**Abstract**—The central idea of contrastive learning is to discriminate between different instances and force different views from the same instance to share the same representation. To avoid trivial solutions, augmentation plays an important role in generating different views, among which random cropping is shown to be effective for the model to learn a generalized and robust representation. Commonly used random crop operation keeps the distribution of the difference between two views unchanged along the training process. In this work, we show that adaptively controlling the disparity between two augmented views along the training process enhances the quality of the learned representations. Specifically, we present a parametric cubic cropping operation, ParamCrop, for video contrastive learning, which automatically crops a 3D cubic by differentiable 3D affine transformations. ParamCrop is trained simultaneously with the video backbone using an adversarial objective, so that it learns to increase the contrastive loss and thus gradually reduces the shared contents between two cropped views. Experiments show that this adaptive and gradual increase in the disparity yielded by ParamCrop is beneficial to learning a strong and generalized representation for downstream tasks, which is shown to be effective on multiple contrastive learning frameworks and video backbones.

**Index Terms**—Parametric cropping, contrastive learning, video representation learning.

### I. INTRODUCTION

Learning representations from massive unlabeled data is a prominent research topic in computer vision for reducing the need for laborious and time-consuming manual annotations [1], [2], [3], [4], [5]. In the video analysis paradigm, which is the focus of our work, such unsupervised learning strategies are more crucial because of their increased labeling difficulty caused by the ambiguous association between videos and their labels. Early works manually design proxy tasks for learning videos, either by generalizing methods from the image domain [6], [7] or by exploiting temporal properties of videos [8], [9], [10], [11], [12], [13], [14], where visual structures and contents are learned through solving these proxy tasks. Inspired by the instance discrimination task [15], contrastive based self-supervised approaches have achieved impressive performances [16], [17], [18], [19], [20], [21], which shows its potential to learn advanced semantic information from unlabelled videos. One of the key factors in the success of the contrastive learning framework is data augmentation [1], [20], [21] for different views of the same instance, which prevents the model from trivial solutions, and leads to robust visual representations. Among common data augmentation strategies, it is shown in [1] that random cropping is one of the most effective operations.

Most current approaches for contrastive learning use random cropping with completely random spatio-temporal location selections. It keeps the distribution of the difference between two cropped views unchanged along the training process, as showcased in grey (i.e., RC) in Fig. 1(b). Inspired by curriculum

---

*Zhiwu Qing* (Corresponding author: Nong Sang), *Ziyuan Huang*, *Shiwei Zhang*, *Mingqian Tang*, *Changxin Gao*, **Member, IEEE**, *Rong Jin*, *Marcelo H. Ang Jr* **Senior Member, IEEE**, and *Nong Sang* **

**ParamCrop: Parametric Cubic Cropping for Video Contrastive Learning**

Zhiwu Qing, Ziyuan Huang, Shiwei Zhang, Mingqian Tang, Changxin Gao, Member, IEEE, Rong Jin, Marcelo H. Ang Jr, Senior Member, IEEE, and Nong Sang

IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 25, 2023

Manuscript received 26 August 2022; revised 11 January 2023; accepted 1 February 2023. Date of publication 15 February 2023; date of current version 12 December 2023. This work was supported in part by the National Natural Science Foundation of China under Grant U22B2053, and in part by Alibaba Group through Alibaba Research Intern Program. The Associate Editor coordinating the review of this manuscript and approving it for publication was Dr. Jianlong Fu. (Corresponding author: Nong Sang; Shiwei Zhang.)

Zhiwu Qing, Changxin Gao, and Nong Sang are with the Key Laboratory of Image Processing and Intelligent Control, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: qzw@hust.edu.cn; cgao@hust.edu.cn; nsang@hust.edu.cn).

Ziyuan Huang and Marcelo H. Ang Jr are with the Advanced Robotics Centre, National University of Singapore, Singapore 119077 (e-mail: ziyuan.huang@u.nus.edu; mpeangh@nus.edu.sg).

Shiwei Zhang, Mingqian Tang, and Rong Jin are with Alibaba Group, Hangzhou 311100, China (e-mail: zhangjin.zsw@alibaba-inc.com; mingqian.tmq@alibaba-inc.com; jinrong.jr@alibaba-inc.com).

Digital Object Identifier 10.1109/TMM.2023.3244126

*Fig. 1.* (a) Comparison between our proposed ParamCrop framework and the traditional random cropping in the contrastive framework. ‘g’ is normal gradient, and ‘−g’ indicates reversed gradient. The parametric cubic crop in ParamCrop is connected after other random transformations. (b) (c) Preliminary experiments show that increasing the mean center distance (i.e., RC, RC++, and RC+++ between two cropping regions in the later stage of contrastive training can benefit the learned representations.

---

1520-9210 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
learning [22], we slightly increase the difficulty in the later stage of contrastive training by increasing the central distance between two cropped views (Fig. 1(b)). Fig. 1(c) shows that this yields a stronger representation for the downstream action recognition task.

Motivated by this, we propose to adaptively control the disparity between two views for contrastive learning. Specifically, we present a parametric cubic cropping dubbed ParamCrop, where cubic cropping refers to cropping a 3D cube from the input video. The central component of ParamCrop is a differentiable spatio-temporal cropping operation. This enables ParamCrop to be trained simultaneously with the video backbone and adjust the cropping strategy on the fly. The objective of ParamCrop is adversarial to the video backbone, i.e., to increase the contrastive loss. Hence, initialized with the simplest setting where two cropped views largely overlap, ParamCrop gradually increases the disparity between two views. Further, we introduce an early stopping strategy for ParamCrop to control the maximum disparity, since there exists a sweet spot in the intensity of augmentations for contrastive learning [23]. Compared to the auto augmentation approaches in the supervised setting [24], [25], [26], [27], our objective is radically different. The auto augmentation methods aim to increase the data diversity, while ParamCrop sets out to discover an adaptive cropping strategy to reasonably control the differences between two views along the training process.

We quantitatively evaluate the representations trained by ParamCrop on two downstream tasks, i.e., video action recognition and video retrieval. Notable improvements are observed on multiple mainstream contrastive learning frameworks and video backbones, which shows that the idea of adaptively increasing the disparity between two views along the training process is crucial to learning generalized representations.

