Self-supervised Video Representation Learning by Pace Prediction

Jiangliu Wang[1], Jianbo Jiao[2], and Yun-Hui Liu[1*]

[1] The Chinese University of Hong Kong
[2] University of Oxford
{jlwang,yhliu}@mae.cuhk.edu.hk , jianbo@robots.ox.ac.uk

Abstract. This paper addresses the problem of self-supervised video representation learning from a new perspective – by video pace prediction. It stems from the observation that human visual system is sensitive to video pace, *e.g.*, slow motion, a widely used technique in film making. Specifically, given a video played in natural pace, we randomly sample training clips in different paces and ask a neural network to identify the pace for each video clip. The assumption here is that the network can only succeed in such a pace reasoning task when it understands the underlying video content and learns representative spatio-temporal features. In addition, we further introduce contrastive learning to push the model towards discriminating different paces by maximizing the agreement on similar video content. To validate the effectiveness of the proposed method, we conduct extensive experiments on action recognition and video retrieval tasks with several alternative network architectures. Experimental evaluations show that our approach achieves state-of-the-art performance for self-supervised video representation learning across different network architectures and different benchmarks. The code and pre-trained models are available at https://github.com/laura-wang/video-pace.
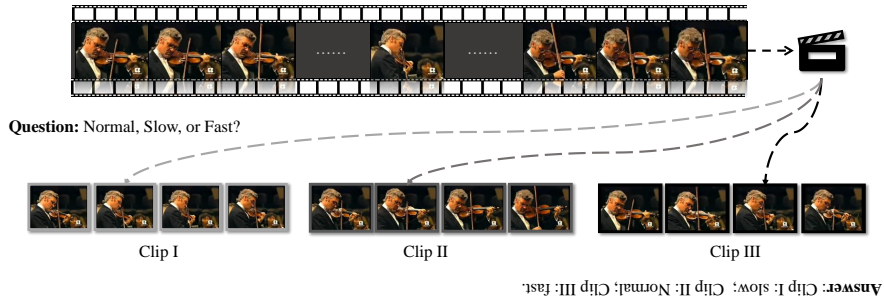
Keywords: Self-supervised learning · Video representation · Pace.

1 Introduction

Convolutional neural networks have witnessed absolute success in video representation learning [7, 47, 13] with human-annotated labels. Researchers have developed a wide range of neural networks [43, 46, 47] ingeniously, which extract powerful spatio-temporal representations for video understanding. Meanwhile, millions of labeled training data [26, 27] and powerful training resources are also the fundamental recipes for such great success. However, obtaining a large number of labeled video samples requires massive human annotations, which is expensive and time-consuming. Whereas at the same time, billions of unlabeled videos are available freely on the Internet. Therefore, video representation learning from unlabeled data is crucial for video understanding and analysis.

---

[*] Corresponding author.

Question: Normal, Slow, or Fast?

Clip I   Clip II   Clip III

Answer: Clip I: slow; Clip II: Normal; Clip III: fast.

Fig. 1. Illustration of the proposed pace prediction task. Given a video sample, frames are randomly selected by different paces to formulate the training inputs. Here, three different clips, *Clip I, II, III*, are sampled by normal, slow and fast pace randomly. Can you ascribe the corresponding pace label to each clip? (Answer is provided below.)

Among all the unsupervised learning paradigms, self-supervised learning is proved to be one promising methodology [38, 1]. The typical solution is to propose appropriate *pretext tasks* that generate free training labels automatically and encourage neural networks to learn transferable semantic spatio-temporal features for the *downstream tasks*. Such pretext tasks can be roughly divided into two categories: (1) generative dense prediction, such as flow fields prediction [15], future frame prediction [45, 48], *etc.* (2) discriminative classification/regression, such as video order prediction [36, 14, 34, 57, 28], rotation transformation prediction [25], motion and appearance statistics regression [51], *etc.* While promising results have been achieved, some of the approaches leverage pre-computed motion channels [51, 15], *e.g.*, optical flow, to generate the training labels. This could be both time and space consuming, especially when the pre-training dataset scales to millions/trillions of data. To alleviate such a problem, in this work, we propose a simple yet effective pretext task without referring to pre-computed motion. Instead, we only base on the original videos as input.

Inspired by the rhythmic montage in film making, we observe that human visual system is sensitive to motion pace and can easily distinguish different paces once understanding the covered content. Such a property has also been revealed in neuroscience studies [17, 54]. To this end, we propose a simple yet effective task to perform self-supervised video representation learning: pace prediction. Specifically, given videos played in natural pace, video clips with different paces are generated according to different temporal sampling rates. A learnable model is then trained to identify which pace the input video clip corresponds to. As aforementioned, the assumption here is that if the model is able to distinguish different paces, it has to understand the underlying content. Fig. 1 illustrates the basic idea of the proposed approach.

In the proposed pace prediction framework, we utilize 3D convolutional neural networks (CNNs) as our backbone network to learn video representation, following prior works [57, 35]. Specifically, we investigated several architectures,

including C3D [46], 3D-ResNet [47, 20], R(2+1)D [47], and S3D-G [56]. Furthermore, we incorporate contrastive learning to enhance the discriminative capability of the model for video understanding. Extensive experimental evaluations with several video understanding tasks demonstrate the effectiveness of the proposed approach. We also present a study of different backbone architectures as well as alternative configurations of contrastive learning. The experimental result suggests that the proposed approach can be well integrated into different architectures and achieves state-of-the-art performance for self-supervised video representation learning.

The main contributions of this work are summarized as follows.

