Self-Supervised Video Representation Using Pretext-Contrastive Learning

Li Tao, Xueting Wang, Toshihiko Yamasaki
The University of Tokyo
{taoli, xt_wang, yamasaki}@hal.t.u-tokyo.ac.jp

Abstract

Pretext tasks and contrastive learning have been successful in self-supervised learning for video retrieval and recognition. In this study, we analyze their optimization targets and utilize the hyper-sphere feature space to explore the connections between them, indicating the compatibility and consistency of these two different learning methods. Based on the analysis, we propose a self-supervised training method, referred as Pretext-Contrastive Learning (PCL), to learn video representations. Extensive experiments based on different combinations of pretext task baselines and contrastive losses confirm the strong agreement with their self-supervised learning targets, demonstrating the effectiveness and the generality of PCL. The combination of pretext tasks and contrastive losses showed significant improvements in both video retrieval and recognition over the corresponding baselines. And we can also outperform current state-of-the-art methods in the same manner. Further, our PCL is flexible and can be applied to almost all existing pretext task methods.

Introduction

With the development of convolutional neural networks (CNNs) and the help of many large labeled datasets, the computer vision community has witnessed unprecedented success in many tasks such as object classification, detection, segmentation, and action recognition. For both image level and video level tasks, pre-training on larger datasets such as ImageNet [1] and Kinetics [2] is important to ensure satisfactory performance.

However, the world is abundant in images and videos, and annotating large-scale datasets requires a wealth of resources. In particular, action recognition task generally requires properly trimmed action video clips to avoid unnecessary noise to ensure the performance, which makes the situation more serious.

To leverage unlabeled data, many self-supervised learning methods have been proposed for video representation and applied to tasks such as video recognition. These methods can be broadly divided into two categories, pretext task-based methods and contrastive learning methods.

Several tasks have been designed to constrain pretext task-based models to learn effective and informative representations. These tasks include solving jigsaw puzzles, image inpainting, and detecting image rotation angles. For video data, some of these spatial tasks are also effective, together with temporal-related tasks such as predicting frame orders or video clip orders, recognizing temporal transformations, and being sensitive to video playback speed. A suitable combination of such different tasks can help improve the performances of the methods in video retrieval and recognition tasks. However, even though high accuracy can be achieved on video retrieval and recognition tasks, it seems to be endless because there can be new and better pretext tasks. Furthermore, identifying which pretext task is more effective and why is theoretically difficult to explain.

In contrastive learning methods, the solution is based on the comparison among different samples. The key idea is to distinguish one instance from another. Usually, different modalities and different spatial/temporal crops of the same video are treated as positives while samples from different videos are treated as negatives, even though they may belong to the same action category. However, once the network can distinguish one instance from another, the learned features would be sufficient for downstream tasks.

The combination of different pretext tasks and contrastive learning seems to be better than each on its own, and such kinds of combinations have succeed in video representation learning. However, the reason why the combination can be effective and reasonable remains undetermined.

In this paper, we formulate the optimization targets of pretext tasks and contrastive learning and use the hyper-sphere feature space to analyze feature representations learned from them. As a results, we find evidence of feature consistency between them. As trained models share similar distributions in the feature space, it is reasonable to employ them simultaneously. We then propose the pretext-contrastive learning (PCL), which can facilitate the advantages of these two technologies. We should clarify that we are not proposing a new pretext task here; instead, we want to bridge the gap between pretext tasks and contrastive learning beyond simple intuition or seeking higher accuracy.

To prove the effectiveness of our PCL, three pretext task-based methods are set as the baselines. Different network backbones are tested to eliminate biases. Extensive experimental results prove the effectiveness of our proposal. The proposed PCL is closer to a framework or a strategy rather than a simple method as it is flexible and can be applied to many existing solutions.

The contributions of this work can be summarized as:

- We analyze optimization targets of pretext tasks and contrastive learning, and utilize the hyper-sphere feature space to find the consistencies between them.
- Based on our analysis, PCL is proposed to combine pretext tasks with contrastive learning methods to learn better video representations.
- Experiments demonstrate that improvements can be obtained over several pretext task baselines, and we can also achieve state-of-the-art performances in two evaluation tasks on two benchmark datasets.
- Our proposal is flexible enough to be applied to any existing pretext tasks.

Related Work

In this section, we divide the existing self-supervised learning methods into two categories according to their optimization targets: pretext tasks and contrastive learning.

Pretext Tasks

Self-supervised learning methods were first proposed for images. Spatial pretext tasks include solving jigsaw problems [3], detecting image rotations [4], image channel prediction [5], and image inpainting [6]. Prior works also include image reconstruction using autoencoders [7] and variational autoencoders [8].