Contributions: (a) We propose a cropping strategy that adaptively controls the disparity between two cropped views for self-supervised video contrastive learning; (b) We propose a differentiable cropping method that can be trained end-to-end together with the video backbone; (c) Extensive experiments on multiple downstream tasks and datasets demonstrate that the adaptive disparity controlled by ParamCrop can effectively improve video representations.

II. RELATED WORK

Self-supervised video representation learning: To avoid the laborious and time-consuming annotation process, a wide range of prior works have proposed different approaches for leveraging unlabelled data. Recent endeavors can be mainly divided into two categories, that is pretext task based approaches [6], [8], [9], [10], [11], [19], [28], [29], [30] and contrastive learning based ones [16], [17], [20], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43]. The former ones usually introduce a proxy task for the model to solve. Besides the simple generalization from the image domain [44], [45] such as rotation prediction [6] and solving puzzles [7], [9], other tasks include predictions on the temporal dimension such as speed prediction [11], frame/clip order prediction [8], [13], [19], [46], spatio-temporal overlap rate prediction [47], and predicting future frames [10], etc. Closely related to our work is contrastive learning based approaches, which were inspired by the instance discrimination task [15]. It requires the model to discriminate augmented samples from the same instance from other instances and map different views of the same instance to the same representation. Based on the formulation in [48], [16], [17] contrast between the representation of the predicted future frames and that of the real ones. Some works exploit video pace variation as augmentation and contrast between representations with different paces [28], [30], [49]. VideoMoCo [35] learns a sub network that adaptively discards frames to increase the difficulty of contrastive learning. TeG [38] and CARL [37] propose learning a fine-grained (e.g., frame-wise) temporal feature alignment to explore temporal granularity. Furthermore, ConST-CL [33] designs a region-based self-supervised pretext task to learn spatio-temporally fine-grained representations. Whether it is in the video paradigm, which is the focus of this paper, or in the image domain, augmentations are all shown to be critical to learning a strong representation. Yet all of them apply random cropping with completely random spatio-temporal location and uniform scale variation parameters along the whole training process. We build our approach for video contrastive learning upon the simplest contrastive framework [1], [2] and show that parameterized cubic cropping controlling the change process is conducive to the improvement of learned representations.

Data Augmentation: The importance of the data augmentation has already been discovered in the supervised learning. The main objective of data augmentation in supervised settings is to enhance the diversity of the training data so that the model can learn generalized representations. The hand-craft data augmentations confuse the network by erasing information [50] or mixing difference samples [51], [52]. To reduce the dependence on human expertise, automatic data augmentations are proposed to search the combination of augmentation policies by undifferentiable methods [24], [25], [26] or online learnable strategy [53]. The aim of ParamCrop is radically different from these works, in that they aim to learn an augmentation strategy for it to be used later in supervised training, while ParamCrop aims to generate cropped views to adaptively adjust the training difficulty in contrastive pre-training, which is achieved by being optimized together with the backbone during the pre-training process. In unsupervised learning, a learnable color transformation is proposed in [54] to improve the robustness of the representation. Concurrently, a semantic-aware spatial sampling module is proposed in [55], while our ParamCrop focuses more on the adaptive disparity between views with respect to the training process.

III. METHOD

This section introduces the proposed parametric cubic cropping framework, ParamCrop, based on contrastive learning. The objective of ParamCrop is to adaptively control the cropping disparity between two generated views during the contrastive training process. To this end, we propose a differentiable spatio-temporal cropping operation, which can utilize the cropping parameters regressed by the cropping networks (i.e., an
MLP for regressing the cropping parameters) to realize the 3D cropping operation. This enables the cropping operation to be jointly optimized with the video backbone. Two identical but independent cropping modules are connected respectively to each of the views. To gradually increase the disparity between views in training, we first initialize the cropping networks so that two crops share a similar space-time location, and then train the networks adversarially along with the video backbone using a gradient reversal strategy. The overall framework, as well as the detailed workflow, is visualized in Fig. 2.

A. Contrastive Learning

Our ParamCrop framework is built upon recent simplified contrastive learning frameworks [1], [2], where the model is trained to maximize the agreement between two augmented views of the same instance and minimize that from different instances. Suppose there are \( N \) different samples, we can generate \( 2N \) augmented views and the contrastive loss can be written as:

\[
\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]
\]

(1)

where \( \ell(i, j) \) defines the loss between two paired samples:

\[
\ell(i, j) = \log \frac{\exp(c_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{I}_{[i \neq j]} \exp(c_{i,k}/\tau)},
\]

(2)

where \( c_{i,j} \) is the cosine similarity between the representation view \( i, j \), and \( \tau \) is the temperature parameter. Typically, the two views are generated by the same set of augmentation, usually consisting of random cropping, color jittering, etc. In standard contrastive methods, the augmentation strategy keeps unchanged along the training process. Hence, the distribution of the view difference is consistent along the training process.

B. Differentiable 3D Affine Cropping

For the proposed ParamCrop to control the cropping disparity during the training process, the cropping operation is firstly required to be trainable. Inspired by Spatial Transformer Network (STN) [56], we extend the image affine transformations to video 3D affine transformations for cropping cubes from original videos in a differentiable way.

Cubic cropping with 3D affine transformations: Before introducing the 3D affine transformation, we first define mathematical notations \( x^a, x^c \) as the original videos and the cropped videos, respectively. Then we further define the original video width \( w_o \), the cropped video width \( w_c \), the temporal length of the original video \( t_o \) and the cropped video \( t_c \). Fig. 3 illustrates their meanings intuitively.

With these notations, a 3D affine transformation matrix \( A_\phi \) for calculating the transformation relationship from the homogeneous coordinate in the cropped video to the original video can be defined as follows:

\[
A_\phi = \begin{bmatrix}
               s_p \cos(\theta) & -s_p \cos(\theta) & 0 & \Delta x \\
               s_p \sin(\theta) & s_p \sin(\theta) & 0 & \Delta y \\
               0 & 0 & s_t & \Delta t 
             \end{bmatrix}
\]

(3)

where \( s_p = w_c/w_o \) is the region scale, \( \theta \) refers to the spatial rotation angle, \( (\Delta x, \Delta y) \) indicates the spatial center position offsets, \( s_t = t_c/t_o \) is the temporal scale and \( \Delta t \) means the temporal offset. Since we only implement cropping operations, irrelevant parameters in \( A_\phi \) are set to 0 by default. Therefore, there are altogether six parameters that can be learned in the affine matrix for 3D cropping operation.

