* [36] show that, through the process of predicting a summary of ambient sound in video frames, a neural network learns a useful representation of objects and scenes. Another way to learn a similar representation is via classification [3]: A network is given an image-sound pair as input and must classify whether they match or not. Korbar *et al.* [25] build audio and video representations by learning to synchronize audio and video signals using a contrastive loss. Recently, [37] use multi-modal data from videos also in a contrastive learning framework. Several methods use optical flow as a supervision signal. For example, Wang *et al.* [47] extract motion and appearance statistics. Luo *et al.* [32] predict future atomic 3D flows given an input sequence, and Gan *et al.* [13] use geometry in the form of flow fields and disparity maps on synthetic and 3D movies. Optical flow is also used as input for video representation learning or filtering of the data [50]. Conversely, we do not make use of any auxiliary signals and learn video representations solely from the raw RGB frames.

3 Learning Video Dynamics

Recent work [47] showed how a careful learning of motion statistics led to a video representation with excellent transfer performance on several tasks and data sets. The learning of motion statistics was made explicit by extracting optical flow between frame pairs, by computing flow changes, and then by identifying the region where a number of key attributes (*e.g.*, maximum magnitude and orientation) of the time-averaged flow-change occurred. In this work, we also aim to learn from motion statistics, but we focus entirely our attention on the temporal evolution without specifying motion attributes of interest or defining a task based on appearance statistics. We hypothesize that these important aspects could be implicitly learned and exploited by the neural network to solve the lone task of discriminating temporal transformations of a video. Our objective is to encourage the neural network to represent well motion statistics that require a long-range observation (in the temporal domain). To do so, we train the network to discriminate videos where the image content has been preserved, but not the temporal domain. For example, we ask the network to distinguish a video at the original frame rate from when it is played 4 times faster. Due to the laws of Physics, one can expect that, in general, *executing* the same task at different speeds leads to different motion dynamics compared to when a video is just *played* at different speeds (*e.g.*, compare marching vs walking played at a higher speed). Capturing the subtleties of the dynamics of these motions requires more than estimating motion between 2 or 3 frames. Moreover, these subtleties are specific to the moving object, and thus they require object detection and recognition.

In our approach, we transform videos by sampling frames according to different schemes, which we call *temporal transformations*. To support our learning

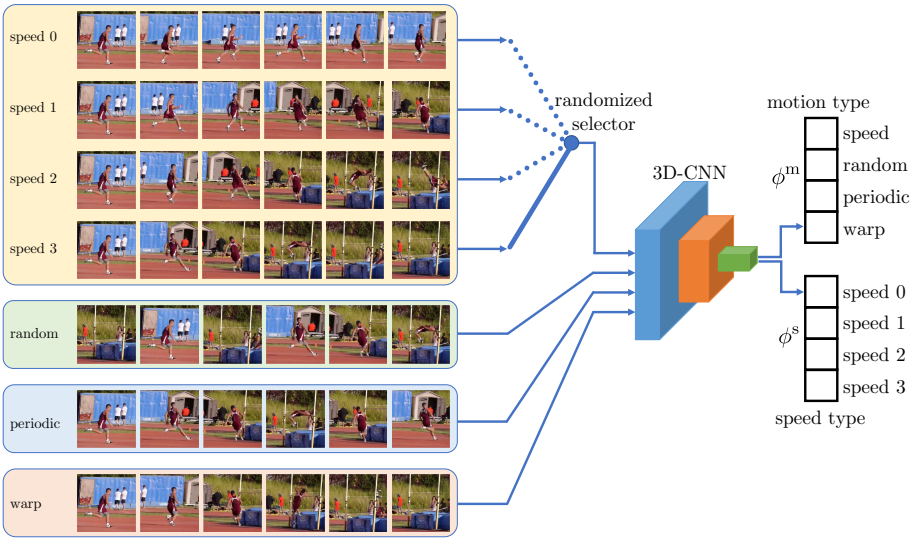

Fig. 2: **Training a 3D-CNN to distinguish temporal transformations.** In each mini-batch we select a video speed (out of 4 possible choices), *i.e.*, how many frames are skipped in the original video. Then, the 3D-CNN receives as input mini-batch a mixture of 4 possible transformed sequences: `speed` (with the chosen frame skipping), `random`, `periodic` and `warp`. The network outputs the probability of which motion type a sequence belongs to and the probability of which speed type the speed-transformed sequence has.

hypothesis, we analyze transformations that require short- (*i.e.*, temporally local) and long-range (*i.e.*, temporally global) video understanding. As will be illustrated in the Experiments section, short-range transformations yield representations that transfer to action recognition with a lower performance than long-range ones.