For video data, some image-based pretext tasks can be directly applied or extended, such as detecting rotation angles [9] and completing space-time cubic puzzles [10]. Compared to image data, videos have an additional temporal dimension. Therefore, to utilize temporal information, many works have designed temporal-specific tasks. In [11], the network was trained to distinguish whether the input frames were in the correct order. [12] trained their odd-one-out network (O3N) to identify unrelated or odd video clips. The order prediction network (OPN) [13] was trained by predicting the correct order of shuffled frames. The video clip order prediction network [14] used video clips together with a spatio-temporal CNN during training. Further, [15] utilized spatial and temporal transformations to train the network. Many recent works have started to utilize the playback speed of the input video clips. SpeedNet [16] was trained to detect whether a video is playing at a normal rate or at a sped-up rate. [17] trained a network to sort video clips according to the corresponding playback rates. The playback rate perception (PRP) [18] used an additional reconstructing decoder branch to help train the model. [19] and [20] also utilized additional transformations, such as random permutation and frame interpolation, to help train the model.

All these pretext tasks can be set as the main branch and can be combined with our PCL for better performance.

Contrastive Learning

The success of contrastive learning also originated from image tasks. The key idea of contrastive learning is to minimize the distance within positive pairs in the feature space while maximizing the distance between negative pairs. After contrastive loss was proposed [21], contrastive learning has become the mainstream method for self-supervised learning of image data. Contrastive predictive coding (CPC) [22] attempted to learn the future from the past by using sequential data. Deep InfoMax [23] and Instance Discrimination [24] were proposed to maximize information probability from the same sample. Contrastive multiview coding (CMC) [25] used different views (e.g. different color spaces) from the same sample. MoCo [26, 27] used a momentum-updated encoder to conduct contrastive learning. In SimCLR [28], different combinations of data augmentation methods were tested for paired samples. BYOL [29] trained the network without negative samples.

The above-mentioned methods mainly focus on image data. Some technologies have been successfully applied to video data. The concept of CMC can be easily adapted to videos by simply using video data as the model input. Similar to CPC, DPC [30] was proposed to handle video data. IIC [31] introduced intra-negative video samples to enhance contrastive learning. These methods are all based on visual data only. The contrastive learning concept can be extended to additional modalities of video, such as audio [32, 33], text and descriptive data [34].

Most of these contrastive learning methods utilize noise cross-entropy (NCE) loss [35] for robust and effective training. [36] explored the learned features in the hyper-sphere feature space and proposed a new loss function, align-uniform loss, which is a possible substitute for NCE loss.

Methods Combinations

A combination of several pretext tasks with proper weights can yield better performances [37] than when they are used alone. In fact, many methods are beyond one simple pretext task, and are already a combination of some particular pretext tasks. We have listed many pretext tasks, and the potential combinations are extensive. These pretext tasks vary widely, and determining why one pretext task or one combination is better than another is difficult.

The combination of pretext tasks and contrastive learning has also been attempted in [38, 39]. However, except for the reported results, few analyses have been conducted on why and how it is effective when combining different methods. In this paper, we address this issue and present the evidence in the feature spaces. Improvements over three pretext task baselines will also prove the effectiveness of our proposal.

Method

We start by formulating existing methods, namely pretext tasks and contrastive learning methods. Based on the analysis of these formulations, an assumption is inspired : these methods learn similar feature representations. It may seem axiomatic, but the learning mechanisms are in fact different. Feature analysis in the hyper-sphere shows the evidence to

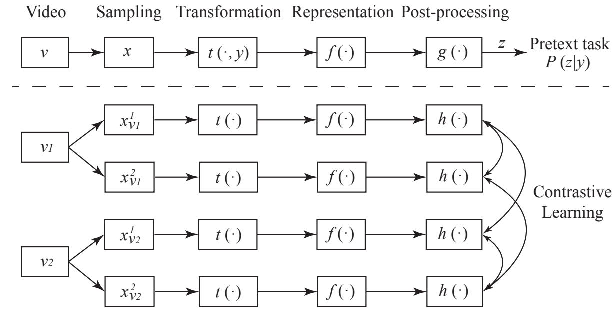


Figure 1: Optimization targets.

support this assumption, making it possible and reasonable to combine different methods together. Our joint optimization framework will be introduced at the end of this section, demonstrating the usage and the flexibility of our PCL.

Formulations

The goal for self-supervised video representation learning is to learn effective feature representations from videos using a backbone network f_θ. The commonly used networks are based on spatio-temporal convolutions, where the input video v_i is decoded to a sequence of frames and several frames are stacked to form video clips x_{v_i}. Video features can be generated by using $f_\theta(x)$.

Pretext task For pretext task-based methods, one or several tasks are used to train the network in a supervised manner. Most pretext tasks are classification tasks. For example, VCP [15] used different transformations on the input video clip x and trained the network by distinguishing which transformation was conducted. 3DRotNet [9] was trained by detecting the rotation angles of the input clip. VCOP [14] shuffled video clips and trained the network by predicting the correct order class of the inputs. All these pretext tasks can be concluded as designing a proper classification task. The video clip x needs to be transformed by a specific transformation function $t(x, y)$, where y is the label of the corresponding transformation. Then the optimization target of these pretext tasks becomes

$$\underset{\forall v_i}{\text{minimize}}\ \mathcal{L}_{cls}(g(f_\theta(t(x_{v_i}, y)), y), \tag{1}$$

where $g(\cdot)$ is the post-process network to process extracted features and $\mathcal{L}_{cls}$ is usually set as cross-entropy loss.