- We propose a simple yet effective approach for self-supervised video representation learning by pace prediction. This novel pretext task provides a solution to learn spatio-temporal features without explicitly leveraging the motion channel, *e.g.*, optical flow.
- We further introduce contrastive learning to regularize the pace prediction objective. Two configurations are investigated by maximizing the mutual information either between same video pace or same video context.
- Extensive experimental evaluations on three network architectures and two downstream tasks across three datasets show that the proposed approach achieves state-of-the-art performance and demonstrates great potential to learn from tremendous amount of video data, in a simple manner.

2   Related Work

Video Representation Learning. Video understanding, especially action recognition, has been extensively studied for decades, where video representation learning serves as the fundamental problem of other video-related tasks, such as complex action recognition [23], action temporal localization [8, 41, 42], video caption [49, 52], *etc.* Initially, various hand-crafted local spatio-temporal descriptors are proposed for video representations, such as STIP [32], HOG3D [29], *etc.* Wang *et al.* [50] proposed improved dense trajectories (iDT) descriptors, which combined the effective HOG, HOF [33] and MBH descriptors [10], and achieved the best results among all hand-crafted features. With the impressive success of CNN in image understanding problem and the availability of large-scale video datasets such as sports1M [26], ActivityNet [5], Kinetics-400 [7], studies on data-driven deep learning-based video analysis started to emerge. According to the input modality, these video representation learning methods can be roughly divided into two categories: one is to directly take RGB videos as inputs, while the other is to take both RGB videos and optical flows as inputs. Tran *et al.* [46] extended the 2D convolution kernels to 3D and proposed C3D network to learn spatio-temporal representations. Simonyan and Zisserman [43] proposed a two-stream network that extracts spatial features from RGB inputs and temporal features from optical flows, followed by a fusion scheme. Recently, [13] proposed to capture the video information in a slow-fast manner, which showed

that input videos with slow and fast speed help the supervised action recognition task. Although impressive results have been achieved, video representation learning with human-annotated label is both time-consuming and expensive.

Self-supervised Learning. Self-supervised learning is becoming increasingly attractive due to its great potential to leverage the large amount of unlabeled data. The key idea behind it is to propose a pretext task that generates pseudo training labels without human annotation. Various pretext tasks have been proposed for self-supervised image representation learning, such as context-based prediction [11], rotation prediction [16], colorization [60], inpainting [40], clustering [6] and contrastive learning [38, 9, 22, 2], to name a few.

Recently, pretext tasks designed for videos are investigated to learn generic representations for downstream video tasks, such as action recognition, video retrieval, *etc*. Intuitively, a large number of studies [14, 34, 36] leveraged the distinct temporal information of videos and proposed to use frame sequence ordering as their pretext tasks. Büchler *et al.* [4] further used deep reinforcement learning to design the sampling permutations policy for order prediction tasks. Gan *et al.* [15] proposed to learn video representations by predicting the optical flow or disparity maps between frames. Although these methods demonstrate promising results, the learned representations are only based on one or two frames as they used 2D CNN for self-supervised learning. Consequently, some recent works [19, 35, 57, 28] proposed to use 3D CNNs as backbone networks for spatio-temporal representations learning, among which [28, 57, 35] extended the 2D frame ordering pretext tasks to 3D video clip ordering, and [19] proposed a pretext task to predict future frames embedding. Some concurrent works [3, 58] also investigate the speed property of video as in this work, and the reader is encouraged to review them for a broader picture. Self-supervised learning from multi-modality sources, *e.g.*, video and audio [39, 30, 1], also demonstrated promising results. In this paper, we focus on the video modality only and leave the potential extension to multi-modality as future research.

3   Our Approach

We address the video representation learning problem in a self-supervised manner. To achieve this goal, rather than training with human-annotated labels, we train a model with labels generated *automatically* from the video inputs X. The essential problem is how to design an appropriate transformation $g(\cdot)$, usually termed as pretext task, so as to yield transformed video inputs $\widetilde{X}$ with human-annotated free labels that encourage the network to learn powerful semantic spatio-temporal features for the downstream tasks, *e.g.*, action recognition.

In this work, we propose pace transformation $g_{pac}(\cdot)$ with a pace prediction task for self-supervised learning. Our idea is inspired by the *slow motion* which is widely used in film making for capturing a key moment and producing dramatic effect. Humans can easily identify it due to their sensitivity of the pace variation and a sense of normal pace. We explore whether a network could also have such
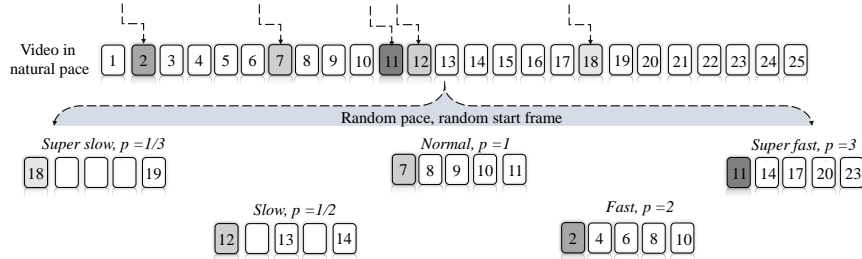
Fig. 2. Generating training samples and pace labels from the proposed pretext task. Here, we show five different sampling paces, named as *super slow*, *slow*, *normal*, *fast*, and *super fast*. The darker the initial frame is, the faster the entire clip plays.

ability to distinguish video play pace. Our assumption is that a network is not capable to perform such pace prediction task effectively unless it understands the video content and learns powerful spatio-temporal representations.

3.1 Pace Prediction

We aim to train a model with pace-varying video clips as inputs and ask the model to predict the video play paces. We assume that such a pace prediction task will encourage the neural network to learn generic transferable video representations and benefit downstream tasks. Fig. 2 shows an example of generating the training samples and pace labels. Note that in this example, we only illustrate one training video with five distinct sampling paces. Whereas in our final implementation, the sampling pace is randomly selected from several pace candidates, not restricted to these five specific sampling paces.