3.1 Transformations of Time

Fig. 2 illustrates how we train our neural network (a 3D-CNN [44]) to build a video representation (with 16 frames). In this section, we focus on the inputs to the network. As mentioned above, our approach is based on distinguishing different temporal transformations. We consider 4 fundamental types of transformations: Speed changes, random temporal permutations, periodic motions and temporal warp changes. Each of these transformations boils down to picking a sequence of temporal indices to sample the videos in our data set. $\mathcal{V}^{\tau}_{\kappa} \subset \{0, 1, 2, \dots\}$ denotes the chosen subset of indices of a video based on the transformation $\tau \in \{0, 1, 2, 3\}$ and with speed κ.

Speed ($\tau = 0$): In this first type we artificially change the video frame rate, *i.e.*,

its playing speed. We achieve that by skipping a different number of frames. We consider 4 cases, **Speed 0, 1, 2, 3** corresponding to $\kappa = 0,\ 1,\ 2,\ 3$ respectively, where we skip $2^\kappa - 1$ frames. The resulting playback speed of **Speed κ** is therefore 2^κ times the original speed. In the generation of samples for the training of the neural network we first uniformly sample $\kappa \in \{0, 1, 2, 3\}$, the playback speed, and then use this parameter to define other transformations. This sequence is used in all experiments as one of the categories against either other speeds or against one of the other transformations below. The index sequence $\mathcal{V}_\kappa^0$ is thus $\rho + [0, 1 \cdot 2^\kappa, 2 \cdot 2^\kappa, \ldots, 15 \cdot 2^\kappa]$, where ρ is a random initial index.

Random ($\tau = 1$): In this second temporal transformation we randomly permute the indices of a sequence without skipping frames. We fix $\kappa = 0$ to ensure that the maximum frame skip between two consecutive frames is not too dissimilar to other transformations. This case is used as a reference, as random permutations can often be detected by observing only a few nearby frames. Indeed, in the Experiments section one can see that this transformation yields a low transfer performance. The index sequence $\mathcal{V}_0^1$ is thus $\rho + \text{permutation}([0, 1, 2, \ldots, 15])$. This transformation is similar to that of the pseudo-task of Misra *et al.* [33].

Periodic ($\tau = 2$): This transformation synthesizes motions that exhibit approximate periodicity. To create such artificial cases we first pick a point $2 \cdot 2^\kappa < s < 13 \cdot 2^\kappa$ where the playback direction switches. Then, we compose a sequence with the following index sequence: 0 to s and then from $s - 1$ to $2s - 15 \cdot 2^\kappa$. Finally, we sub-sample this sequence by skipping $2^\kappa - 1$ frames. Notice that the randomization of the midpoint s in the case of $\kappa > 0$ yields pseudo-periodic sequences, where the frames in the second half of the generated sequence often do not match the frames in the first half of the sequence. The index sequence $\mathcal{V}_\kappa^2$ is thus $\rho + [0, 1 \cdot 2^\kappa, 2 \cdot 2^\kappa, \ldots, \bar{s} \cdot 2^\kappa, (\bar{s} - 1) \cdot 2^\kappa + \delta, \ldots, (2\bar{s} - 15) \cdot 2^\kappa + \delta])$, where $\bar{s} = \lfloor s / 2^\kappa \rfloor$, $\delta = s - \bar{s} \cdot 2^\kappa$, and $\rho = \max(0, (15 - 2\bar{s}) \cdot 2^\kappa - \delta)$.

Warp ($\tau = 3$): In this transformation, we pick a set of 16 ordered indices with a non-uniform number of skipped frames between them (we consider sampling any frame so we let $\kappa = 0$). In other words, between any of the frames in the generated sequence we have a random number of skipped frames, each chosen independently from the set $\{0, \ldots, 7\}$. This transformation creates a warping of the temporal dimension by varying the playback speed from frame to frame. To construct the index sequence $\mathcal{V}_0^3$ we first sample the frame skips $s_j \in \{0, \ldots, 7\}$ for $j = 1, \ldots, 15$ and set $\mathcal{V}_0^3$ to $\rho + [0, s_1, s_1 + s_2, \ldots, \sum_{j=1}^{15} s_j]$.