Contrastive learning For contrastive learning methods, after extracting features from the backbone, a two-linear-layer multi-layer perceptron (MLP) is usually used to project features $f_\theta(x)$ to another feature space. Let us denote the projector network as $h(\cdot)$. Positive pairs and negative pairs are required to constrain the network. $x^1_{v_i}$ is one video clip from the video v_i, and when another video clip $x^2_{v_i}$ is from the same video, these two video clips are treated as positive pairs. Conversely, when a video clip x_{v_j} is from a different video, v_j, Then x_{v_i} and x_{v_j} are negative pairs. The encoded feature in the projection feature space is $h(f(x_{v_i}))$, which is denoted as z_{v_i} instead for simplicity. Define $D(z_{v_i}, z_{v_j})$ as the similarity distance between feature z_{v_i} and z_{v_j}; then for video v_i, the contrastive learning target is

$$\text{minimize}\ \mathcal{L}^{v_i}_{NCE} = \mathcal{L}^{v_i^1}_{NCE} + \mathcal{L}^{v_i^2}_{NCE}, \tag{2}$$

where

$$\begin{aligned} \mathcal{L}^{v_i^1}_{NCE} &= -\log \frac{D(z^1_{v_i}, z^2_{v_i})}{D(z^1_{v_i}, z^2_{v_i}) + \sum_{j\neq i} D(z^1_{v_i}, z^1_{v_j})}, \\ \mathcal{L}^{v_i^2}_{NCE} &= -\log \frac{D(z^1_{v_i}, z^2_{v_i})}{D(z^1_{v_i}, z^2_{v_i}) + \sum_{j\neq i} D(z^2_{v_i}, z^2_{v_j})}. \end{aligned} \tag{3}$$

This type of contrastive loss is widely used together with a memory bank or a queue to augment negative samples. In [36], a new form of loss function was proposed to conduct contrastive learning, aiming to align features from the same video while constraining features from different videos to distribute uniformly. The optimization target of this type of contrastive learning becomes

$$\text{minimize}\ \mathcal{L}_{au} = \mathcal{L}_{align} + \mathcal{L}_{uniform}, \tag{4}$$

where

$$\begin{aligned} \mathcal{L}_{align} &= \mathbb{E}_{(v_i^1, v_i^2)\sim v_i} \|z^1_{v_i}, z^2_{v_i}\|^2_2, \\ \mathcal{L}_{uniform} &= \mathbb{E}_{(v_i, v_j)\sim v_{batch}} e^{-t\|z_{v_i} - z_{v_j}\|^2_2}. \end{aligned} \tag{5}$$

Differences and connections Although both kinds of methods try to learn efficient representations from videos, by comparing the optimization targets (Eq. 1 for pretext tasks and Eq. 2 or Eq. 4 for contrastive learning), we find that the optimization target of pretext tasks focuses on the intra-information within a sample itself, while contrastive learning focuses on the inter-information between samples (Fig. 1).

As pretext tasks treat all samples equally, after training, the learned features of videos from the entire datasets may tend to distribute uniformly in the feature space. As contrastive learning attempts to minimize the feature distance between video clips of the same video and maximize it if they are from different videos, the distribution of the learned features should also tend to distribute uniformly in the feature space. Thus, even though the optimization targets are different, we can still have the following assumption.

Assumption. Pretext task-based methods and contrastive learning methods aim to learn similar feature representations during training.

If this assumption is true, it is possible and reasonable to make use of both to train a network in a joint optimization framework. Next, we will show the evidence in the feature space to support this assumption.

Feature Distribution on the Hyper-sphere

For the feature distribution of contrastive learning, the hyper-sphere feature space was used in [36] to show that the contrastive learning loss function can be separated into two parts, alignment and uniformity. We follow this work and

