ViewCLR: Learning Self-supervised Video Representation for Unseen Viewpoints

Srijan Das
University of North Carolina at Charlotte
sdas24@uncc.edu

Michael S. Ryoo
Stony Brook University
mryoo@cs.stonybrook.edu

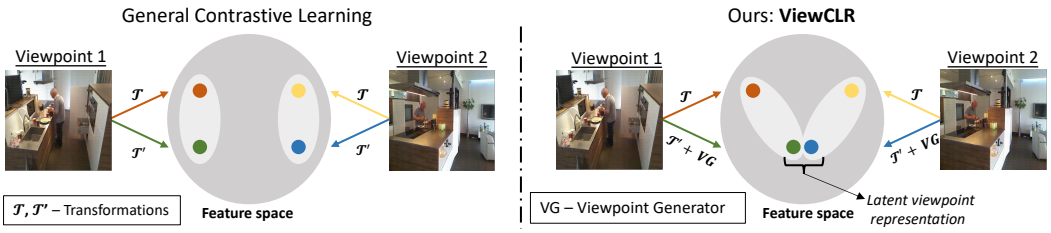

Figure 1: An illustration of how contrastive learning with standard data augmentations ($\mathcal{T}$ and $\mathcal{T}'$) embeds two video clips of the same action captured from different camera angles in the feature space versus the feature space representation of ViewCLR. Augmentation through Viewpoint Generator (VG) induces similarities among the video clips by generating latent viewpoint representation in the feature space.

Abstract

*Learning self-supervised video representation predominantly focuses on discriminating instances generated from simple data augmentation schemes. However, the learned representation often fails to generalize over unseen camera viewpoints. To this end, we propose **ViewCLR**, that learns self-supervised video representation invariant to camera viewpoint changes. We introduce a viewpoint-generator that can be considered as a learnable augmentation for any self-supervised pre-text tasks, to generate latent viewpoint representation of a video. ViewCLR maximizes the similarities between the representation of the latent viewpoint and that of the original viewpoint, enabling the learned video encoder to generalize over unseen camera viewpoints. Experiments on cross-view benchmark datasets including NTU RGB+D dataset show that ViewCLR stands as a state-of-the-art viewpoint invariant self-supervised method.*

1. Introduction

Video understanding has taken a new stride with the advancements of 3D CNNs [15, 5, 48]. But one major limitation of CNNs is that they are unable to recognize samples out of training distribution. For instance, if we train an action classifier with videos acquired from one camera viewpoint and test the learned model on videos from a different camera viewpoint, the model drastically fails to recognize. This issue persists in a greater extent while learning self-supervised video representation expecting the learned encoder to generalize over a large diversity of viewpoints.

Self-supervised Learning (SSL) has been very successful with the use of instance discrimination through contrastive learning [13]. The concept of contrastive learning is based on maximizing similarities of positive pairs while minimizing similarities of negative pairs. It is to be noted that the terminology 'views' refer to the version of data obtained through data augmentation whereas 'viewpoints' refer to the data acquired from different camera angles. Learning invariance to different views is important in contrastive learning. These views can be generated through data augmentation like random cropping, Gaussian blurring, rotating inputs etc. Towards self-supervised video representation, augmentation by tampering the temporal segments in a video is explored to learn invariance across the temporal dimension in videos [21, 22, 12, 36, 33]. However, these pretext tasks and the associated augmentations are not designed to encode viewpoint invariant characteristics to the learned video encoder.

In this paper, we focus on learning self-supervised video representation that generalizes over unseen camera viewpoints. We do not aim at designing a new pretext task suitable for a specific downstream scenario but instead propose a module that can be incorporated with the existing self-

supervised methods. Several methods have been proposed in the literature to address the challenge of camera view invariant features, mostly using 3D Poses [24, 30, 61]. These poses provide geometric information which are robust to camera viewpoint changes. The availability of large scale 3D Poses [41] have facilitated the research community to propose unsupervised skeleton representations [26, 27, 34]. However, the use of 3D Poses are limited to indoor scenarios and most importantly lacks encoding the appearance information. Thus, in this paper, we aim at learning viewpoint invariant features with RGB input in order to generalize SSL for real-world applications.