As shown in Fig. 2, given a video in natural pace with 25 frames, training clips will be sampled by different paces p. Typically, we consider five pace candidates {super slow, slow, normal, fast, super fast}, where the corresponding paces p are 1/3, 1/2, 1, 2, and 3, respectively. Start frame of each video clip is randomly generated and will loop over the video sample if the desired training clip is longer than the original video sample. Methods to generate each training clip with a specific p are illustrated in the following:

- *Normal motion*, where $p = 1$, training clips are sampled consecutively from the original video. The video play speed is the same as the normal pace.
- *Fast motion*, where $p > 1$, we directly sample a video frame from the original video for every p frames, *e.g.*, super fast clip with $p = 3$ contains frames 11, 14, 17, 20 and 23. As a result, when we play the clip in nature 25 fps, it looks like the video is speed up compared to the original pace.
- *Slow motion*, where $p < 1$, we put the sampled frames into the five-frames clip for every $1/p$ frames instead, *e.g.*, slow clip with $p = 1/2$, only frames 1, 3, 5 are filled with sampled frames. Regarding the blank frames, one may

consider to fill it with preceding frame, or apply interpolation algorithms [24] to estimate the intermediate frames. In practice, for simplicity, we use the preceding frame for the blank frames.

Formally, we denote the pace sampling transformation as $g(x)$. Given a video x, we apply $g(x|p)$ to obtain the training clip $\widetilde{x}$ with a training pace p. The pace prediction pretext task is formulated as a classification problem and the neural network $f(\widetilde{x})$ is trained with cross entropy loss $\mathcal{L}_{cls}$ described as:

$$\mathcal{L}_{cls} = -\sum_{i=1}^{M} y_i (\log \frac{\exp(h_i)}{\sum_{j=1}^{M} \exp(h_j)}), \quad h = f(\widetilde{x}) = f(g_{pac}(x|p)), \qquad (1)$$

where M is the number of all the pace rate candidates.

Avoid Shortcuts. As first pointed out in [11], when designing a pretext task, one must pay attention to the possibility that a network could be cheating or taking shortcuts to accomplish the pretext task by learning trivial solutions/low-level features rather than the desired high-level semantic representations. Such observations are also reported in [19, 28] for self-supervised video representation learning. In this work, we adopt color jittering augmentation to avoid such shortcuts. Empirically, we find that color jittering applied to each frame achieves much better performance than to the entire video clip. More details see Sec. 4.2.

3.2   Contrastive Learning

To further enhance the pace prediction task and regularize the learning process, we propose to leverage contrastive learning as an additional objective. Contrastive learning in a self-supervised manner has shown great potential and achieved comparable results with supervised visual representation learning recently [38, 9, 2, 22, 19, 55]. It stems from Noise-Contrastive Estimation [18] and aims to distinguish the positive samples from a group of negative samples. The fundamental problem of contrastive learning lies in the definition of *positive* and *negative* samples. For example, Chen *et al.* [22] consider the pair with different data augmentations applied to the same sample as positive, while Bachman *et al.* [2] takes different views of a shared context as positive pair. In this work, we consider two possible strategies to define positive samples: same context and same pace. In the following, we elaborate on these two strategies.

Same Context. We first consider to use clips from the same video but with different sampling paces as positive pairs, while those clips sampled from different videos as negative pair, *i.e.*, content-aware contrastive learning.

Formally, given a mini-batch of N video clips $\{x_1, \ldots, x_N\}$, for each video input x_i, we randomly sample n training clips from it by different paces, resulting in an actual training batch size $n * N$. Here, for simplicity, we consider $n = 2$, and the corresponding positive pairs are $\{(\widetilde{x}_i, p_i), (\widetilde{x}_i', p_i')\}$, where $\widetilde{x}_i$ and $\widetilde{x}_i'$

are sampled from the same video. Video clips sampled from different video are considered as negative pairs, denoted as $\{(\widetilde{x}_i, p_i), (\widetilde{x}_{\mathcal{J}}, p_{\mathcal{J}})\}$. Each video clip is then encoded into a feature vector z_i in the latent space by the neural network $f(\cdot)$. Then the positive feature vector pair is $(z_i, z_i{}')$ while the negative pairs are $\{(z_i, z_{\mathcal{J}})\}$. Denote $\mathrm{sim}(z_i, z_i{}')$ as the similarity between feature vector z_i and $z_i{}'$ and $\mathrm{sim}(z_i, z'_{\mathcal{J}})$ as the similarity between feature vector z_i and $z'_{\mathcal{J}}$, the content-aware contrastive loss is defined as:

$$\mathcal{L}_{ctr\_sc} = -\frac{1}{2N} \sum_{i,\mathcal{J}} \log \frac{\exp(\mathrm{sim}(z_i, z_i{}'))}{\sum\limits_{i} \exp(\mathrm{sim}(z_i, z_i{}')) + \sum\limits_{i,\mathcal{J}} \exp(\mathrm{sim}(z_i, z_{\mathcal{J}}))}, \qquad (2)$$

where $\mathrm{sim}(z_i, z_i{}')$ is achieved by the dot product $z_i{}^\top z_i{}'$ between the two feature vectors and so as $\mathrm{sim}(z_i, z'_{\mathcal{J}})$.

Same Pace. Concerning the proposed pace prediction pretext task, another alternative contrastive learning strategy based on same pace is explored. Specifically, we consider video clips with the same pace as positive samples regardless of the underlying video content, *i.e.*, content-agnostic contrastive learning. In this way, the contrastive learning is investigated from a different perspective that is explicitly related to pace.