With the six parameters, the 3D affine matrix \( A_\phi \) is essentially a coordinate transformation function that maps the coordinate system from the cropped video \( x^c \) to the original video \( x^o \) with scaling, rotation, and translation in the spatial dimensions as well as scaling and translation in the temporal dimension. Given the
homogeneous coordinate \((x_i^c, y_i^c, t_i^c, 1)\) in the cropped videos \(x^c\), its corresponding coordinate \((x_i^o, y_i^o, t_i^o)\) in the original video \(x^o\) can be calculated as follows:

\[
\begin{pmatrix}
  x_i^o \\
  y_i^o \\
  t_i^o \\
  1
\end{pmatrix} = A_\phi \begin{pmatrix}
  x_i^c \\
  y_i^c \\
  t_i^c \\
  1
\end{pmatrix}
\]  

(4)

where all the coordinates in both original and cropped videos are normalized, i.e., \(\{x_i^o, y_i^o, t_i^o, x_i^c, y_i^c, t_i^c\} \in [-1, 1]\). The coordinates in \(x^c\) are known, which are uniformly distributed between \([-1, 1]\) according to the resolution of \(x^o\). For example, to crop a video \(x^o\) with a spatio-temporal resolution of \(16 \times 112^2\), we first generate 16 uniform points between \([-1, 1]\) in the temporal axis and 112 points in two spatial axes. Then the obtained 3D grids with \(16 \times 112^2\) coordinates in \(x^c\) are transformed to \(x^o\) by (4). Because the transformed coordinates in \(x^o\) may not accurately correspond to the pixel, the pixel values for \((x_i^o, y_i^o, t_i^o)\) are sampled by bilinear interpolation. This cropping process is visualized in Fig. 4 for intuitive understanding. Please refer to Appendix for more details.

Generating transformation parameters: To enable this cropping process to be learnable, we employ a multi-layer perceptron to predict the aforementioned six 3D affine transformation parameters by:

\[
v = \sigma(W_2 \delta(W_1 n))
\]  

(5)

where \(n \in \mathbb{R}^n\) is a random vector, which can provide diversities for different cropped regions. \(W_1 \in \mathbb{R}^{d \times m}, W_2 \in \mathbb{R}^{6 \times d}\) are parameters of the multi-layer perceptron and \(\delta\) denotes ReLU activation between two linear layers. The elements in \(v\) corresponds to \([s_p, s_t, \theta, \Delta x, \Delta y, \Delta t]^\top\), which is a vector composed by the six controlling parameters in the 3D affine matrix \(A_\phi\). The sigmoid function \(\sigma\) is employed to constrain the range of output values to avoid generating meaningless views. Moreover, for the cropped region scale \(s_p\) and temporal scale \(s_t\), a near-zero or zero value indicates that an extremely small cube is cropped, which is meaningless as well and degenerate the learned representation. Therefore, we set a limited interval for each transformation parameter in \(v\):

\[
v = \begin{bmatrix}
  s_p \\
  s_t \\
  \theta \\
  \Delta x \\
  \Delta y \\
  \Delta t
\end{bmatrix} + v \odot \begin{bmatrix}
  s_p \\
  s_t \\
  \theta \\
  \Delta x \\
  \Delta y \\
  \Delta t
\end{bmatrix}
\]  

(6)

where \(\odot\) indicates the element-wise multiplication and \(\hat{s}_p\) and \(\hat{s}_t\) represent the minimum and maximum value allowed during training, respectively. To ensure that the cropped region will always fall within the cube of the original video, the offsets in spatial and temporal need be constrained by: \(\Delta x = \Delta y = s_p - 1\), \(\Delta x = \Delta y = 1 - s_p\), \(\Delta t = s_t - 1\) and \(\Delta t = 1 - s_t\). This avoids exceeding the boundary of the original video and yielding invalid views.

Temporal sampling strategy: In our 3D affine cropping, especially in the spatial dimensions, pixel values in the output video are obtained by bilinear interpolation. For input 3D video tubes, we should theoretically use trilinear interpolation for cropping. However, simply performing trilinear interpolation on the temporal axis will fuse the visual cues between two consecutive frames, which is physically meaningless. Therefore, we employ a differentiable way to round down the transformed temporal coordinates \(t_i^c\) in the original video \(x^o\):

\[
t_i^c = t_i^o - \text{StopGradient}(t_i^o - \text{Round}(t_i^o)).
\]  

(7)

In this way, we essentially perform bilinear interpolation on frame \(t_i^c\).
shows the code of our two diagonal regions in the space-time cube. Blindly maximizing the con-
trols the optimization approach, and some constraints when optimizing for the objective.

Objective: gradual increase of view disparity: Recall that the objective of the ParamCrop framework is to control the cropping strategy adaptively. Inspired by curriculum learning [22], we aim to crop two views based on the differentiable 3D affine transformation such that the disparity between two cropped views gradually increases along the training process. Because the weights of the multi-layer perceptron for generating transformation parameters are randomly initialized, which usually contains small values, it generally maps the random noise vector \( \mathbf{n} \) to values around 0. This means that the initially cropped cubes has a substantial overlap and thus a high similarity. With this initialization condition, we set the goal to be in the opposite direction of the video optimization direction, that is, to be adversarial to the contrastive loss, i.e., \( \varphi^* = \text{argmax}_\varphi \mathcal{L} \), where \( \varphi^* \) denotes the optimal solution to the cropping module. This achieves a gradual increase in the disparity of two augmented views. The rationale behind this training objective is that two identical views for the same video naturally give the lowest contrastive loss. Therefore, to gradually generate distinct views, ParamCrop needs to gradually increase the contrastive loss.

Optimization: gradient reversal: To optimize this goal, we apply a simple gradient reversal for the multi-layer perceptron during the back propagation. As the backbone aims to learn representations by minimizing the contrastive loss, this reversal operation forces the cropping module to maximize the contrastive loss, so that the disparity between two augmented views can be gradually increased. Although the parameters in two cropping modules are not shared, their gradients are derived from the same contrastive learning loss. Therefore, the updates of their parameters are highly correlated, which means that the two cropping views are also related.