General contrastive learning methods using standard data augmentation schemes are not explicitly designed to encourage instances from the same class (for example, similar actions) but different camera viewpoints to pull closer to each other in the feature space as illustrated in Fig 1. To this end, we propose **ViewCLR** that provides a learnable augmentation to induce viewpoint changes while learning self-supervised representation. This is achieved by a viewpoint-generator (VG) that learns latent viewpoint representation of a given feature map by imposing the features to follow 3D geometric transformations and projections. However, these transformations and projections are learned by minimizing the contrastive loss. This constraint is achieved by performing a mixup between the features learned by the encoder and the latent viewpoint representation in the manifold space. The outcome is a trained video encoder that takes into account the latent viewpoint representation of the videos, while maximizing its similarities with representation from the original camera viewpoint. As shown in Fig. 1, the latent viewpoint representation of the videos enables ViewCLR to pull representations from similar classes but different camera angles closer to each other.

We demonstrate the effectiveness of ViewCLR by evaluating the learned representations for the task of action recognition. Our experimental analysis shows that ViewCLR significantly improves the action classification performance with regards to generalizing over unseen videos captured from different camera angles. On popular multi-view datasets like NTU RGB+D and NUCLA, ViewCLR with self-supervised pre-training performs on par with the supervised models pre-trained with huge video datasets. This observation substantiates the importance of learning representations that are invariant to camera angles which is crucial for real-world video analysis.

2. Background: Neural Projection Layer

In this section, we recall a recently introduced algorithm Neural Projection Layer (NPL), which learns a latent representation of different camera viewpoint for a given action in supervised settings [38]. Our ViewCLR is a spiritual successor of NPL for learning self-supervised viewpoint invariant video representation. NPL is derived from the standard 3D geometric camera model used in computer vision. NPL learns a latent 3D representation of actions and its multi-view 2D projections. This is done by imposing the latent action representations to follow 3D geometric transformations and projections, in addition to minimizing the cross-entropy to learn the action labels. First, the feature map $F \in \mathcal{R}^{c\times m\times n}$ which is an intermediate representation of an image of dimension $M \times N$ is fed to a CNN that estimates the 3D space coordinates $p_{x,y}$ for each pixel in F. Also, a fully-connected layer is used to estimate the transformation matrices - rotation and translation (R and t) for each image in a video. The learned matrices are used to transform a specific camera view to a 3D world coordinate system as $p^w_{x,y} = p_{x,y} \cdot [R^T|R^T t]$ for each pixel in F. The world 3D representation is then given by:

$$F^W_{x,y,z} = \sum_{i=0,j=0}^{m,n} (1 - |x - p^w_{i,j}[x]|)(1 - |y - p^w_{i,j}[y]|) \\ (1 - |z - p^w_{i,j}[z]|)F^{'}_{i,j} \tag{1}$$

where F' is obtained by concatenating feature map F and $p_{x,y}$ across channels.

Next, the world feature representation F^W is projected back to 2D. This is done by estimating a camera matrix K, which is given by

$$K = R \begin{pmatrix} s_x & 0 & x_0 \\ 0 & s_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2}$$

where (s_x, s_y) and (x_0, y_0) are the scaling factors and the offsets respectively. R here is a 3×3 camera rotation matrix which is derived from a set of learned parameters. Thus, the 2D projection of the 3D points is estimated as

$$F^p_{c,x,y} = \sum_{i=0,j=0}^{m,n} (1 - |x - Kp^w_{i,j}[x]|) \\ (1 - |y - Kp^w_{i,j}[y]|)F^{'}_{c,i,j} \tag{3}$$

These frame-level operations are performed across the temporal dimension of a video to compute the viewpoint invariant representation of a video. In addition to learning the action labels, NPL is constrained over a 3D loss $\mathcal{L}_{3D}$ as

$$\mathcal{L}_{3D}(V, U) = ||F^W(V) - F^W(U)||_F \tag{4}$$

where two videos U and V belong to the same action class. This loss encourages the representations from different viewpoints of the same action result in the same 3D representation. However, learning such viewpoint invariant latent representation with NPL is difficult in the absence of action labels. In ViewCLR, we adopt strategies to learn such latent 3D representation even without the need of human annotated action labels.