Formally, given a mini-batch of N video clips $\{x_1, \ldots, x_N\}$, we first apply the pace sampling transformation $g_{pac}(\cdot)$ described above to each video input to obtain the training clips and their pace labels, denoted as $\{(\widetilde{x}_1, p_1), \ldots, (\widetilde{x}_N, p_N)\}$. Each video clip is then encoded into a feature vector z_i in the latent space by the neural network $f(\cdot)$. Consequently, (z_i, z_j) is considered as positive pair if $p_i = p_j$ while (z_i, z_k) is considered as negative pair if $p_i \neq p_k$, where $j, k \in \{1, 2, \ldots, N\}$. Denote $\mathrm{sim}(z_i, z_j)$ as the similarity between feature vector z_i and z_j and $\mathrm{sim}(z_i, z_k)$ as the similarity between feature vector z_i and z_k, the contrastive loss is defined as:

$$\mathcal{L}_{ctr\_sp} = -\frac{1}{N} \sum_{i,j,k} \log \frac{\exp(\mathrm{sim}(z_i, z_j))}{\sum\limits_{i,j} \exp(\mathrm{sim}(z_i, z_j)) + \sum\limits_{i,k} \exp(\mathrm{sim}(z_i, z_k))}, \qquad (3)$$

where $\mathrm{sim}(z_i, z_j)$ is achieved by the dot product $z_i{}^\top z_j$ between the two feature vectors and so as $\mathrm{sim}(z_i, z_k)$.

3.3　Network Architecture and Training

The framework of the proposed pace prediction approach is illustrated in Fig. 3. Given a set of unlabeled videos, we firstly sample various video clips with different paces. Then these training clips are fed into a deep model (3D CNN here) for spatio-temporal representation learning. The final objective is to optimize the model to predict the pace of each video clip and maximize the agreement (mutual information) between positive pairs at the same time.
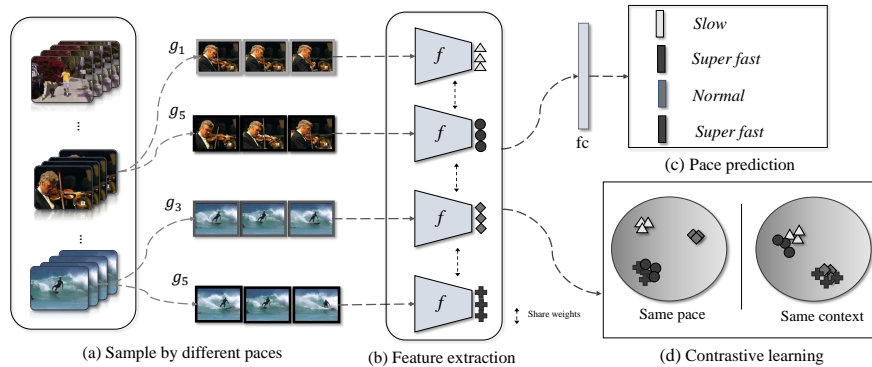
Fig. 3. Framework of the proposed approach. (a) Training clips are sampled by different paces. Here, g_1, g_3, g_5 illustrates examples of *slow*, *normal* and *super fast* pace. (b) A 3D CNN f is leveraged to extract spatio-temporal features. (c) The model is trained to predict the specific pace applied to each video clip. (d) Two possible contrastive learning strategies are considered to regularize the learning process at the latent space. The symbols at the end of the CNNs represent feature vectors extracted from different video clips, where the intensity represents different video pace.

In terms of the network architecture, we mainly consider three backbone networks, *i.e.* C3D [46], 3D-ResNet [20] and R(2+1)D [47], to study the effectiveness of the proposed approach. C3D is a classic neural network which operates on 3D video volume by extending 2D convolutional kernel to 3D. 3D-ResNet (R3D) [20] is an extension of the ResNet [21] architecture to videos. In this work, we use 3D-ResNet18 (R3D-18). R(2+1)D [47] is proposed to break the original spatio-temporal 3D convolution into a 2D spatial convolution and a 1D temporal convolution. Apart from these three networks, we also use a state-of-the-art model S3D-G [56] to further exploit the potential of the proposed approach.

By jointly optimizing the classification objective (Eq. 1) and the contrastive objective (Eq. 2 or 3), the final training loss is defined as:

$$\mathcal{L} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{ctr}\mathcal{L}_{ctr}, \tag{4}$$

where λ_{cls}, λ_{ctr} are weighting parameters to balance the optimization of pace prediction and contrastive learning, respectively. The $\mathcal{L}_{ctr}$ refers to either contrastive learning with same pace $\mathcal{L}_{ctr\_sp}$ or with same context $\mathcal{L}_{ctr\_sc}$.

4 Experiments

4.1 Datasets and Implementation Details

Datasets. We use three datasets as follows:

UCF101 [44] is a widely used dataset in action recognition task, consisting of 13,320 video samples with 101 action classes. The dataset is divided into three

training/testing splits and in this paper, following prior works [57, 3], we use training split 1 as pre-training dataset and training/testing split 1 for evaluation.

Kinetics-400 (K-400) [27] is a large action recognition dataset, which consists of 400 human action classes and around 306k videos. It is divided into three splits: training/validation/testing. In this work, we use the training split (around 240k video samples) as the pre-training dataset, to validate the proposed approach.

HMDB51 [31] is a relatively small action recognition dataset, which contains around 7,000 videos with 51 action classes. It is divided into three training/testing splits and following prior works [57, 3], we use training/testing split 1 for downstream task evaluation.