Constraints: early stopping: Blindly maximizing the contrastive loss without any constraint may cause the cropping module to rapidly converge into an extreme position to fulfill the training goal, e.g., two diagonal regions in the space-time cube. Since it keeps to maximize the contrastive loss, it further yields views with no shared contents until the end of training. This makes the disparity of two views too large, which is difficult for the model to learn robust representations. As recent research shows that a sweet spot exists in the intensity of augmentations for a generalized representation [23], we propose to apply an early stopping strategy to avoid the extreme solutions:

\[
    v_i = \begin{cases} 
    v_i & |v_i - 0.5| \leq 0.5 - b_{\text{stop}} \\
    \text{StopGradient}(v_i) & \text{else}
    \end{cases}
\]  

(8)

where \( v_i \) is the \( i \)-th entry of \( v \), and \( b_{\text{stop}} \in [0, 0.5] \) is the bound of gradient stopping. \( b_{\text{stop}} = 0 \) is equivalent to disabling the early stopping, while \( b_{\text{stop}} = 0.5 \) disables the training for the cropping module. With the early stopping strategy, the video backbone can provide the augmented views with more diversities by avoiding
Fig. 5. The curve of IoU (dashed) and center Manhattan distance (solid) between two cropping regions by ParamCrop along the training process. The red/blue curve indicates the training process with/without Early Stopping. The yellow curve is a smooth center distance curve simulated manually. (a1, a2, b1, b2, c1, c2) show the relative position between the two cropping regions at different stages in training. (a2, b2, c2) employs early stopping while (a1, b1, c1) does not.

Fig. 6. Sensitivity analysis for the detach bound $b_{stop}$ in early stopping on both HMDB51 and UCF101. The default value of $b_{stop}$ is 0.2.

Fig. 7. Sensitivity analysis for minimum cropping region scale $\hat{s}$ in (6). Both $s_p$ and $s_t$ are equal to $\hat{s}$. The minimum region scale $\hat{s}$ is set to 0.5 by default.

Fig. 8. Sensitivity analysis for maximum cropping region scale $\hat{s}$ in (6). Both $s_p$ and $s_t$ are equal to $\hat{s}$. The maximum region scale $\hat{s}$ is set to 1.0 by default.

Fig. 9. Comparisons between ParamCrop and random cropping. We perform fully fine-tuning on HMDB51 and UCF101 with different training periods.

Fig. 10. Comparisons between ParamCrop and random cropping. Linear evaluations are performed on HMDB51 and UCF101 with different training periods.

IV. EXPERIMENTS

Training dataset: We pre-train the models on the training set of Kinetics-400 [57], containing 240 k training videos with each lasting about 10 seconds.

Pre-training settings: We adopt S3D-G [58] and R2D3D [59] as our backbone, and employ SimCLR [11] and MoCo [2] as our contrastive learning frameworks. The 3D affine cropping module takes 64 frames with $128^2$ resolution as inputs, and outputs the augmented views with spatial-temporal size $16 \times 112^2$ to the video backbone. LARS [60] is employed as optimizer. The batch size, learning rate, and weight decay are set to 1024, 0.3, and 1e-6, respectively. Color jittering and random horizontal flip are employed before our cropping module. Unless otherwise specified, we set the minimum scales $s_p$ and $s_t$ to 0.5 and the maximum scales $s_p$ and $s_t$ to 1.0. The detach bound $b_{stop}$ for early stopping are set to 0.2. When compared with other methods, the models are pre-trained with 100 epochs. For ablation studies, if not specific, we employ S3D-G networks with SimCLR for pre-training, and network are pre-trained with 20 epochs for efficiency.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Evaluations: The evaluations of the trained representations are performed on two downstream tasks, i.e., action recognition and video retrieval, on two public datasets: (i) UCF101 [61] dataset with 13320 videos from 101 action categories; (ii) HMDB51 [62] dataset contains 6849 videos from 51 action classes.

Fully fine-tuning (FT) and Linear fine-tuning (LFT) settings: We use Adam [63] with batch size of 128 and weight decay $1e^{-3}$. The learning rate for fully fine-tuning is set to 0.0002, while 0.002 for linear fine-tuning. In the fine-tuning phase, the common data augmentation strategies are adopted, such as color jittering, random cropping and random horizontal flip.

Note: In ablation studies, unless otherwise specified, our default settings are marked gray in tables and figures.

A. Understanding ParamCrop

In this section, we first visualize the curve of spatio-temporal IoU and 3D center Manhattan distance between two cropped regions along with the contrastive training in Fig. 5 and show two properties of ParamCrop, before we analyze the source of improvement of ParamCrop in Table I.

Property 1: ParamCrop gradually increases the disparity between views: For random cropping, the average distance between two views does not change much with the training process, which indicates the distribution of disparity between views keeps almost unchanged. For ParamCrop, the disparity gradually increases: at the initialization stage, the cropped cubes share a large portion of common visual contents (Fig. 5(a1), (a2)); with the gradual increase in the center distance and decrease in the IoU, the amount of shared content gradually decreases, indicating the increase of disparity.

Property 2: Early stopping ensures reasonable overlap between views: Without the early stopping strategy, the maximum distance is quickly reached after a short training (diagonal locations for two views in Fig. 5(b1)), before the distance reduces to 0.75 and oscillates around it. This is probably because the learning process without early stopping has a strong momentum, but a distance around 0.75 is sufficiently large to generate two views with no overlap at late stage of training, as in Fig. 5(c1). However, ParamCrop with early stopping can prevent the extreme locations and box sizes, which ensures enough shared semantic information between the two views, as in Fig. 5(b2), (c2).

Source of improvement analysis: To investigate the improvement brought by ParamCrop, we compare the performance of the cropping strategies without and with curriculum learning in Tables I and II, respectively.
In Table I, ‘AutoAugment’ means the augmentation strategy found by [25]; ‘Simple’ indicates much shared contents as in Fig. 5(a1); ‘Hard’ refers to less shared contents as in Fig. 5(b1) and (c1). Please note that the training difficulty of these strategies will not change with the training process. One can observe that AutoAugment seems to be not more beneficial than random cropping to the contrastive representation learning, as its strategy is suited specifically for supervised classification. Fixed disparity (i.e., Simple and Hard) between views decreases the quality of learned representation, regardless of its difficulty.