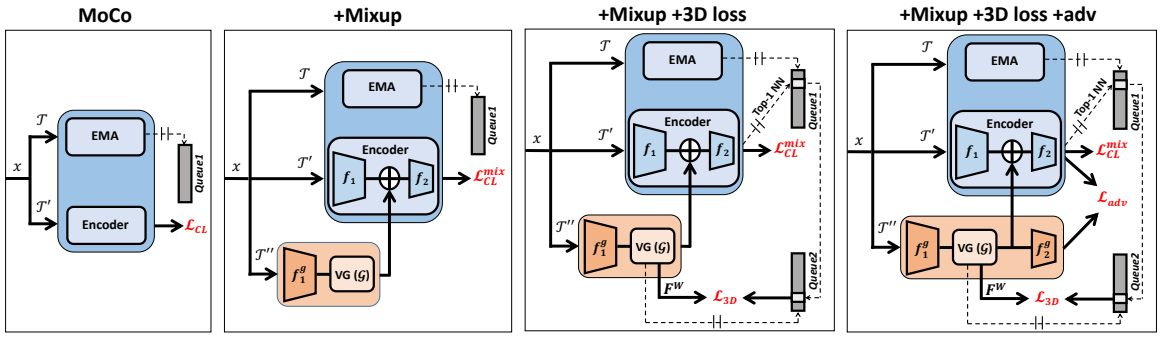

Figure 2: Illustration of each component in ViewCLR. The input sample x is a video clip. First, the MoCo framework with an encoder and EMA (momentum encoder) is presented at the left. Second, we present ViewCLR with the mixup contrastive loss $\mathcal{L}_{CL}^{\text{mix}}$. Then, the viewpoint-generator (VG) with the 3D loss $\mathcal{L}_{3D}$ is presented. The world 3D representations F^W are encoded in Queue2. Top-1 NN of $f(x)$ in Queue1 is selected from Queue2 for computing the 3D loss with F^W. For brevity, we ignore the auto-encoder representation of F^W in this figure. Finally, we present ViewCLR with all its components, including the adversarial loss $\mathcal{L}_{adv}$.

3. ViewCLR

In this section, we describe our proposed ViewCLR to learn self-supervised video representation such that the learned representation is robust to different viewpoints. For self-supervised representation learning, we use instance discrimination approach proposed in MoCo [17]. We remind the readers that the terminology 'views' indicates different data views obtained through data augmentation whereas 'viewpoints' refer to the videos captured from different camera angles.

3.1. MoCo

To formulate, given a video x, a set of augmentation transformations $\mathcal{T}$ and $\mathcal{T}'$ is applied on x to generate its two views. Note that these views of the same video are results of standard augmentations and do not involve generating different camera viewpoints. Given a video encoder $f(\cdot)$, contrastive loss like InfoNCE [13] maximizes the similarity of a video sample $f(x)$ with positive ones k_+ and minimizes similarity to negative ones k_-. MoCo uses an explicit momentum updated version (EMA) of encoder $f(\cdot)$ to compute the embeddings k_+ of video x, and negative embeddings k_- are encoded in a dictionary queue referred to as Queue1. Therefore, the InfoNCE loss is formulated as

$$\mathcal{L}_{CL} = -\log \frac{\exp(f(x) \cdot k_+/\tau)}{\sum_{i=0}^{\mathcal{N}} \exp(f(x) \cdot k_i/\tau)} \tag{5}$$

where embeddings $k_i \in \{k_+, k_-\}$ and τ is a scaling temperature parameter. The sum is over one positive and $\mathcal{N}$ negative samples in Queue1. For brevity, we loosely use the same notation for both the augmented version of input x throughout the paper. This framework as shown in Fig. 2, relies solely on standard transformations $\mathcal{T}$ and $\mathcal{T}'$ to learn discriminative video representation. ViewCLR goes one step beyond to generate latent viewpoint representation of the input videos to learn representation invariant to camera angles. This is achieved by invoking a viewpoint-generator that projects a viewpoint of a video to another arbitrary viewpoint. The question remains, *how do we use such a generator in the MoCo framework*?