Self-supervised Pre-training Stage. When pre-training on the K-400 dataset, for each video input, we first generate a frame index randomly and then start from the index, sample a consecutive 16-frame video clip. While when pre-training on UCF101 dataset, as it only contains around 9k videos in the training split, we set epoch size to be around 90k for temporal jittering following [1]. As for data augmentation, we randomly crop the video clip to 112×112 and flip the whole video clip horizontally. The batch size is 30 and SGD is used as optimizer with an initial learning rate 1×10^{-3}. The leaning rate is divided by 10 for every 6 epochs and the training process is stopped after 18 epochs. When jointly optimizing $\mathcal{L}_{cls}$ and $\mathcal{L}_{ctr}$, λ_{cls} is set to 1 and λ_{ctr} is set to 0.1.

Supervised Fine-tuning Stage. Regarding the action recognition task, during the fine-tuning stage, weights of convolutional layers are retained from the self-supervised learning networks while weights of fully-connected layers are randomly initialized. The whole network is then trained with cross-entropy loss. Image pre-processing method and training strategy are the same as the self-supervised pre-training stage, except that the initial learning rate is set to 0.01 for S3D-G and 3×10^{-3} for the others.

Evaluation. During inference, following the evaluation protocol in [57, 35], we sample 10 clips uniformly from each video in the testing set of UCF101 and HMDB51. For each clip, center crop is applied. The predicted label of each video is generated by averaging the softmax probabilities of all clips in the video.

4.2 Ablation Studies

In this section, we firstly explore the best sampling pace design for the pace prediction task. We apply it to three different backbone networks to study the effectiveness of the pretext task and the network architectures. Experimental results show that respectable performance can be achieved by only using the pace prediction task. When introducing the contrastive learning, the performance is further boosted, and the same content configuration performs much better than the same pace one. More details are illustrated in the following.

Table 1. Explore the best setting for pace prediction task. Sampling pace $p = [a, b]$ represents that the lowest value of pace p is a and the highest is b with an interval of 1, except $p = [\frac{1}{3}, 3]$, where p is selected from $\{\frac{1}{3}, \frac{1}{2}, 1, 2, 3\}$.

Color jittering	Method	#Classes	UCF101
$\times$	Random	-	56.0
$\times$	$p = [1, 3]$	3	71.4
$\times$	$p = [1, 4]$	4	**72.0**
$\times$	$p = [1, 5]$	5	72.0
$\times$	$p = [1, 6]$	6	71.1
$\checkmark$	$p = [1, 4]$	4	**73.9**
$\checkmark$	$p = [\frac{1}{3}, 3]$	5	73.9
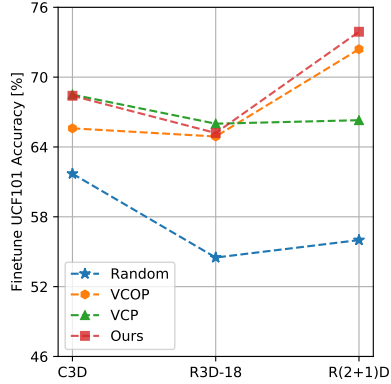
Fig. 4. Action recognition accuracy on three backbone architectures (horizontal axis) using four initialization methods.

Sampling Pace. We investigate the best setting for pace prediction task with R(2+1)D backbone network [47] in Table 1. To study the relationship between the complexity of the pretext task and the effectiveness on the downstream task, we first use a *relative* pace design with only normal and fast motion. For example, regarding $p = [1, 3]$, we assume clips with $p = 2$ to be the anchor, *i.e.*, normal pace and therefore, clips with $p = 3$ are with faster pace and $p = 1$ are with slower pace. It can be seen from the table that with the increase of the maximum pace, namely the number of training classes, the accuracy of the downstream action recognition task keeps increase, until $p = [1, 4]$. When the sampling pace increases to $p = [1, 6]$, the accuracy starts to drop. We believe that this is because such a pace prediction task is becoming too difficult for the network to learn useful semantic features. This provides an insight on the pretext task design that a pretext task should not be too simple nor too ambiguous to solve, in consistent with the observations found in [37, 14].

We further validate the effectiveness of color jittering based on the best sampling pace design $p = [1, 4]$. It can be seen from Table 1 that with color jittering, the performance is further improved by 1.9%. It is also interesting to note that the *relative* pace, *i.e.*, $p = [1, 4]$, achieves comparable result with the *absolute* pace, *i.e.*, $p = [\frac{1}{3}, 3]$, but with less number of classes. In the following experiments, we use sampling pace $p = [1, 4]$ along with color jittering by default.

Backbone Network. We validate the proposed pace prediction task without contrastive learning using three backbone networks. Some recent works [57, 35] validate their proposed self-supervised learning approaches on modern spatio-temporal representation learning networks, such as R3D-18 [20, 47], R(2+1)D [47], *etc.* This practice could influence the direct evaluation of the pretext tasks, as

Table 2. Evaluation of different contrastive learning configurations on both UCF101 and HMDB51 datasets. *Note that paramters when adding an fc layer only increase ∼4k, which is negligible compared to the original 14.4M parameters.

Experimental setup					Downstream tasks	
Pace Pred.	Ctr. Learn.	Network	Configuration	Params	UCF101	HMDB51
✓	×	R(2+1)D	-	14.4M	**73.9**	**33.8**
×	✓	R(2+1)D	Same pace	14.4M	59.4	20.3
×	✓	R(2+1)D	Same context	14.4M	67.3	28.6
✓	✓	R(2+1)D	Same pace	14.4M	73.6	32.3
✓	✓	R(2+1)D	Same context	14.4M	75.8	35.0
✓	✓	R(2+1)D + fc	Same context	14.4M*	**75.9**	**35.9**

the performance improvement can also come from the usage of more powerful networks. Therefore, we study the effectiveness of the pace prediction task and compare with some recent works on three backbone networks, as shown in Fig. 4. For a fair comparison, following [57, 35], we use the first training split of UCF101 as the pre-training dataset and evaluate on training/testing split 1.