3.2 Training

Let ϕ denote our network, and let us denote with ϕ^{m} (`motion`) and ϕ^{s} (`speed`) its two softmax outputs (see Fig. 2). To train ϕ we optimize the following loss

$$-\text{E}_{\kappa \sim \mathcal{U}[0,3], p \in \mathcal{V}_\kappa^0, q \in \mathcal{V}_0^1, s \in \mathcal{V}_\kappa^2, t \in \mathcal{V}_0^3, x} \Big[\log \left(\phi_0^{\text{m}}(x_p)\, \phi_1^{\text{m}}(x_q)\, \phi_2^{\text{m}}(x_s)\, \phi_3^{\text{m}}(x_t) \right) \Big] \quad (1)$$

$$-\text{E}_{\kappa \sim \mathcal{U}[0,3], p \in \mathcal{V}_\kappa^0, x} \big[\log \left(\phi_\kappa^{\text{s}}(x_p) \right) \big]$$

where x is a video sample, the sub-index denotes the set of frames. This loss is the cross entropy both for `motion` and `speed` classification (see Fig. 2).

3.3 Implementation

Following prior work [47], we use the smaller variant of the C3D architecture [44] for the 3D-CNN transformation classifier in most of our experiments. Training was performed using the AdamW optimizer [31] with parameters $\beta_1 = 0.9, \beta_2 = 0.99$ and a weight decay of 10^{-4}. The initial learning rate was set to $3 \cdot 10^{-4}$ during pre-training and $5 \cdot 10^{-5}$ during transfer learning. The learning rate was decayed by a factor of 10^{-3} over the course of training using cosine annealing [30] both during pre-training and transfer learning. We use batch-normalization [19] in all but the last layer. Mini-batches are constructed such that all the different coarse time warp types are included for each sampled training video. The batch size is set 28 examples (including all the transformed sequences). The speed type is uniformly sampled from all the considered speed types. Since not all the videos allow a sampling of all speed types (due to their short video duration) we limit the speed type range to the maximal possible speed type in those examples. We use the standard pre-processing for the C3D network. In practice, video frames are first resized to 128×171 pixels, from which we extract random crops of size 112×112 pixels. We also apply random horizontal flipping of the video frames during training. We use only the raw unfiltered RGB video frames as input to the motion classifier and do not make use of optical flow or other auxiliary signals.

4 Experiments

Datasets and Evaluation. In our experiments we consider three datasets. Kinetics [55] is a large human action dataset consisting of around 500K videos. Video clips are collected from YouTube and span 600 human action classes. We use the training split for self-supervised pre-training. UCF101 [41] contains around 13K video clips spanning 101 human action classes. HMDB51 [26] contains around 5K videos belonging to 51 action classes. Both UCF101 and HMDB51 come with three pre-defined train and test splits. We report the average performance over all splits for transfer learning experiments. We use UCF101 train split 1 for self-supervised pre-training. For transfer learning experiments we skip 3 frames corresponding to transformation **Speed 2**. For the evaluation of action recognition classifiers in transfer experiments we use as prediction the maximum class probability averaged over all center-cropped sub-sequences for each test video. More details are provided in the supplementary material.

Understanding the Impact of the Temporal Transformations. We perform ablation experiments on UCF101 and HMDB51 where we vary the number of different temporal transformations the 3D-CNN is trained to distinguish. The 3D-CNN is pre-trained for 50 epochs on UCF101 with our self-supervised learning task. We then perform transfer learning for action recognition on UCF101 and HMDB51. On UCF101 we freeze the weights of the convolutional layers and train three randomly initialized fully-connected layers for action recognition. This experiment treats the transformation classifier as a fixed video feature extractor. On HMDB51 we fine-tune the whole network including convolutional layers on the target task. This experiment therefore measures the quality of

Table 1: **Ablation experiments.** We train a 3D-CNN to distinguish different sets of temporal transformations. The quality of the learned features is evaluated through transfer learning for action recognition on UCF101 (with frozen convolutional layers) and HMDB51 (with fine-tuning of the whole network).