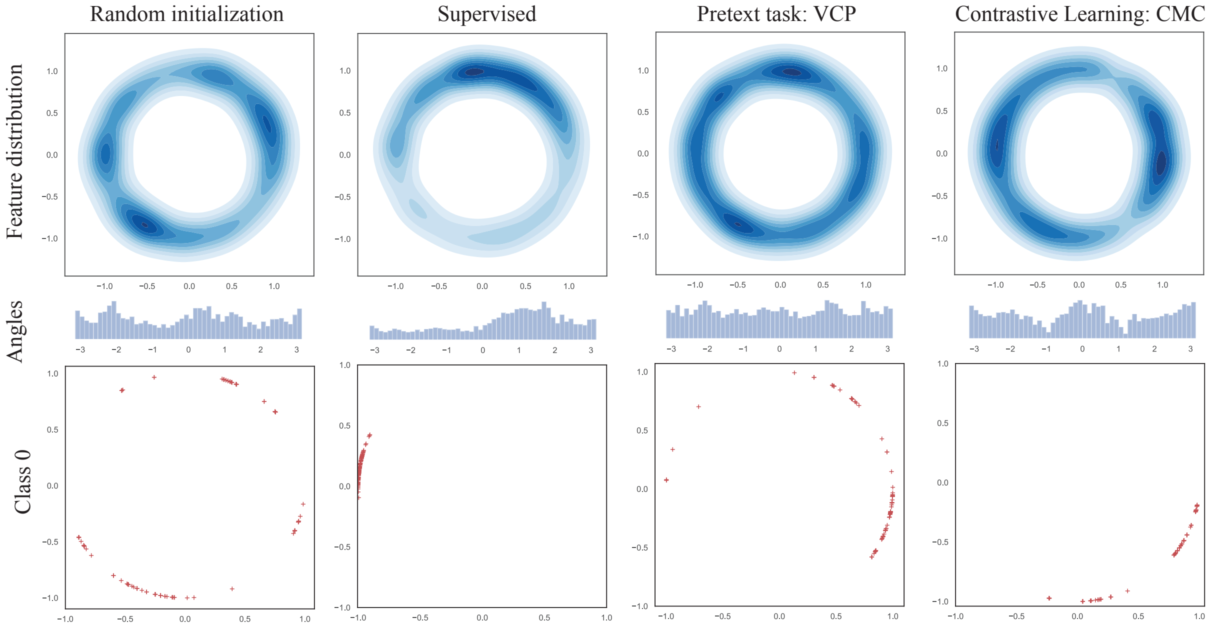


Figure 2: Representations of UCF101 dataset on $\mathcal{S}^1$. We plot the feature distributions with Gaussian kernel density estimation (KDE) in $\mathbb{R}^2$ and histograms of angles (i.e., $\arctan 2(y, x)$ or each point $(x, y) \in \mathcal{S}^1$. The features of class 0 are also visualized in points to show the distributed situation of a specific action.

use a unit hyper-sphere to empirically verify our assumption.

We visualize UCF101 [40] representations on $\mathcal{S}^1$ obtained via four methods:

- Random initialization.
- Supervised learning.
- Pretext task-based self-supervised learning.
- Unsupervised / self-supervised contrastive learning.

As there are a lot of pretext task based methods, we visualized them and found that they share similar performances when being projected in the sphere feature space. Here, we only visualize VCP [15] in Fig. 2. Visualizations of some other methods are shown in the supplementary materials.

From Fig. 2, we can clearly observe that after the supervised learning, the feature distribution is more concentrated, especially for features from the same action category. This is different from the other methods because no action labels are available for them. Random initialization without training, or using different types of self-supervised learning methods, the features tend to distribute uniformly in the feature space even though there may be small biases. In spite of different optimization targets between pretext tasks and contrastive learning, after training, the learned features share similarities: (1) features are still distributed uniformly on the hyper-sphere and (2) features from the same category are more concentrated than those in random initialization. This evidence strongly supports our assumption.

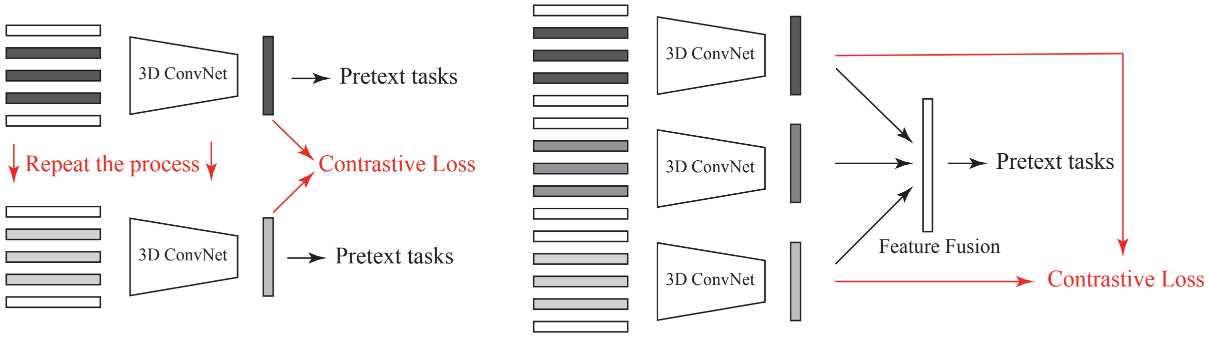
Based on the consistence in the feature space, we can infer that utilizing pretext tasks and contrastive learning together can help train a network in a self-supervised manner. Next, we will introduce the joint optimization framework, which combines pretext tasks with contrastive learning methods.

PCL: Joint Optimization Framework

There are plenty of pretext tasks, and some tasks use only one video clip to conduct experiments. For example, 3DRotNet rotated the input video clip and trained the model by predicting the rotation angles. Some tasks use multiple video clips during training, such as VCOP, which shuffled the temporal order of several video clips and trained the network by prediction the order of input clips. The training styles for almost all pretext tasks can be divided into two categories, single-clip methods and multi-clip methods.

We illustrate the use of our proposal in Fig. 3. For single-clip methods, the contrastive loss will use the encoded features from the backbone network. As contrastive loss requires positive pair and negative pairs to train, the encoding process is duplicated. The input video clip is generated from the same video as the original path, using similar kinds of data augmentation methods, such as random spatial cropping and random temporal cropping. Then the input video clips of both branches are different although they originate from one instance, which can be treated as a positive pair. Negative pairs are taken directly from one batch data because different samples are from different videos in one training batch.