Some key observations are listed in the following: (1) The proposed approach achieves significant improvement over the random initialization across all three backbone networks. (2) Although in the random initialization setting, C3D achieves the best results, R(2+1)D and R3D-18 benefit more from the self-supervised pre-training and R(2+1)D finally achieves the best performance. (3) Without contrastive learning, the proposed pace prediction task already demonstrates impressive effectiveness to learn video representations, achieving comparable performance with current state-of-the-art methods VCP [35] and VCOP[57] on C3D and R3D-18 and outperforms them when using R(2+1)D.

Contrastive Learning. The performances of the two contrastive learning configurations are shown in Table 2. Some key observations are listed for a better understanding of contrastive learning: (1) The same pace configuration achieves much worse results than the same context configuration. We suspect the reason is that in the same pace configuration, as there are only four pace candidates $p = [1, 4]$, video clips are tend to belong to the same pace. Therefore, compared with the same context configuration, much fewer negative samples are presented in the training batches, withholding the effectiveness of contrastive learning. (2) Pace prediction task achieves much better performance compared to each of the two contrastive learning settings. This demonstrates the superiority of the proposed pace prediction task.

When combining the pace prediction task with contrastive learning, similar to the observation described above, regarding the same pace configuration, performance is slightly deteriorated and regarding the same context configuration, performance is further improved both on UCF101 and HMDB51 datasets. It shows that appropriate multi-task self-supervised learning can further boost

Table 3. Comparison with the state-of-the-art self-supervised learning methods on UCF101 and HMDB51 dataset (Pre-trained on video modality only).*The input video clips contain 64 frames.

Method	Pre-training settings				Evaluation	
	Network	Input size	Params	Dataset	UCF101	HMDB51
Fully supervised	S3D-G	$224 \times 224^*$	9.6M	ImageNet	86.6	57.7
Fully supervised	S3D-G	$224 \times 224^*$	9.6M	K-400	96.8	74.5
Object Patch[53]	AlexNet	227×227	62.4M	UCF101	42.7	15.6
ClipOrder[36]	CaffeNet	227×227	58.3M	UCF101	50.9	19.8
Deep RL[4]	CaffeNet	227×227	-	UCF101	58.6	25.0
OPN [34]	VGG	80×80	8.6M	UCF101	59.8	23.8
VCP [35]	R(2+1)D	112×112	14.4M	UCF101	66.3	32.2
VCOP [57]	R(2+1)D	112×112	14.4M	UCF101	72.4	30.9
PRP [58]	R(2+1)D	112×112	14.4M	UCF101	72.1	35.0
Ours	R(2+1)D	112×112	14.4M	UCF101	**75.9**	**35.9**
MAS[51]	C3D	112×112	27.7M	K-400	61.2	33.4
RotNet3D [25]	R3D-18	224×224	33.6M	K-400	62.9	33.7
ST-puzzle [28]	R3D-18	224×224	33.6M	K-400	65.8	33.7
DPC [19]	R3D-18	128×128	14.2M	K-400	68.2	34.5
DPC [19]	R3D-34	224×224	32.6M	K-400	75.7	35.7
Ours	R(2+1)D	112×112	14.4M	K-400	**77.1**	**36.6**
SpeedNet [3]	S3D-G	$224 \times 224^*$	9.6M	K-400	81.1	48.8
Ours	S3D-G	$224 \times 224^*$	9.6M	UCF101	**87.1**	**52.6**

the performances, in consistent with the observation in [12]. Based on the same video content configuration, we further introduce a nonlinear layer between the embedding space and the final contrastive learning space to alleviate the direct influence on the pace prediction learning. It is shown that such a practice can further improve the performance (last row in Table 2).

4.3   Action Recognition

We compare our approach with other methods on the action recognition task in Table 3.We have the following key observations: (1) Our method achieve the state-of-the-art results on both UCF101 and HMDB51 dataset. When pre-trained on UCF101, we outperform the current best-performing method PRP [58]. When pre-trained on K-400, we outperform the current best-performing method DPC [19]. (2) Note that here the DPC method uses R3D-34 as their backbone network and the video input size is 224×224 while we only use 112×112. When the input size of DPC is at the same scale as ours, *i.e.*, 128×128, we outperform it by 8.9% on UCF101 dataset. We attribute such success to both our pace predic-tion task and the usage of R(2+1)D. It can be observed that with R(2+1)D and only UCF101 as pre-train dataset, VCOP [57] can achieve 72.4% on UCF101 and
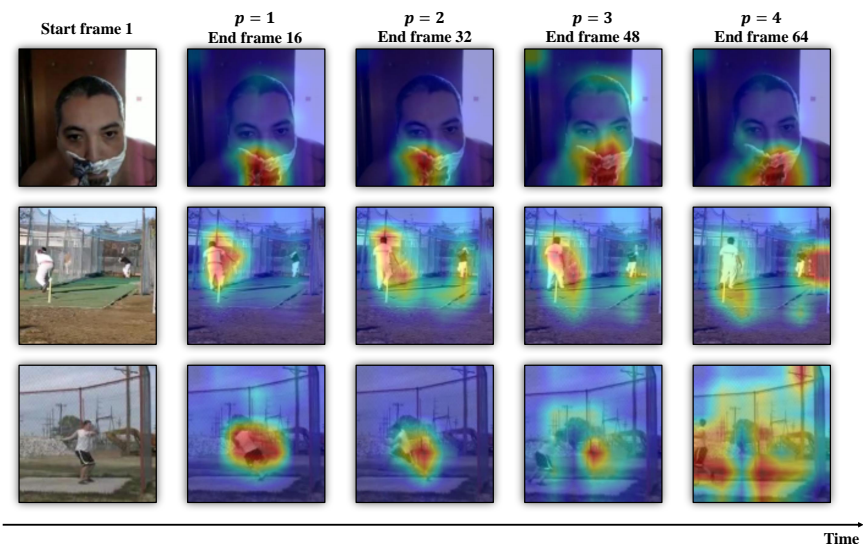
Fig. 5. Attention visualization of the conv5 layer from self-supervised pre-trained model using [59]. Attention map is generated with 16-frames clip inputs and applied to the last frame in the video clips. Each row represents a video sample while each column illustrates the end frame *w.r.t.* different sampling pace *p*.