3.2. Viewpoint Generator

In this section, we describe our proposed viewpoint-generator $\mathcal{G}$ whose working principle is similar to that of NPL but adopted for unsupervised setting. First, we justify the design choice of architecture for ViewCLR. Note that we aim at training an encoder with contrastive loss that learns viewpoint invariant representation. The viewpoint-generator is an additional module that can be placed as a block at any intermediate position within the encoder or on top of an encoder. But this design choice would hamper the representation learned by the encoder when we remove the viewpoint-generator for performing downstream tasks. Therefore, ViewCLR introduces another branch in the MoCo framework which consists of an encoder $f^g(\cdot)$ and the viewpoint-generator $\mathcal{G}$. We decompose the encoder with input x as $f^g(x) = f_2^g(f_1^g(x))$ where $f_1^g(\cdot)$ and $f_2^g(\cdot)$ are parts of the encoder $f^g(\cdot)$. We plug in our viewpoint-generator module $\mathcal{G}$ with parameters θ_g in this stream.
In ViewCLR, first an augmentation transformation $\mathcal{T}''$ of the same video x is fed to the partial encoder $f_1^g(\cdot)$. Assuming that we have a viewpoint-generator that can compute latent viewpoint representation of a given feature map, $f_1^g(x)$ is then fed to the viewpoint-generator $\mathcal{G}$. The output of this module $\mathcal{G}(f_1^g(x))$ is a representation of the video projected in an arbitrary latent viewpoint. The idea is to utilize this representation to train a viewpoint invariant video encoder, so we infuse this feature $\mathcal{G}(f_1^g(x))$ into the MoCo framework by performing a Mixup [23] operation in the manifold space.

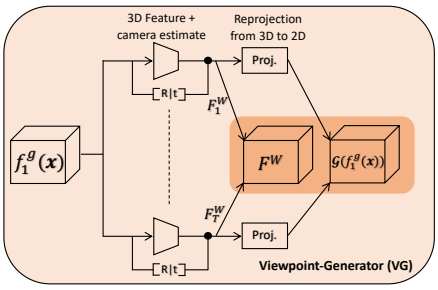

Figure 3: Learning latent viewpoint representation through Viewpoint-Generator. It is a zoom of VG($\mathcal{G}$) presented in Fig. 2.

(I) Mixup for infusing the latent viewpoints. Earlier data mixing strategies [60, 50, 58, 59] have shown that mixing two instances enforces a model to learn discriminative features by providing relevant contextual information. In ViewCLR, we perform mixup between the output feature map of the viewpoint-generator and the intermediate feature map of the encoder $f(\cdot)$. We use this strategy to infuse the output of viewpoint-generator to the MoCo framework. We perform the mixup in the manifold space as in [50]. Let $y_i \in \{0,1\}^{\mathcal{B}}$ be the virtual labels of the input x_i and its augmented version in a batch, where $y_{i,i} = 1$ and $y_{i,j\neq i} = 0$. Then, the $(\mathcal{N}+1)-$ way discrimination loss for a sample in a batch is:

$$\mathcal{L}_{CL}(x_i, y_i) = -y_{i,b} \cdot \log \frac{\exp(f(x_i) \cdot k_+/\tau)}{\sum_{j=0}^{\mathcal{N}} \exp(f(x_i) \cdot k_j/\tau)} \quad (6)$$

where b ranges from 1 to $\mathcal{B}$. Thus, the video instances are mixed within a batch for which the loss is defined as:

$$\mathcal{L}_{CL}^{\text{mix}}((x_i, y_i), (x_r, y_r), \lambda) = \mathcal{L}_{CL}(\text{mix}(x_i, x_r; \lambda), Y) \quad (7)$$

where $Y = \lambda y_i + (1-\lambda) y_r$, $\lambda \sim$ Beta(α, α) is a mixing coefficient, $r \sim rand(\mathcal{B})$, and Mix() is the Mixup operator. In ViewCLR, we perform mixup, i.e. simple interpolation of two video cuboids in the feature space such that

$$Mix(x_i, x_r, \lambda) = \lambda f_1(x) + (1-\lambda)\mathcal{G}(f_1^g(x)) \quad (8)$$

where encoder $f(\cdot)$ is decomposed into $f_1(\cdot)$ and $f_2(\cdot)$. The mixed feature is processed by the partial encoder $f_2(\cdot)$ to optimize the mixed contrastive loss presented in equation 7. This is how, we infuse the latent viewpoint representation $\mathcal{G}(f_1^g(x))$ in the encoder $f(\cdot)$ to minimize the contrastive loss as shown in Fig. 2.