For multi-clip methods, different video clips are set as inputs and they are encoded to features using a shared encoder. These features are natural positive pairs because they



(a) Pretext-Contrastive Learning in single-clip methods

(b) Pretext-Contrastive Learning in multi-clip methods

Figure 3: The use of PCL in pretext task-based methods. (a) For single-clip methods, two different clips from the same video will be processed and the contrastive loss will be calculated among one batch data. (b) For multi-clip methods, different clips from the same video have been already processed, and the contrastive loss can be easily calculated.

are from the same video. Negative pairs are also from video clips from one batch data.

It can be observed that it is very simple to construct a joint optimization framework based on any pretext task baseline method, and the final training loss becomes

$$\mathcal{L}_{total} = \mathcal{L}_{pretext} + \alpha \mathcal{L}_{contrast}, \tag{6}$$

where α is a weight to balance losses between pretext tasks and contrastive learning.

Experiments

To demonstrate the effectiveness of the proposed PCL, we used (1) several pretext task baseline methods; (2) different contrastive learning loss functions; (3) different backbone networks; and (4) different evaluation tasks to prove the success of our PCL.

Experimental Settings

There are several pretext task-based methods in self-supervised video representation learning. We chose three recent works: 3DRotNet [9], VCOP [14], and VCP [15]. 3DRotNet is trained by recognizing the rotated angles of the input video clip. VCOP aims to detect the correct orders of several input video clips. VCP conducts different types of transformations and the network is trained to distinguish which transformation has been performed. These pretext tasks, as well as the training styles, are different. For example, 3DRotNet is a one-clip method while VCOP and VCP use several video clips as input data.

For the network backbone, there are several 3D CNNs such as C3D [41], R3D [42], and R(2+1)D [43]. Different network backbones were used in our experiments to eliminate model biases. R3D in VCOP and VCP, and ResNet-18-3D in 3DRotNet are composed of 3D convolution instead of 2D convolution in the original ResNet [44] while the numbers of convolutional layers in each residual block vary. To compare with the baselines, we used the same network architectures as them. It is possible to use other network architectures such as I3D [45], S3D [46], or other deeper networks, but we simply follow the baselines for fair comparison.

We also note that in 3DRotNet [9], two different modalities were tested, namely original RGB frame and frame difference. The frame difference is also called residual clips, which was proven to effectively represent motion information in [47] and has been used in many existing methods [12, 9, 31].

As we have introduced two loss functions of contrastive learning, NCE loss and AlignUniform loss, we also attempted to combine each of them with pretext task baselines.

Two commonly used datasets, UCF101 [40] and HMDB51 [48] were used in our experiments. The experimental settings, such as data size and data transformations, are the same as those of the corresponding baselines.

For more detailed experimental settings, please refer to our supplementary materials.

Evaluation Metrics

To evaluate the performance of the trained models, two evaluation tasks were used: video retrieval and video recognition. After self-supervised training, the trained models can be evaluated directly in video retrieval tasks on both UCF101 and HMDB51. Note that the self-supervised learning part was only conducted on UCF101 *split* 1. Therefore, when conducting video retrieval on UCF101, the task-level generalization ability was tested. When conducting video retrieval on HMDB51 using the same model, both task-level and dataset-level generalization ability were tested.

Action recognition is a fundamental task in video feature learning. Following previous works, we also conducted experiments by fine-tuning trained models on both UCF101 and HMDB51 datasets to check the transfer learning ability of the models.

Table 1: Comparisons with baselines. Video retrieval results are evaluated on *split* 1 of UCF101 and HMDB51. Video recognition results are averaged on three splits. Best results in each block are in **bold**. *nce* and *au* are two different contrastive loss functions and R18 is short for ResNet-18-3D. † indicates methods evaluated on *split* 1 only. *More results and analysis can be found in supplementary material.*