When combining ParamCrop with other augmentations, the heavily biased augmentations (e.g., AutoAugment and Hard) still show poor performances. Since the biased augmentations may generate two cropped views with less shared contents, further applying ParamCrop cannot introduce more shared contents. However, the two cropped views generated by ‘Simple’ and ‘RandomCrop’ usually share more contents. Therefore, ParamCrop can attempt to increase the learning difficulty by gradually reducing the overlaps between two cropped views. Further, the ‘RandomCrop’ can provide some additional uniform randomness, which enriches the diversity of the views generated by ParamCrop. Therefore, ParamCrop with RandomCrop achieves best accuracy and surpasses ParamCrop with AutoAugment by 3.2% on HMDB51 under linear evaluation. The results demonstrate that ParamCrop shows notable advantages over the existing data augmentations.

We next compare ParamCrop with various curriculum-based cropping strategies in Table II, which can be divided into four groups:

1) Increasing the cropping difficulty in the latter training stage, such as ‘Random Crop+++,’ etc;
2) Three different curriculum learning functions (i.e., $x$, $x^2$, and $\sqrt{T}$) to control the cropping from easy to hard;
3) Manual simulation: We define a MSigmoid($x$) function to approximate the red curve in Fig. 5, which can be written as:

$$\text{MSigmoid}(x) = \frac{\alpha}{1 + e^{-(x-\beta)\gamma}} + \tau, \quad (9)$$

where $\alpha$, $\beta$, $\gamma$ and $\tau$ control the vertical scale, horizontal position, slope and the minimal value of MSigmoid($x$), respectively. By simply adjusting these parameters, we can obtain a curriculum learning curve close to ParamCrop. In training, we replace the MLP in ParamCrop with MSigmoid($x$) to generate cropping parameters.

4) We replace the MLP in ParamCrop with a lightweight ResNet-18. In this way, ParamCrop can receive the frames as input to generate input-dependent cropping parameters.

As in Table II, training easy samples first and then difficult samples (such as ‘Random Crop++’) can improve representations to a certain extent on HMDB51, but the performances are still much weaker than ParamCrop. Furthermore, the curriculum function, i.e., $\sqrt{T}$, achieves better performances, which implies that a reasonable curriculum cropping strategy is crucial for video representations. Next, if we manually simulate the cropping process of ParamCrop with a near-sigmoid function, we can observe a notable improvement from Random Crop in both fully (∼1% on average) and linear finetuning (∼1.5% on average). This shows the benefit of gradually increasing the disparity between cropped views. The manual simulation with Early Stopping is the closest to our ParamCrop and achieves considerable improvements. However, the manual simulation is still unable to perceive the training state of the backbone, therefore, the performance is slightly weaker than ParamCrop around 1% on both datasets. Over the above variants, ParamCrop can achieve better performances, mainly because the training process not only increases the disparity gradually for curriculum learning, but is also controlled adaptively by the training process indicated by the contrastive loss, thus leading to a better adaptive path than hand-crafted ones.

Finally, comparing input-independent ParamCrop with input-dependent ParamCrop, the latter damages the performances up to 3.6% and 3.1% on HMDB51 and UCF101, respectively. By observing the training curve, it can be seen that the training curve of input-dependent ParamCrop is more unstable than input-independent ParamCrop, which indicates that the video backbone struggles to adapt to unstable cropping areas, resulting in unsatisfactory representation quality.

### B. Ablation Studies

**Spatial cropping and temporal cropping:** We decompose ParamCrop into the parametric spatial cropping and the parametric temporal cropping, and evaluate them independently by fixing the other to central crop. The results are shown in Table III. First, the parametric temporal cropping yields a slightly lower performance than random cropping under FT and LFT. Since only temporal variants with more shared contents may mislead the video contrastive learning to overfit simple backgrounds, while the introduction of random cropping into spatial dimension can alleviate this. The parametric temporal cropping yields gains of 1.5% and 0.4% on HMDB51 and UCF101, respectively. Second, the parametric spatial cropping shows better performance than random cropping under FT, which indicates that spatial cropping can guide the models to learn better initialization. However, the linear evaluation of the only-spatial ParamCrop is still poor due to the inability to benefit from various temporal variants in videos. Intriguingly, the additional temporal random cropping can also alleviate this problem and demonstrate the effectiveness of the spatial parametric cropping. The

| TABLE III | DECOMPOSING PARAMCROP (PC.) INTO SPATIAL (SPAT.) AND TEMPORAL CROPPING (TEMP.) |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|                | Temp. | Spat. | R.C. | P.C. | R.C. | P.C. | FT | LFT | FT | LFT |
|                |       |       |      |      |      |      |     |     |     |     |
|                | ✓     | ✓     | ✓    | ✓    | ✓    | ✓    | ✓   | ✓   | ✓   | ✓   |
|                | ✓     | ✓     | ✓    | ⨁   | ✓    | ✓    | ✓   | ⨁   | ✓   | ✓   |
|                | ✓     | ✓     | ⨁   | ⨁   | ✓    | ✓    | ✓   | ⨁   | ✓   | ✓   |
|                | ✓     | ✓     | ⨁   | ⨁   | ✓    | ✓    | ✓   | ⨁   | ✓   | ✓   |

*R.C.* indicates the usage of random cropping in $R_1$, $R_2$, for the combination of random cropping and paramcrop, we insert random cropping before paramcrop.
Spatial aspect ratio and rotation: Adjusting spatial aspect ratio and rotation angle can be realized by affine transformation, which are controlled by the region scale $s_p$ and rotation angle $\theta$ in transformation matrix. However, as shown in Table IV, introducing aspect ratio and rotation to ParamCrop has an adverse effect on the representation. Since there is less abnormal scaling and rotation in natural videos. This is in line with previous research findings [1]. Hence, we disable aspect ratio learning and set the rotation angle $\theta$ and $\tilde{\theta}$ both to 0.0 in other experiments.

Gradient reversal: Table V shows the importance of gradient reversal for ParamCrop. Removing gradient reversal makes the objective of ParamCrop identical to the backbone, i.e., to minimize the contrastive loss. Hence, in this case, ParamCrop tries to increase the amount of shared visual contents between two spatio-temporal cubes cropped by the cropping module, which encourages the model to find shortcuts, yielding sub-optimal representations.

Early stopping: Qualitatively, it is observed in Fig. 5(a2-b2-c2) that early stopping avoids extreme locations (compared to Fig. 5(a1-b1-c1) with no early stopping), which ensures shared contents for cropped regions. Quantitatively, the results in Table V shows that early stopping notably increases the linear separability of the learned representation, while it has less effect on the fully fine-tuning.