Pre-Training Signal	`speed` loss	**UCF101** (conv frozen)	**HMDB51** (conv fine-tuned)
Action Labels UCF101	-	60.7%	28.8%
Speed	YES	49.3%	32.5%
Speed + Random	NO	44.5%	31.7%
Speed + Periodic	NO	40.6%	29.5%
Speed + Warp	NO	43.5%	32.6%
Speed + Random	YES	55.1%	33.2%
Speed + Periodic	YES	56.5%	36.1%
Speed + Warp	YES	55.8%	36.9%
Speed + Random + Periodic	NO	47.4%	30.1%
Speed + Random + Warp	NO	54.8%	36.6%
Speed + Periodic + Warp	NO	50.6%	36.4%
Speed + Random + Periodic	YES	60.0%	37.1%
Speed + Random + Warp	YES	60.4%	39.2%
Speed + Periodic + Warp	YES	59.5%	39.0%
Speed + Random + Periodic + Warp	NO	54.2%	34.9%
Speed + Random + Periodic + Warp	YES	60.6%	38.0%

the network initialization obtained through self-supervised pre-training. In both cases we again train for 50 epochs on the action recognition task. The results of the ablations are summarized in Table 1. For reference we also report the performance of network weights learned through supervised pre-training on UCF101.

We observe that when considering the impact of a single transformation across different cases, the types **Warp** and **Speed** achieve the best transfer performance. With the same analysis, the transformation **Random** leads to the worst transfer performance on average. We observe that **Random** is also the easiest transformation to detect (based on training performance – not reported). As can be seen in Fig. 1 (e) this transformation can lead to drastic differences between consecutive frames. Such examples can therefore be easily detected by only comparing pairs of adjacent frames. In contrast, the motion type **Warp** can not be distinguished based solely on two adjacent frames and requires modelling long range dynamics. We also observe that distinguishing a larger number of transformations generally leads to an increase in the transfer performance. The effect of the **speed** type classification is quite noticeable. It leads to a very significant transfer performance increase in all cases. This is also the most difficult pseudo task (based on the training performance – not reported). Recognizing the speed of an action is indeed challenging, since different action classes naturally

Table 2: **Comparison to prior work on self-supervised video representation learning.** Whenever possible we compare to results reported with the same data modality we used, *i.e.*, unprocessed RGB input frames. * are our reimplementations.

Method	Ref	Network	Train Dataset	UCF101	HMDB51
Shuffle&Learn [33]	[33]	AlexNet	UCF101	50.2%	18.1%
O3N [12]	[12]	AlexNet	UCF101	60.3%	32.5%
AoT [50]	[50]	VGG-16	UCF101	78.1%	-
OPN [28]	[28]	VGG-M-2048	UCF101	59.8%	23.8%
DPC [15]	[15]	3D-ResNet34	Kinetics	75.7%	35.7%
SpeedNet [4]	[4]	S3D-G	Kinetics	81.1%	48.8%
AVTS [25] (RGB+audio)	[25]	MC3	Kinetics	85.8%	56.9%
Shuffle&Learn [33]*	-	C3D	UCF101	55.8%	25.4%
3D-RotNet [23]*	-	C3D	UCF101	60.6%	27.3%
Clip Order [51]	[51]	C3D	UCF101	65.6%	28.4%
Spatio-Temp [47]	[47]	C3D	UCF101	58.8%	32.6%
Spatio-Temp [47]	[47]	C3D	Kinetics	61.2%	33.4%
3D ST-puzzle [24]	[24]	C3D	Kinetics	60.6%	28.3%
Ours	-	C3D	UCF101	68.3%	38.4%
Ours	-	C3D	Kinetics	**69.9%**	**39.6%**
3D ST-puzzle [24]	[24]	3D-ResNet18	Kinetics	65.8%	33.7%
3D RotNet [23]	[23]	3D-ResNet18	Kinetics	66.0%	37.1%
DPC [15]	[15]	3D-ResNet18	Kinetics	68.2%	34.5%
Ours	-	3D-ResNet18	UCF101	77.3%	47.5%
Ours	-	3D-ResNet18	Kinetics	**79.3%**	**49.8%**
Clip Order [51]	[51]	R(2+1)D	UCF101	72.4%	30.9%
PRP [52]	[52]	R(2+1)D	UCF101	72.1%	35.0%
Ours	-	R(2+1)D	UCF101	**81.6%**	**46.4%**

exhibit widely different motion speeds (*e.g.*, "applying make-up" vs. "biking"). This task might often require a deeper understanding of the physics and objects involved in the video. Notice also that our pre-training strategy leads to a better transfer performance on HMDB51 than supervised pre-training using action labels. This suggests that the video dynamics learned through our pre-training generalize well to action recognition and that such dynamics are not well captured through the lone supervised action recognition.