Method	Backbone	Modality	UCF101						HMDB51					
			Top1	Top5	Top10	Top20	Top50	Recog	Top1	Top5	Top10	Top20	Top50	Recog
VCP														
Baseline	C3D	RGB	17.3	31.5	42.0	52.6	67.7	68.5	7.8	23.8	35.3	49.3	71.6	32.5
Baseline	C3D	Res	22.8	40.1	50.4	61.1	76.8	76.5†	9.0	27.2	39.4	56.2	77.3	44.8†
+ PCL (au)	C3D	RGB	24.9	39.8	48.9	59.5	72.5	68.3	11.3	27.9	41.2	55.5	75.5	33.4
+ PCL (au)	C3D	Res	31.5	49.4	58.6	68.9	81.8	**79.7**	10.8	29.8	42.1	58.6	80.4	**42.3**
+ PCL (nce)	C3D	RGB	35.4	50.9	59.6	69.6	80.8	65.0	**14.6**	**36.1**	**49.7**	**63.9**	**81.5**	30.6
+ PCL (nce)	C3D	Res	**38.2**	**56.8**	**67.0**	**75.5**	**86.3**	78.7	14.3	33.6	46.9	62.5	80.8	39.8
Baseline	R3D	RGB	18.6	33.6	42.5	53.5	68.1	66.0	7.6	24.4	36.3	53.6	76.4	31.5
Baseline	R3D	Res	25.6	43.0	53.2	64.8	79.2	77.0†	10.8	30.6	40.4	58.5	78.9	37.8†
+ PCL (au)	R3D	RGB	30.8	47.4	57.8	68.0	80.5	66.0	11.0	30.0	43.3	59.7	80.2	32.1
+ PCL (au)	R3D	Res	33.4	53.0	63.5	72.3	83.6	**79.6**	13.1	34.9	50.4	64.8	83.6	**43.2**
+ PCL (nce)	R3D	RGB	35.0	51.1	60.9	71.0	82.4	63.9	13.5	33.2	47.1	63.8	81.6	30.3
+ PCL (nce)	R3D	Res	**40.5**	**59.4**	**68.9**	**77.4**	**87.0**	79.5	**16.8**	**38.4**	**53.4**	**68.9**	**85.1**	41.7
Baseline	R(2+1)D	RGB	19.9	33.7	42.0	50.5	64.4	66.3	6.7	21.3	32.7	49.2	73.3	32.2
Baseline	R(2+1)D	Res	9.9	24.5	35.7	49.2	66.9	77.5†	6.7	19.3	30.9	44.7	70.2	27.4†
+ PCL (au)	R(2+1)D	RGB	22.2	36.3	45.5	56.3	69.3	69.9	9.8	26.8	38.9	55.3	75.0	34.8
+ PCL (au)	R(2+1)D	Res	**29.7**	**49.4**	**57.7**	**68.0**	**81.9**	**80.7**	**11.0**	**31.6**	**44.8**	**60.4**	**80.6**	**44.6**
+ PCL (nce)	R(2+1)D	RGB	17.3	28.5	36.0	44.2	58.5	53.6	6.9	20.2	31.3	46.5	70.0	20.2
+ PCL (nce)	R(2+1)D	Res	8.0	23.3	33.8	46.9	65.3	77.8	5.0	18.3	29.2	45.0	68.7	40.1
3DRotNet														
Baseline	R18	RGB	14.2	25.2	33.5	43.7	59.5	62.9	6.2	18.7	31.0	46.6	70.5	33.7
Baseline	R18	Res	14.5	30.5	40.2	53.1	70.7	70.8	6.0	21.6	33.8	49.1	71.8	40.0
+ PCL (au)	R18	RGB	13.5	25.7	33.5	43.1	58.7	73.4	6.3	18.5	30.1	45.0	68.8	35.7
+ PCL (au)	R18	Res	16.0	34.2	44.1	55.9	71.4	79.6	6.9	22.1	35.8	50.3	74.3	43.1
+ PCL (nce)	R18	RGB	18.2	30.7	39.1	49.2	64.4	72.8	7.2	19.4	30.5	45.9	69.2	35.8
+ PCL (nce)	R18	Res	**22.7**	**41.8**	**52.3**	**63.2**	**77.2**	**82.3**	**8.0**	**25.7**	**38.6**	**53.7**	**75.6**	**43.2**
VCOP														
Baseline	C3D	RGB	12.5	29.0	39.0	50.6	66.9	65.6	7.4	22.6	34.4	48.5	70.1	28.4
+ PCL (au)	C3D	RGB	18.1	32.8	40.5	49.5	63.7	67.1	8.6	23.8	34.9	49.5	70.4	30.6
+ PCL (au)	C3D	Res	16.0	30.1	39.2	49.3	64.8	**77.2**	6.5	20.7	31.0	46.4	69.6	**40.3**
+ PCL (nce)	C3D	RGB	**30.0**	**49.1**	**58.6**	**69.3**	**81.8**	67.1	**13.7**	**36.3**	**48.0**	**63.7**	**81.2**	31.8
+ PCL (nce)	C3D	Res	14.4	28.2	37.7	48.9	65.2	75.6	6.7	21.3	31.1	45.6	70.5	37.4

Results and Discussion

In this section, we first compare our proposed method with baseline methods. To further prove the effectiveness of our strategy, we also compare our results with those of current state-of-the-art methods. We mainly used VCP as the baseline pretext task and used C3D, R3D, or R(2+1)D as the network backbone. For the other two methods, 3DRotNet and VCOP, we used the same mainstream backbones reported in the corresponding papers: ResNet-18-3D for 3DRotNet and C3D for VCOP.

Comparison with Baselines

All models were trained on UCF101 *split* 1 and tested on both UCF101 and HMDB51 datasets. For baseline methods, VCOP and VCP only used RGB modality. We further found that by using our proposed method together with residual clips, the performance can be significantly improved. Results are presented in Table. 1.

For VCP, the combination with the PCL (AlignUniform loss) yielded 24.9% at top 1 retrieval accuracy on UCF101 dataset, which is 7.6% points above the C3D baseline. The combination with the other contrastive loss, NCE loss, achieved 35.4%. By using residual clips together with our proposed method, the performance can be further improved. The best practice in such training settings becomes a combination with NCE loss and residual inputs, reaching 38.2% whose number is more than double that of the baseline. On the HMDB51 dataset, this setting also has high accuracy, which indicates that the combination of pretext tasks and contrastive learning can help train the network to extract better motion features. With respect to the action recognition results, better performances were also achieved on both UCF101 and HMDB51. The best performance using our PCL is 79.7% for UCF101 and 42.3% for HMDB51, which are almost 10% points over the corresponding baselines. For the other backbone networks, R3D and R(2+1)D,

Table 2: Comparison with state-of-the-art methods in video retrieval on UCF101.