Sensitivity analysis of the detach bound $b_{\text{stop}}$: As in Fig. 6, for fully fine-tuning (FT), when $b_{\text{stop}} \leq 0.2$, the performances of FT on HMDB51 and UCF101 fluctuate slightly by 0.1% and 0.5%, respectively. Then because of the decreased disparity, the performance of FT gradually drops with the increase of the detach bound. For LFT, since there exists a sweet spot of the intensity of the augmentations in contrastive learning for the representation quality [23], the performance of linear evaluation gradually increases before it drops. To strike the balance between linear and full evaluations, we choose 0.2 as our detach bound.

Sensitivity analysis for cropping scale: To study how the hyper-parameters $s_p$, $s_t$, $\tilde{s}_p$ and $\tilde{s}_t$ in (6) affect the performance of ParamCrop, sensitivity tests are performed for both HMDB51 and UCF101 in Figs. 7 and 8. In these experiments, we use the equal spatial and temporal scale for cropping, i.e., $s_p = s_t = \hat{s}$ and $\tilde{s}_p = \tilde{s}_t = \hat{s}$. It can be observed that the representations learned by ParamCrop achieves best generalization ability with $\hat{s} = 0.5$ and $\hat{s} = 1.0$. From the impact of FT, the change of $\hat{s}$ and $\hat{s}$ near the default values only brings around 0.5% fluctuation on HMDB51 and UCF101. In fact, decreasing cropping scales will significantly reduce the shared visual cues between two cropped views, which is too difficult for the network to map them to a shared feature embedding. Conversely, increasing $\hat{s}$ may make the network find the shortcuts easier since the two cropped regions tend to be similar. Therefore, for LFT, $\hat{s}$ can achieve the best performance with $\hat{s} = 0.5$ on both HMDB51 and UCF101. In addition, $\hat{s}$ is positively correlated with $\hat{s}$ on the two down-stream datasets. We can summarize that the effects of $\hat{s}$ and $\hat{s}$ are clear and significant on both FT and LFT.

Different frameworks and backbones: To evaluate the applicability of the proposed approach, ParamCrop is integrated to two mainstream contrastive learning frameworks SimCLR [1] and MoCo [2] and two common video backbones 3D-G [58] and R-2D3D [59]. The results in Table VI demonstrate that ParamCrop can be generalized to multiple contrastive frameworks and video backbones.

### Table IV
Ablation on the Aspect Ratio (A.R.) and Rotation (Rot.) in ParamCrop

| Method    | A. R. | Rot. | HMDB51 FT | LFT | UCF101 FT | LFT |
|-----------|-------|------|-----------|-----|-----------|-----|
| RandomCrop | ✓     | ✗    | 56.0      | 33.5 | 85.3      | 57.9 |
| ParamCrop | ✓     | ✓    | 55.7      | 34.7 | 84.5      | 53.6 |
| ParamCrop | ✗     | ✓    | 55.8      | 29.5 | 86.2      | 48.6 |
| ParamCrop | ✗     | ✗    | 59.9      | 37.3 | 86.9      | 59.3 |

### Table V
Ablation on the Early Stopping (E.S.) Strategy and Gradient Reversal (G.R.) in ParamCrop

| Method    | E.S. | G.R. | HMDB51 FT | LFT | UCF101 FT | LFT |
|-----------|------|------|-----------|-----|-----------|-----|
| RandomCrop | -    | -    | 56.0      | 33.5 | 85.3      | 57.9 |
| ParamCrop | ✓    | ✓    | 56.0      | 33.2 | 84.3      | 55.2 |
| ParamCrop | ✗    | ✓    | 59.9      | 32.6 | 87.0      | 55.8 |
| ParamCrop | ✓    | ✗    | 59.9      | 37.3 | 86.9      | 59.3 |

### Table VI
Integrating ParamCrop (P.C.) with Different Contrastive Frameworks and Video Backbones. The Fully Fine-Tuning Performances are Reported Here.

| Framework | Backbone | ParamCrop | HMDB51 | UCF101 |
|-----------|----------|-----------|--------|--------|
| SimCLR    | S3D-G    | ✗         | 56.0   | 85.3   |
| SimCLR    | S3D-G    | ✓         | 59.9   | 86.9   |
| SimCLR    | R-2D3D   | ✗         | 50.4   | 77.2   |
| SimCLR    | R-2D3D   | ✓         | 53.0   | 79.4   |
| MoCo      | S3D-G    | ✗         | 52.4   | 84.1   |
| MoCo      | S3D-G    | ✓         | 54.3   | 85.1   |
| MoCo      | R-2D3D   | ✗         | 45.4   | 72.8   |
| MoCo      | R-2D3D   | ✓         | 48.2   | 73.8   |

### Table VII
Evaluating Random Cropping (RandomC.) and ParamCrop (ParamC.) by Fully Fine-Tuning Pre-Trained Models With Less Labelled Data on HMDB51 and UCF101

| Backbone | Label | HMDB51 RandomC. | ParamC. | UCF101 RandomC. | ParamC. |
|----------|-------|-----------------|---------|-----------------|---------|
| S3D-G    | 50%   | 51.5            | 52.8    | 82.4            | 83.3    |
|          | 30%   | 44.9            | 46.0    | 76.9            | 78.5    |
|          | 10%   | 33.3            | 34.6    | 57.6            | 60.1    |
**TABLE VIII**

| Backbone | CroppingMethod | 10% Label | 50% Label |
|----------|----------------|-----------|-----------|
| S3D-G    | RandomCrop    | 13.9      | 29.9      |
|          | ParamCrop     | 14.4      | 31.6      |

All experiments are based on the mean teacher [64] framework on hmdb51 dataset, and the models are initialized randomly.

**TABLE IX**

| Backbone | Layers | P.S. | C.C.(Flops) | HMDB51 | UCF101 |
|----------|--------|------|-------------|--------|--------|
| S3D-G    | 0      | 10.3M | 71.9G       | 56.0   | 85.3   |
|          | 1      | +396  | +0.002G     | 57.7   | 86.6   |
|          | 2      | 0.005M| +0.002G     | 59.9   | 86.9   |
|          | 3      | 0.014M| +0.002G     | 58.0   | 86.8   |

+" means the absolute increase from baseline. The fully fine-tuning accuracy is reported here.