30.9% on HMDB51. (3) Backbone networks, input size and clip length do play important roles in the self-supervised video representation learning. As shown in the last row, by using the S3D-G [56] architecture with 64-frame clips as inputs, pre-training only on UCF101 can already achieve remarkable performance, even superior to fully supervised pre-training on ImageNet (on UCF101).

   To further validate the proposed approach, we visualize the attention maps based on the pre-trained R(2+1)D model, as shown in Fig. 5. It can be seen from the attention maps that the neural network will pay more attention to the motion areas when learning the pace prediction task. It is also interesting to note that in the last row, as attention map on $p = 4$ computes the layer information spanning 64 frames, it is activated at several motion locations.

4.4   Video Retrieval

We further validate the proposed approach on the video retrieval task. Basically, we follow the same evaluation protocol described in [35, 57]. Ten 16-frames clips are sampled from each video and then go through a feed-forward pass to generate features from the last pooling layer (p5). For each clip in the testing split, the Top$k$ nearest neighbors are queried from the training split by computing the cosine distances between every two feature vectors. We consider k to be 1, 5, 10, 20, 50. To align the experimental results with prior works [35, 57] for fair comparison, we use pre-trained models from the pace prediction task on

Table 4. Comparison with state-of-the-art methods for nearest neighbour retrieval task on UCF101 dataset.

	Method	Top1	Top5	Top10	Top20	Top50
AlexNet	Jigsaw[37]	19.7	28.5	33.5	40.0	49.4
	OPN[34]	19.9	28.7	34.0	40.6	51.6
	Deep RL[4]	_25.7_	36.2	42.2	49.2	59.5
C3D	Random	16.7	27.5	33.7	41.4	53.0
	VCOP[57]	12.5	29.0	39.0	50.6	66.9
	VCP[35]	17.3	31.5	42.0	52.6	67.7
	Ours (p5)	20.0	37.4	46.9	58.5	73.1
	Ours(p4)	**31.9**	**49.7**	**59.2**	**68.9**	**80.2**
R3D-18	Random	9.9	18.9	26.0	35.5	51.9
	VCOP[57]	14.1	30.3	40.4	51.1	66.5
	VCP[35]	18.6	33.6	42.5	53.5	68.1
	Ours (p5)	19.9	36.2	46.1	55.6	69.2
	Ours(p4)	**23.8**	**38.1**	**46.4**	**56.6**	**69.8**
R(2+1)D	Random	10.6	20.7	27.4	37.4	53.1
	VCOP[57]	10.7	25.9	35.4	47.3	63.9
	VCP[35]	19.9	33.7	42.0	50.5	64.4
	Ours (p5)	17.9	34.3	44.6	55.5	72.0
	Ours(p4)	**25.6**	**42.7**	**51.3**	**61.3**	**74.0**

Table 5. Comparison with state-of-the-art methods for nearest neighbor retrieval task on HMDB51 dataset.

	Method	Top1	Top5	Top10	Top20	Top50
C3D	Random	7.4	20.5	31.9	44.5	66.3
	VCOP[57]	7.4	22.6	34.4	48.5	70.1
	VCP[35]	7.8	23.8	35.5	49.3	71.6
	Ours (p5)	8.0	25.2	37.8	54.4	77.5
	Ours(p4)	**12.5**	**32.2**	**45.4**	**61.0**	**80.7**
R3D-18	Random	6.7	18.3	28.3	43.1	67.9
	VCOP[57]	7.6	22.9	34.4	48.8	68.9
	VCP[35]	7.6	24.4	36.6	53.6	76.4
	Ours (p5)	8.2	24.2	37.3	53.3	74.5
	Ours(p4)	**9.6**	**26.9**	**41.1**	**56.1**	**76.5**
R(2+1)D	Random	4.5	14.8	23.4	38.9	63.0
	VCOP[57]	5.7	19.5	30.7	45.8	67.0
	VCP[35]	6.7	21.3	32.7	49.2	73.3
	Ours (p5)	10.1	24.6	37.6	54.4	77.1
	Ours(p4)	**12.9**	**31.6**	**43.2**	**58.0**	**77.1**

UCF101 dataset. As shown in Table 4 and Table 5, our method outperforms the VCOP [57] and VCP [35] in most cases on the two datasets across the three backbone networks. In practice, we also find that significant improvement can be achieved by using the second last pooling layer (p4).

5   Conclusion

In this paper, we proposed a new perspective towards self-supervised video representation learning, by pace prediction. Contrastive learning was also incorporated to further encourage the networks to learn high-level semantic features. To validate our approach, we conducted extensive experiments across several network architectures on two different downstream tasks. The experimental results demonstrated the superiority of our method on learning powerful spatio-temporal representations. Besides, the pace prediction task does not rely on any motion channel as prior information/input during training. As a result, such a pace prediction task can serve as a simple yet effective supervisory signal when applying the self-supervised video representation learning in real world, taking advantage of billions of video data freely.