Methods	Backbone	Top1	Top5	Top10	Top20	Top50
PRP [18]	C3D	23.2	38.1	46.0	55.7	68.4
PacePrediction [20]	C3D	31.9	49.7	59.2	68.9	80.2
Ours (VCP + PCL)	C3D	**38.2**	**56.8**	**67.0**	**75.5**	**86.3**
PRP [18]	R3D	22.8	38.5	46.7	55.2	69.1
IIC [31]	R3D	36.5	54.1	62.9	72.4	83.4
Ours (VCP + PCL)	R3D	**40.5**	**59.4**	**68.9**	**77.4**	**87.0**
PRP [18]	R(2+1)D	20.3	34.0	41.9	51.7	64.2
PacePrediction [20]	R(2+1)D	25.6	42.7	51.3	61.3	74.0
Ours (VCP + PCL)	R(2+1)D	**29.7**	**49.4**	**57.7**	**68.0**	**81.9**
RTT [19]	R18	26.1	48.5	59.1	69.6	82.8
PacePrediction [20]	R18	23.8	38.1	46.4	56.6	69.8
Ours (VCP + PCL)	R18	**33.2**	**52.7**	**62.4**	**71.7**	**83.7**

similar performances can be achieved, which indicates that our proposal can be generalized to different networks.

For 3DRotNet, we used the same network architecture as the baseline [9]. Improvements can be achieved using our PCL in both video retrieval and video recognition on UCF101 and HMDB51. Similar conclusions can be drawn when we set VCOP as the baseline.

Comparison with the State-of-the-art Methods

The baselines used in our study, VCP [15], 3DRotNet [9], and VCOP [14] are not currently state-of-the-art methods. Some recent works have used new pretext tasks such as pace prediction [20] or more complex temporal transformation recognition [19] and achieved better performances. Here we compared our methods with state-of-the-art methods to demonstrate the effectiveness of our PCL. Note that it is possible to apply our proposal with these new pretext tasks. Further, we want to clarify that there are some other works that used larger pre-trained datasets together with audio or text information of videos and achieved even higher performance. Here, we did not include them and only referred to these methods using similar settings for fair comparison.

The results are shown in Table. 2 and Table. 3. In video retrieval, we can see that our PCL can outperform other state-of-the-art methods, no matter which backbone is used. In video recognition, we can observe that without our proposal, the performances of the baseline methods are lower than those of recent state-of-the-art methods. However, with the proposed PCL, which only has minor changes in the baseline, the performance can be significantly improved and in some settings, it performs better than the state-of-the-art methods.

Discussion

The mechanism of pretext tasks is not well explained in theory. Researchers aim to design tasks related to their final target tasks. For example, action retrieval and action recognition require temporal information to distinguish between samples. Thus, temporal related tasks have been proposed. However, for individual pretext task, it is not clear which is the best, except based on a particular performance metrics.

For contrastive learning, the basic idea is to distinguish one sample from another. However, determining why it functions well for motion representation extraction is difficult because spatial information may sometimes be enough. And same action clips in different instances will be treated as negatives during training.

Owing to many unclear issues, it is difficult to model the training target in a clear way, and we tried to use the optimization formulations and the hyper-sphere feature space to show the differences and consistencies of these training targets. The evidence proves that our assumption is correct, and the combination of pretext tasks and contrastive loss also demonstrates the success of our proposal.

A simple combination of one pretext task and one contrastive loss can also provide significant improvements. We conducted extensive experiments to show the generality of this training strategy.

Table 3: Comparison with state-of-the-art methods in video recognition.

Method	UCF101	HMDB51
C3D (VCOP) [14]	65.6	28.4
C3D (VCP) [15]	68.5	32.5
C3D (PRP) [18]	69.1	34.5
C3D (RTT) [19]	69.9	39.6
C3D (VCP + PCL, ours)	**79.7**	**42.3**
R3D (VCOP) [14]	64.9	29.5
R3D (VCP) [15]	66.0	31.5
R3D (PRP) [18]	66.5	29.7
R3D (IIC) [31]	74.4	38.3
R3D (VCP + PCL, ours)	**79.5**	**41.7**
R(2+1)D (VCP) [15]	66.3	32.2
R(2+1)D (VCOP) [14]	72.4	30.9
R(2+1)D (PRP) [18]	72.1	35.0
R(2+1)D (RTT) [19]	81.6	46.4
R(2+1)D (PacePrediction) [20]	75.9	35.9
R(2+1)D (VCP + PCL, ours)	**80.7**	**44.6**
R18 (3DRotNet) [9]	70.8	40.0
R18 (RTT) [19]	77.3	47.5
R18 (3DRotNet + PCL, ours)	**82.3**	**43.2**

Conclusion

In this paper, we proposed the training of self-supervised video representation learning networks using both pretext tasks and contrastive learning. The proposed PCL is not a single method, but a joint optimization framework. The comparative analysis of the optimization targets of pretext tasks and contrastive learning provided us the assumption that both types of methods attempt to learn similar feature representations despite their different training ways. Feature visualization on the hyper-sphere proved our assumptions and it is possible and more reasonable to combine them to improve the performance of self-supervised learning. Experiments using different pretext task baselines with different network backbones in different evaluation tasks on two benchmark datasets revealed the effectiveness and the universality of our proposal. Results showed that the proposed PCL is sufficiently flexible enough and can be easily applied to almost any existing pretext task method.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[2] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[3] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European Conference on Computer Vision*, pp. 69–84, Springer, 2016.