**Data efficiency:** Table VII shows that as we reduce the number of labelled training data during fine-tuning, ParamCrop consistently outperforms random cropping under variant label ratios, which shows that the representation yielded by ParamCrop has a higher data efficiency.

**Semi-supervised learning:** As a popular method in the semi-supervised task, Mean Teacher [64] also contains a contrastive learning branch similar to MoCo [2]. Table VIII shows ParamCrop can also replace the random cropping in the semi-supervised settings to enhance the performance.

**Parameters and computations:** The additional parameters and calculation overhead introduced by ParamCrop are shown in Table IX. The ParamCrop only contains an MLP, which can be negligible for S3D-G both parameters and computations. Notably, the ablations for the layers of MLP show that a two-layer MLP achieves best performance compared with other settings. Since the one-layer MLP can result in poor nonlinear ability and the MLP with more layers may involve optimization difficulties.

**Training epochs:** The ablation studies are pre-trained only 20 epochs. To verify whether ParamCrop will fail after long-term training, we provide performance comparisons between ParamCrop and random crop counterpart with different training periods in Figs. 9 and 10. Overall, a longer training period gives a better classification accuracy in the downstream tasks under both fully fine-tuning and linear evaluation. In comparison with random cropping, with the extension of training time, ParamCrop achieves consistent improvements on both HMDB51 and UCF101.

**Visualizations:** To better comprehend what views are generated by ParamCrop, we visualize four cases comparing random cropping and different variants of ParamCrop in Fig. 11, i.e., (a) random cropping, (b) ParamCrop without gradient reversal, (c) ParamCrop without early stopping, and (d) ParamCrop. Figures show that: (a) Random crop generates views that are uniform distributed in the spatial and temporal dimension, and the visual content in both cropped views are shared to a large extent. (b) ParamCrop without gradient reversal always outputs all spatial contents for two views, as this variant aims to minimize contrastive loss by preserving more visual cues shared by two augmented views. (c) ParamCrop without early stopping convergences to extreme regions and generates completely different views. Although this strategy can learn a good initialization for backbone, it will hurt the performance of linear evaluation. (d) The two cropped views generated by ParamCrop are different, but the ratio of the shared content between two views is relatively increased compared to no early stopping and decreased compared to no gradient reversal and random cropping.

C. Comparison With the Existing Approaches

**Video Retrieval:** In the video retrieval task, we mainly follow the settings in previous works [8], [9], [17]. The models pre-trained by ParamCrop on Kinetics400 using SimCLR are employed as the feature extractor without fine-tuning. We conduct experiments on both HMDB51 and UCF101, and compare with other approaches in Tables X and XI respectively. Results show that ParamCrop exceeds the MemDPC [17] by 6.4% and 2.8% with the same backbone on HMDB51 and UCF101, respectively, which indicates that our learned representation is more generalized.

**Action Recognition:** Table XII compares ParamCrop existing methods under linear fine-tuning and fully fine-tuning on HMDB51 and UCF101. In these experiments, we choose SimCLR [1] as the base contrastive learning framework. From linear evaluation results, we can observe that compared with MemDPC [17], ParamCrop respectively gains 6.1% and 8.3% on HMDB51 and UCF101 using R-2D3D, demonstrating ParamCrop can learn powerful video representations. For fully fine-tuning results, we have the following observations: (i) ParamCrop achieves competitive performance on the two datasets with
existing approaches. Specifically, we outperform most existing approaches with the same pre-training resolution. Compared to CVRL [21], ParamCrop has a slightly lower performance, which may due to the deeper backbone (i.e., R3D-50) and larger pre-training resolution (i.e., $16 \times 224^2$ compared to our $16 \times 112^2$) used by CVRL. Still, our performance is competitive with CVRL on UCF101; (ii) ParamCrop can be trained on less data but achieve remarkable performance. DynamoNet [10] is trained with 8 M videos on Youtube8 M, while ParamCrop just uses 240 K videos but still obtains 2.4% and 0.8% gains.

V. CONCLUSION

In this work, we propose a parametric cubic cropping for adaptively controlling the disparity between two cropped views along the training process. Specifically, we enable online training by first extending the affine transformation matrix to the 3D affine transformation and learn to regress the transformation parameters such that the cropping operation is fully differentiable. For the optimization, the parametric cubic cropping operation is trained with an adversarial objective to the video backbone, and optimized using a simple gradient reversal operation. We additionally show that an early stopping strategy in the optimization process is beneficial for the video backbone to learn more robust representations. Empirical results demonstrate that the adaptively controlled disparity between views is indeed effective for improving the representation quality. Extensive ablation studies validate the effectiveness of each proposed component and evaluations are performed on both action recognition and video retrieval.

REFERENCES

[1] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.
[2] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.
[3] M. Caron et al., “Unsupervised learning of visual features by contrasting cluster assignments,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 9912–9924.
[4] J.-B. Grill et al., “Bootstrap your own latent-a new approach to self-supervised learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 21271–21284.
[5] X. Chen and K. He, “Exploring simple siamese representation learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 15750–15758.

[6] L. Jing and Y. Tian, “Self-supervised spatiotemporal feature learning by video geometric transformations,” 2018, arXiv:1811.11387.

[7] D. Kim, D. Cho, and I. S. Kweon, “Self-supervised video representation learning with space-time cubic puzzles,” in Proc. AAAI Conf. Artif. Intell., 2019, vol. 33, pp. 8545–8552.

[8] D. Xu et al., “Self-supervised spatiotemporal learning via video clip order prediction,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 10334–10343.

[9] D. Luo et al., “Video cloze procedure for self-supervised spatio-temporal learning,” in Proc. AAAI Conf. Artif. Intell., 2020, vol. 34, pp. 11701–11708.

[10] A. Diba, V. Sharma, L. V. Gool, and R. Stiefelhagen, “Dynamonet: Dynamic action and motion network,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 6192–6201.

[11] S. Benaim et al., “Speednet: Learning the speediness in videos,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 9922–9931.

[12] I. Misra, C. L. Zitnick, and M. Hebert, “Shuffle and learn: Unsupervised learning using temporal order verification,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 527–544.

[13] D. Zhukov, J.-B. Alayrac, I. Laptev, and J. Sivic, “Learning actionness via long-range temporal order verification,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 470–487.