Transfer to UCF101 and HMDB51. We compare to prior work on self-supervised video representation learning in Table 2. A fair comparison to much of the prior work is difficult due to the use of very different network architectures and training as well as transfer settings. We opted to compare with some commonly used network architectures (*i.e.*, C3D, 3D-ResNet, and R(2+1)D) and re-implemented two prior works [33] and [23] using C3D. We performed self-supervised pre-training on UCF101 and Kinetics. C3D is pre-trained for 100 epochs on UCF101 and for 15 epoch on Kinetics. 3D-ResNet and R(2+1)D are

Fig. 3: **Time-Related Pseudo-Tasks**. (a) Synchronization problem: The network is given two sequences with a time delay (4 frames in the example) and a classifier is trained to determine the delay. (b) The before-after problem: The network is given two non-overlapping sequences, and it needs to determine which comes first (the bottom sequence after the top one in the example).

both pre-trained for 200 epochs on UCF101 and for 15 epochs on Kinetics. We fine-tune all the layers for action recognition. Fine-tuning is performed for 75 epochs using C3D and for 150 epochs with the other architectures. When pre-training on UCF101 our features outperform prior work on the same network architectures. Pre-training on Kinetics leads to an improvement in transfer in all cases.

Long-Range vs Short-Range Temporal Statistics. To illustrate how well our video representations capture motion, we transfer them to other pseudo-tasks that focus on the temporal evolution of a video. One task is the classification of the synchronization of video pairs, *i.e.*, how many frames one video is delayed with respect to the other. A second task is the classification of two videos into which one comes first temporally. These two tasks are illustrated in Fig. 3. In the same spirit, we also evaluate our features on other tasks and data sets and we report the results at our project page `https://sjenni.github.io/temporal-ssl`.

For the synchronization task, two temporally overlapping video sequences x_1 and x_2 are separately fed to the pre-trained C3D network to extract features $\psi(v_1)$ and $\psi(v_2)$ at the `conv5` layer. These features are then fused through $\psi(v_1) - \psi(v_2)$ and fed as input to a randomly initialized classifier consisting of three fully-connected layers trained to classify the offset between the two sequences. We consider random offsets between the two video sequences in the range -6 to +6. For the second task we construct a single input sequence by sampling two non-overlapping 8 frame sub-sequences x_{i1} and x_{i2}, where x_{i1} comes before x_{i2}. The network inputs are then either (x_{i1}, x_{i2}) for class "before" or (x_{i2}, x_{i1}) for the class "after". We reinitialize the fully-connected layers in this case as well.

In Table 3 we compare the performance of different pre-training strategies on the time-related pseudo-tasks. We see that our self-supervised features perform

Table 3: **Time-Related Pseudo-Tasks.** We examine how well features from different pre-training strategies can be transferred to time-related tasks on videos. As tasks we consider the synchronization of two overlapping videos and the temporal ordering of two non-overlapping videos. We report the accuracy on both tasks on the UCF101 test set and also report Mean Absolute Error (MAE) for the synchronization task. * are our reimplementations.

	Sync.		**Before-After**
Method	Accuracy	MAE	Accuracy
Action Labels (UCF101)	36.7%	<u>1.85</u>	66.6%
3D-RotNet [23]*	28.0%	2.84	57.8%
Shuffle&Learn [33]*	<u>39.0%</u>	1.89	<u>69.8%</u>
Ours	**42.4**%	**1.61**	**76.9**%

better at these tasks than supervised features and other self-supervised features, thus showing that they capture well the temporal dynamics in the videos.

Visualization. What are the attributes, factors or features of the videos that self-supervised and supervised models are extracting to perform the final classification? To examine what the self-supervised and supervised models focus on, we apply Guided Backpropagation [42]. This method allows us to visualize which part of the input has the most impact on the final decision of the model. We slightly modify the procedure by subtracting the median values from every frame of the gradient video and by taking the absolute value of the result. We visualize the pre-trained self-supervised and supervised models on several test samples from UCF101. As one can see in Fig. 4, a model pre-trained on our self-supervised task tends to ignore the background and focuses on persons performing an action and on moving objects. Models trained with supervised learning on the other hand tend to focus more on the appearance of foreground and background. Another observation we make is that the self-supervised model identifies the location of moving objects/people in past and future frames. This is visible in row number 2 of blocks *(a)* and *(c)* of Fig. 4, where the network tracks the possible locations of the moving ping-pong and billiard balls respectively. A possible explanation for this observation is that our self-supervised task only encourages the learning of dynamics. The appearance of non-moving objects or static backgrounds are not useful to solve the pretext task and are thus ignored.