References

1. Alwassel, H., Mahajan, D., Torresani, L., Ghanem, B., Tran, D.: Self-supervised learning by cross-modal audio-video clustering. arXiv preprint arXiv:1911.12667 (2019)
2. Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning representations by maximizing mutual information across views. In: NeurIPS (2019)
3. Benaim, S., Ephrat, A., Lang, O., Mosseri, I., Freeman, W.T., Rubinstein, M., Irani, M., Dekel, T.: Speednet: Learning the speediness in videos. In: CVPR (2020)
4. Buchler, U., Brattoli, B., Ommer, B.: Improving spatiotemporal self-supervision by deep reinforcement learning. In: ECCV) (2018)
5. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: Activitynet: A large-scale video benchmark for human activity understanding. In: CVPR (2015)
6. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: ECCV (2018)
7. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR (2017)
8. Chao, Y.W., Vijayanarasimhan, S., Seybold, B., Ross, D.A., Deng, J., Sukthankar, R.: Rethinking the faster r-cnn architecture for temporal action localization. In: CVPR (2018)
9. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709 (2020)
10. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: ECCV (2006)
11. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: ICCV (2015)
12. Doersch, C., Zisserman, A.: Multi-task self-supervised visual learning. In: ICCV (2017)
13. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: ICCV (2019)
14. Fernando, B., Bilen, H., Gavves, E., Gould, S.: Self-supervised video representation learning with odd-one-out networks. In: CVPR (2017)
15. Gan, C., Gong, B., Liu, K., Su, H., Guibas, L.J.: Geometry guided convolutional neural networks for self-supervised video representation learning. In: CVPR (2018)
16. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: ICLR (2018)
17. Giese, M.A., Poggio, T.: Neural mechanisms for the recognition of biological movements. Nature Reviews Neuroscience **4**(3), 179–192 (2003)
18. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: AISTATS (2010)
19. Han, T., Xie, W., Zisserman, A.: Video representation learning by dense predictive coding. In: ICCV Workshops (2019)
20. Hara, K., Kataoka, H., Satoh, Y.: Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In: CVPR (2018)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
22. Hénaff, O.J., Razavi, A., Doersch, C., Eslami, S., Oord, A.v.d.: Data-efficient image recognition with contrastive predictive coding. arXiv preprint arXiv:1905.09272 (2019)

23. Hussein, N., Gavves, E., Smeulders, A.W.: Timeception for complex action recognition. In: CVPR (2019)
24. Jiang, H., Sun, D., Jampani, V., Yang, M.H., Learned-Miller, E., Kautz, J.: Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In: CVPR (2018)
25. Jing, L., Yang, X., Liu, J., Tian, Y.: Self-supervised spatiotemporal feature learning via video rotation prediction. arXiv preprint arXiv:1811.11387 (2018)
26. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR (2014)
27. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
28. Kim, D., Cho, D., Kweon, I.S.: Self-supervised video representation learning with space-time cubic puzzles. In: AAAI (2019)
29. Klaser, A., Marszałek, M., Schmid, C.: A spatio-temporal descriptor based on 3d-gradients. In: BMVC (2008)
30. Korbar, B., Tran, D., Torresani, L.: Cooperative learning of audio and video models from self-supervised synchronization. In: NeurIPS (2018)
31. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: ICCV (2011)
32. Laptev, I.: On space-time interest points. IJCV **64**(2-3), 107–123 (2005)
33. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
34. Lee, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Unsupervised representation learning by sorting sequences. In: ICCV (2017)
35. Luo, D., Liu, C., Zhou, Y., Yang, D., Ma, C., Ye, Q., Wang, W.: Video cloze procedure for self-supervised spatio-temporal learning. arXiv preprint arXiv:2001.00294 (2020)
36. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and learn: unsupervised learning using temporal order verification. In: ECCV (2016)
37. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV (2016)
38. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
39. Owens, A., Efros, A.A.: Audio-visual scene analysis with self-supervised multisensory features. In: ECCV (2018)
40. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: CVPR (2016)
41. Shou, Z., Chan, J., Zareian, A., Miyazawa, K., Chang, S.F.: Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In: CVPR (2017)
42. Shou, Z., Wang, D., Chang, S.F.: Temporal action localization in untrimmed videos via multi-stage cnns. In: CVPR (2016)
43. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NeruIPS (2014)
44. 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)
45. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using lstms. In: ICML (2015)
46. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: ICCV (2015)

47. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: CVPR (2018)
48. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics. In: NeurIPS (2016)
49. Wang, B., Ma, L., Zhang, W., Liu, W.: Reconstruction network for video captioning. In: CVPR (2018)
50. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: ICCV (2013)
51. 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: CVPR (2019)
52. Wang, J., Jiang, W., Ma, L., Liu, W., Xu, Y.: Bidirectional attentive fusion with context gating for dense video captioning. In: CVPR (2018)
53. Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: ICCV (2015)
54. Watamaniuk, S.N., Duchon, A.: The human visual system averages speed information. Vision research **32**(5), 931–941 (1992)
55. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: CVPR (2018)
56. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: ECCV (2018)
57. Xu, D., Xiao, J., Zhao, Z., Shao, J., Xie, D., Zhuang, Y.: Self-supervised spatiotemporal learning via video clip order prediction. In: CVPR (2019)
58. Yao, Y., Liu, C., Luo, D., Zhou, Y., Ye, Q.: Video playback rate perception for self-supervised spatio-temporal representation learning. In: CVPR (2020)
59. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In: ICLR (2017)
60. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV (2016)