[14] H.-Y. Lee, J.-B. Huang, M. Singh, and M.-H. Yang, “Unsupervised representation learning by sorting sequences,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 667–676.

[15] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, “Unsupervised feature learning via non-parametric instance discrimination,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 3733–3742.

[16] T. Han, W. Xie, and A. Zisserman, “Video representation learning by dense predictive coding,” in Proc. IEEE Int. Conf. Comput. Vis. Workshops, 2019.

[17] T. Han, W. Xie, and A. Zisserman, “Memory-augmented dense predictive coding for video representation learning,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 312–329.

[18] J.-B. Alayrac et al., “Self-supervised multimodal versatile networks,” Adv. Neural Inf. Process. Syst., vol. 33, pp. 25–37, 2020.

[19] K. Hu et al., “Contrast and order representations for video self-supervised learning,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 7939–7949.

[20] C. Feichtenhofer, H. Fan, B. Xiong, R. Girshick, and K. He, “A large-scale study on unsupervised spatiotemporal representation learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 3299–3309.

[21] R. Qian et al., “Spatiotemporal contrastive video representation learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 6964–6975.

[22] Y. B. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in Proc. Eur. Conf. Comput. Vis., 2012, pp. 725–742.

[23] Y. B. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning by sorting sequences,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2012, pp. 7939–7949.

[24] Y. Liu, K. Wang, L. Liu, H. Lan, and L. Lin, “TCCGL: Temporal contrastive graph for self-supervised video representation learning,” IEEE Trans. Image Process., vol. 31, pp. 1978–1993, 2022.

[25] Y. N. D. D. L.-P. H. Zhang and M. Cisse, “mixup: Beyond empirical risk minimization,” in Proc. IEEE Int. Conf. Comput. Vis., 2018, pp. 527–544.

[26] J. Wang, G. Bertasius, D. Tran, and L. Torresani, “Long-shot temporal contrastive video learning for video transformers,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 14010–140 20.

[27] Y. Zhang et al., “Contrastive spatio-temporal pretext learning for self-supervised video representation,” in Proc. AAAI Conf. Artif. Intell., 2022, vol. 36, pp. 3380–3389.

[28] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive prediction,” in Proc. Int. Conf. Learn. Representations, 2018.

[29] J. Park, J. Lee, I.-J. Kim, and K. Sohn, “Probabilistic representations for video contrastive learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 14711–14721.

[30] Y. Liu, K. Wang, L. Liu, H. Lan, and L. Lin, “TCGL: Temporal contrastive graph for self-supervised video representation learning,” IEEE Trans. Image Process., vol. 31, pp. 1978–1993, 2022.

[31] S. Giradis, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” in Proc. Int. Conf. Learn. Representations, 2018.

[32] M. Noroozi and P. Favaro, “Unsupervised learning of visual representations by solving jigsaw puzzles,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 69–84.

[33] H.-Y. Lee, J.-B. Huang, M. Singh, and M.-H. Yang, “Unsupervised representation learning by sorting sequences,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 667–676.

[34] Y. Zhang et al., “Contrastive spatio-temporal pretext learning for self-supervised video representation,” in Proc. AAAI Conf. Artif. Intell., 2022, vol. 36, pp. 3380–3389.

[35] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” 2018, arXiv:1807.03748.

[36] P. Chen et al., “Rspnet: Relative speed perception for unsupervised video representation learning,” in Proc. AAAI Conf. Artif. Intell., vol. 35, no. 2, pp. 1045–1053, 2021.

[37] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, “Random erasing data augmentation,” in Proc. AAAI Conf. Artif. Intell., 2020, vol. 34, pp. 13001–13008.

[38] Y. N. D. L.-P. H. Zhang and M. Cisse, “mixup: Beyond empirical risk minimization,” in Proc. Int. Conf. Learn. Representations, 2018.

[39] D. Hendrycks et al., “AugMix: A simple data processing method to improve robustness and uncertainty,” in Proc. Int. Conf. Learn. Representations, 2020.

[40] Y. Li et al., “Differentiable automatic data augmentation,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 580–595.

[41] A. Tamkin, M. Wu, and N. Goodman, “Viewmaker networks: Learning views for unsupervised representation learning,” in Proc. Int. Conf. Learn. Representations, 2021.

[42] X. Peng, K. Wang, Z. Zhu, M. Wang, and Y. You, “Crafting better contrastive views for siamese representation learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 16031–16040.

[43] M. Jaderberg et al., “Spatial transformer networks,” in Proc. Adv. Neural Inf. Process. Syst., 2015, vol. 28.
[57] J. Carreira and A. Zisserman, “Quo vadis, action recognition? A new model and the kinetics dataset,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 6299–6308.

[58] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, “Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 305–321.

[59] D. Tran et al., “A closer look at spatiotemporal convolutions for action recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 6450–6459.

[60] Y. You, I. Gitman, and B. Ginsburg, “Large batch training of convolutional networks,” 2017, arXiv:1708.03888.

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

[62] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “Hmdb: A large video database for human motion recognition,” in Proc. IEEE Int. Conf. Comput. Vis., 2011, pp. 2556–2563.

[63] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. Int. Conf. Learn. Representations, 2015.

[64] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in Proc. Adv. Neural Inf. Process. Syst., 2017, vol. 30.

[65] J. Wang et al., “Enhancing unsupervised video representation learning by decoupling the scene and the motion,” Proc. AAAI Conf. Artif. Intell., vol. 35, no. 11, pp. 10 129–10 137, 2021.

[66] Q. Kong, W. Wei, Z. Deng, T. Yoshinaga, and T. Murakami, “Cycle-contrast for self-supervised video representation learning,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 8089–8100.

[67] C. Sun, F. Baradel, K. Murphy, and C. Schmid, “Learning video representations using contrastive bidirectional transformer,” 2019, arXiv:1906.05743.

[68] R. Qian et al., “Enhancing self-supervised video representation learning via multi-level feature optimization,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 7990–8001.

[69] J. Wang et al., “Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 4006–4015.

[70] L. Jing, X. Yang, J. Liu, and Y. Tian, “Self-supervised spatiotemporal feature learning via video rotation prediction,” 2018, arXiv:1811.11387.

[71] J. Wang et al., “Self-supervised video representation learning by uncovering spatio-temporal statistics,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 7, pp. 3791–3806, Jul. 2022.