Learning Dynamics vs. Frame Features. The visualizations in Fig. 4 indicate that features learned through motion discrimination focus on the dynamics in videos and not so much on static content present in single frames (*e.g.*, background) when compared to supervised features. To further investigate how much the features learned through the two pre-training strategies rely on motion, we performed experiments where we remove all the dynamics from videos. To this end, we create input videos by replicating a single frame 16 times (resulting in a still video) and train the three fully-connected layers on `conv5` features for action classification on UCF101. Features obtained through supervised pre-

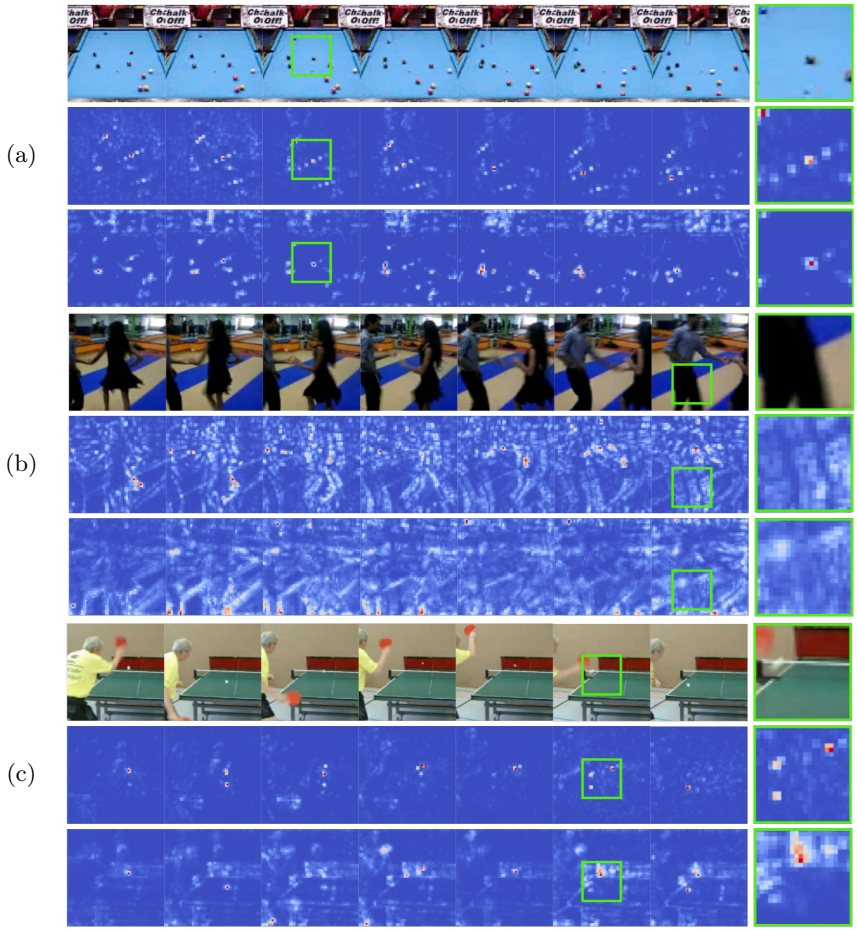

Fig. 4: **Visualization of active pixels**. The first row in each block corresponds to the input video. Rows two and three show the output of our adaptation of Guided Backpropagation [42] when applied to a network trained through self-supervised learning and supervised learning respectively. In all three cases we observe that the self-supervised network focuses on image regions of moving objects or persons. In (a) we can also observe how long range dynamics are being detected by the self-supervised model. The supervised model on the other hand focuses a lot on static frame features in the background.

training achieve an accuracy of 18.5% (vs. 56.5% with dynamics) and features from our self-supervised task achieve 1.0% (vs. 58.1%). Although the setup in this experiment is somewhat contrived (since the input domain is altered) it still illustrates that our features rely almost exclusively on motion instead of features present in single frames. This can be advantageous since motion features might generalize better to variations in the background appearance in many cases.

Table 4: **Video Retrieval Performance on UCF101.** We compare to prior work in terms of k-nearest neighbor retrieval accuracy. Query videos are taken from test split 1 and retrievals are computed on train split 1. A query is correctly classified if the query class is present in the top-k retrievals. We report mean retrieval accuracy for different values of k.