[4] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *arXiv preprint arXiv:1803.07728*, 2018.

[5] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European conference on computer vision*, pp. 649–666, Springer, 2016.

[6] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

[7] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[8] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[9] L. Jing, X. Yang, J. Liu, and Y. Tian, "Self-supervised spatiotemporal feature learning via video rotation prediction," *arXiv preprint arXiv:1811.11387*, 2018.

[10] D. Kim, D. Cho, and I. S. Kweon, "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.

[11] I. Misra, C. L. Zitnick, and M. Hebert, "Shuffle and learn: unsupervised learning using temporal order verification," in *European Conference on Computer Vision*, pp. 527–544, Springer, 2016.

[12] B. Fernando, H. Bilen, E. Gavves, and S. Gould, "Self-supervised video representation learning with odd-one-out networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3636–3645, 2017.

[13] H.-Y. Lee, J.-B. Huang, M. Singh, and M.-H. Yang, "Unsupervised representation learning by sorting sequences," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 667–676, 2017.

[14] D. Xu, J. Xiao, Z. Zhao, J. Shao, D. Xie, and Y. Zhuang, "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.

[15] D. Luo, C. Liu, Y. Zhou, D. Yang, C. Ma, Q. Ye, and W. Wang, "Video cloze procedure for self-supervised spatio-temporal learning," *arXiv preprint arXiv:2001.00294*, 2020.

[16] S. Benaim, A. Ephrat, O. Lang, I. Mosseri, W. T. Freeman, M. Rubinstein, M. Irani, and T. Dekel, "Speednet: Learning the speediness in videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9922–9931, 2020.

[17] H. Cho, T. Kim, H. J. Chang, and W. Hwang, "Self-supervised spatio-temporal representation learning using variable playback speed prediction," *arXiv preprint arXiv:2003.02692*, 2020.

[18] Y. Yao, C. Liu, D. Luo, Y. Zhou, and Q. Ye, "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.

[19] S. Jenni, G. Meishvili, and P. Favaro, "Video representation learning by recognizing temporal transformations," *arXiv preprint arXiv:2007.10730*, 2020.

[20] J. Wang, J. Jiao, and Y.-H. Liu, "Self-supervised video representation learning by pace prediction," *arXiv preprint arXiv:2008.05861*, 2020.

[21] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 1735–1742, IEEE, 2006.

[22] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[23] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," *arXiv preprint arXiv:1808.06670*, 2018.

[24] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018.

[25] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," *arXiv preprint arXiv:1906.05849*, 2019.

[26] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," *arXiv preprint arXiv:1911.05722*, 2019.

[27] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020.

[28] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *arXiv preprint arXiv:2002.05709*, 2020.

[29] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning," *arXiv preprint arXiv:2006.07733*, 2020.

[30] T. Han, W. Xie, and A. Zisserman, "Video representation learning by dense predictive coding," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 1483–1492, IEEE, 2019.

[31] L. Tao, X. Wang, and T. Yamasaki, "Self-supervised video representation learning using inter-intra contrastive framework," *arXiv preprint arXiv:2008.02531*, 2020.

[32] A. Owens and A. A. Efros, "Audio-visual scene analysis with self-supervised multisensory features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 631–648, 2018.

[33] B. Korbar, D. Tran, and L. Torresani, "Cooperative learning of audio and video models from self-supervised synchronization," in *Advances in Neural Information Processing Systems*, pp. 7763–7774, 2018.

[34] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, "Videobert: A joint model for video and language representation learning," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 7464–7473, 2019.

[35] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304, 2010.

[36] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," *arXiv preprint arXiv:2005.10242*, 2020.

[37] A. Piergiovanni, A. Angelova, and M. S. Ryoo, "Evolving losses for unsupervised video representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 133–142, 2020.

[38] T. Han, W. Xie, and A. Zisserman, "Memory-augmented dense predictive coding for video representation learning," *arXiv preprint arXiv:2008.01065*, 2020.

[39] Y. Tian, Z. Che, W. Bao, G. Zhai, and Z. Gao, "Self-supervised motion representation via scattering local motion cues," 2020.

[40] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[41] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497, 2015.

[42] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA*, pp. 18–22, 2018.

[43] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "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.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[45] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308, 2017.

[46] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 305–321, 2018.

[47] L. Tao, X. Wang, and T. Yamasaki, "Rethinking motion representation: Residual frames with 3d convnets for better action recognition," *arXiv preprint arXiv:2001.05661*, 2020.

[48] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: a large video database for human motion recognition," in *2011 International Conference on Computer Vision*, pp. 2556–2563, IEEE, 2011.