Method	**Network**	**Top1**	**Top5**	**Top10**	**Top20**	**Top50**
Jigsaw [34]	AlexNet	19.7	28.5	33.5	40.0	49.4
OPN [28]	AlexNet	19.9	28.7	34.0	40.6	51.6
Büchler *et al.* [6]	AlexNet	25.7	36.2	42.2	49.2	59.5
Clip Order [51]	R3D	14.1	30.3	40.0	51.1	66.5
SpeedNet [4]	S3D-G	13.0	28.1	37.5	49.5	65.0
PRP [52]	R3D	22.8	38.5	46.7	55.2	69.1
Ours	3D-ResNet18	**26.1**	**48.5**	**59.1**	**69.6**	**82.8**

Nearest-Neighbor Evaluation. We perform an additional quantitative evaluation of the learned video representations via the nearest-neighbor retrieval. The features are obtained by training a 3D-ResNet18 network on Kinetics with our pseudo-task and are chosen as the output of the global average pooling layer, which corresponds to a vector of size 512. For each video we extract and average features of 10 temporal crops. To perform the nearest-neighbor retrieval, we first normalize the features using the training set statistics. Cosine similarity is used as the metric to determine the nearest neighbors. We follow the evaluation proposed by [6] on UCF101. Query videos are taken from test split 1 and all the videos of train split 1 are considered as retrieval targets. A query is considered correctly classified if the k-nearest neighbors contain at least one video of the correct class (*i.e.*, same class as the query). We report the mean accuracy for different values of k and compare to prior work in Table 4. Our features achieve state-of-the-art performance.

5 Conclusions

We have introduced a novel task for the self-supervised learning of video representations by distinguishing between different types of temporal transformations. This learning task is based on the principle that recognizing a transformation of time requires an accurate model of the underlying natural video dynamics. This idea is supported by experiments that demonstrate that features learned by distinguishing time transformations capture video dynamics more than supervised learning and that such features generalize well to classic vision tasks such as action recognition or time-related task such as video synchronization.

Acknowledgements. This work was supported by grants 169622&165845 of the Swiss National Science Foundation.

References

1. Agrawal, P., Carreira, J., Malik, J.: Learning to see by moving. The IEEE International Conference on Computer Vision (ICCV) (December 2015)
2. Ando, R., Zhang, T.: A framework for learning predictive structures from multiple tasks and unlabeled data. JMLR (2005)
3. Arandjelovic, R., Zisserman, A.: Look, listen and learn. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 609–617. IEEE (2017)
4. Benaim, S., Ephrat, A., Lang, O., Mosseri, I., Freeman, W.T., Rubinstein, M., Irani, M., Dekel, T.: Speednet: Learning the speediness in videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9922–9931 (2020)
5. Brattoli, B., Büchler, U., Wahl, A.S., Schwab, M.E., Ommer, B.: Lstm self-supervision for detailed behavior analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 2 (2017)
6. Büchler, U., Brattoli, B., Ommer, B.: Improving spatiotemporal self-supervision by deep reinforcement learning. arXiv preprint arXiv:1807.11293 (2018)
7. Caruana, R., de Sa, V.R.: Promoting poor features to supervisors: Some inputs work better as outputs. NIPS (1996)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
9. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. ICCV (2015)
10. Epstein, D., Chen, B., Vondrick, C.: Oops! predicting unintentional action in video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 919–929 (2020)
11. Everingham, M., Zisserman, A., Williams, C., Van-Gool, L.: The pascal visual object classes challenge. VOC (2006)
12. Fernando, B., Bilen, H., Gavves, E., Gould, S.: Self-supervised video representation learning with odd-one-out networks. In: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. pp. 5729–5738. IEEE (2017)
13. Gan, C., Gong, B., Liu, K., Su, H., Guibas, L.J.: Geometry guided convolutional neural networks for self-supervised video representation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5589–5597 (2018)
14. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=S1v4N2l0-
15. Han, T., Xie, W., Zisserman, A.: Video representation learning by dense predictive coding. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 0–0 (2019)
16. Hara, K., Kataoka, H., Satoh, Y.: Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 6546–6555 (2018)
17. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

19. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
20. Iwana, B.K., Uchida, S.: Time series classification using local distance-based features in multi-modal fusion networks. Pattern Recognition **97**, 107024 (2020). https://doi.org/https://doi.org/10.1016/j.patcog.2019.107024, http://www.sciencedirect.com/science/article/pii/S0031320319303279
21. Jenni, S., Favaro, P.: Self-supervised feature learning by learning to spot artifacts. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2733–2742 (2018)
22. Jenni, S., Jin, H., Favaro, P.: Steering self-supervised feature learning beyond local pixel statistics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6408–6417 (2020)
23. Jing, L., Yang, X., Liu, J., Tian, Y.: Self-supervised spatiotemporal feature learning via video rotation prediction. arXiv preprint arXiv:1811.11387 (2018)
24. Kim, D., Cho, D., Kweon, I.S.: Self-supervised video representation learning with space-time cubic puzzles. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 8545–8552 (2019)
25. Korbar, B., Tran, D., Torresani, L.: Cooperative learning of audio and video models from self-supervised synchronization. In: Advances in Neural Information Processing Systems. pp. 7763–7774 (2018)
26. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: a large video database for human motion recognition. In: Proceedings of the International Conference on Computer Vision (ICCV) (2011)
27. Larsson, G., Maire, M., Shakhnarovich, G.: Colorization as a proxy task for visual understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6874–6883 (2017)
28. Lee, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Unsupervised representation learning by sorting sequences. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 667–676 (2017)
29. Li, X., Liu, S., De Mello, S., Wang, X., Kautz, J., Yang, M.H.: Joint-task self-supervised learning for temporal correspondence. In: Advances in Neural Information Processing Systems. pp. 317–327 (2019)
30. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
31. Loshchilov, I., Hutter, F.: Fixing weight decay regularization in adam. arXiv preprint arXiv:1711.05101 (2017)
32. Luo, Z., Peng, B., Huang, D.A., Alahi, A., Fei-Fei, L.: Unsupervised learning of long-term motion dynamics for videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2203–2212 (2017)
33. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and learn: unsupervised learning using temporal order verification. In: European Conference on Computer Vision. pp. 527–544. Springer (2016)
34. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: European Conference on Computer Vision. pp. 69–84. Springer (2016)
35. Owens, A., Efros, A.A.: Audio-visual scene analysis with self-supervised multisensory features. In: The European Conference on Computer Vision (ECCV) (September 2018)
36. Owens, A., Wu, J., McDermott, J.H., Freeman, W.T., Torralba, A.: Ambient sound provides supervision for visual learning. In: European Conference on Computer Vision. pp. 801–816. Springer (2016)

37. Patrick, M., Asano, Y.M., Fong, R., Henriques, J.F., Zweig, G., Vedaldi, A.: Multi-modal self-supervision from generalized data transformations. arXiv preprint arXiv:2003.04298 (2020)
38. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
39. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
40. Schindler, K., Van Gool, L.: Action snippets: How many frames does human action recognition require? In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2008)
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. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net (2014)
43. Sun, C., Baradel, F., Murphy, K., Schmid, C.: Contrastive bidirectional transformer for temporal representation learning. arXiv preprint arXiv:1906.05743 (2019)
44. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 4489–4497 (2015)
45. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 6450–6459 (2018)
46. Vondrick, C., Shrivastava, A., Fathi, A., Guadarrama, S., Murphy, K.: Tracking emerges by colorizing videos. In: Proc. ECCV (2018)
47. Wang, J., Jiao, J., Bao, L., He, S., Liu, Y., Liu, W.: Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4006–4015 (2019)
48. Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2794–2802 (2015)
49. Wang, X., Jabri, A., Efros, A.A.: Learning correspondence from the cycle-consistency of time. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2566–2576 (2019)
50. Wei, D., Lim, J., Zisserman, A., Freeman, W.T.: Learning and using the arrow of time. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8052–8060 (2018)
51. Xu, D., Xiao, J., Zhao, Z., Shao, J., Xie, D., Zhuang, Y.: Self-supervised spatiotemporal learning via video clip order prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10334–10343 (2019)
52. Yao, Y., Liu, C., Luo, D., Zhou, Y., Ye, Q.: Video playback rate perception for self-supervised spatio-temporal representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6548–6557 (2020)
53. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: European Conference on Computer Vision. pp. 649–666. Springer (2016)

54. Zhang, R., Isola, P., Efros, A.A.: Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1058–1067 (2017)
55. Zisserman, A., Carreira, J., Simonyan, K., Kay, W., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., et al.: The kinetics human action video dataset. ArXiv (2017)