(II)Latent 3D representation. The viewpoint-generator is implemented using NPL and is presented in Fig. 3. Here, we extend the notations introduced in section 2. The viewpoint-generator learns the transformation matrices R, t, and the 3D space coordinates $p_{x,y}$ using the temporal slices of the spatio-temporal feature map $f_1^g(x)$ as input frames. The re-projection of each 3D World representation (for example F_1^W) is performed by estimating the camera matrix K within a video. The 2D projected output when combined across all the temporal slices, we obtain the latent viewpoint representation $\mathcal{G}(f_1^g(x))$ for video x. Different from NPL in [38], the viewpoint-generator here learns the transformation matrices to optimize the mixed contrastive loss $\mathcal{L}_{CL}^{\text{mix}}$. The question remains, *how does the viewpoint-generator learns the world 3D representation*? Although we do not have the leverage to use action labels, but we propose to mine positive samples from the dictionary queue (Queue1) that encodes history of embeddings. Inspired from the assumptions in [14, 27, 10], we take into account that the nearest neighbor representation of x in Queue1 belongs to the same action category. Consequently on one hand, we obtain the Top-1 nearest neighbor of $f(x)$ in Queue1 such that

$$NN(f(x), \text{Queue1}) = \arg \min_{q \in \text{Queue1}} ||f(x) - q||_2 \quad (9)$$

On the other hand, we encode the world 3D representation of a video, referred to as F^W in another dictionary queue, namely Queue2. Note that the representation F^W is obtained by combining all the world 3D representation per temporal slices in a video. In order to optimize the memory requirement incurred in storing the world 3D representation in Queue2, we use an auto-encoder with a reconstruction loss to squeeze the $(c+3) \times T \times m \times n$ world 3D representation of the video to a d_{low} dimensional embedding vector (F_{inter}^W). The details of this auto-encoder is provided in the implementation details. Thus, the d_{low} dimensional vector after L2-normalization is enqueued to Queue2 while maintaining consistency with the embeddings in Queue1 (see Fig. 2). Now, we reformulate 3D loss $\mathcal{L}_{3D}$ presented in equation 4 as

$$\mathcal{L}_{3D} = ||F_{inter}^W(x) - \text{Queue2(idx)}||_F \quad (10)$$

where idx $= NN(f(x), \text{Queue1})$ represents the index of the Top-1 nearest neighbor 3D world representation in Queue2.

(III)Adversarial learning of the viewpoint-generator. Although the viewpoint-generator learns relevant transformations and projection, it might be prone to learning viewpoints similar to the original viewpoint of the input video x. Thus, we adopt an adversarial learning to generate different viewpoints as shown in Fig. 2. In contrast to the previous two constraints, here we introduce the partial encoder $f_2^g(\cdot)$. The adversarial learning of viewpoint-generator is achieved by using a Gradient Reversal Layer (GRL) on top of $f_2^g(\cdot)$ to maximize the following

$$\max_{\theta_g} L_{adv} = \mathbb{E}_{x \sim P_x}(|f_2^g(\mathcal{G}(f_1^g(x))) - f(x)|_F) \quad (11)$$

where P_x is the data distribution of x. This adversarial loss maximizes that the distance between the feature represen-

tations $f(x)$ and $f_2^g(\mathcal{G}(f_1^g(x)))$, resulting in generation of complementary camera viewpoints by $\mathcal{G}$.

3.3. Training ViewCLR

Training ViewCLR is quite straightforward. We first train an encoder $f(\cdot)$ using MoCo framework with infoNCE loss for 300 epochs. Then, we introduce the third stream with partial encoders $f_1^g(\cdot)$ and $f_2^g(\cdot)$ initialized with the weights of $f(\cdot)$ and train them altogether with the summation of mixup contrastive loss $\mathcal{L}_{CL}^{\text{mix}}$, 3D Loss $\mathcal{L}_{3D}$, adversarial loss $\mathcal{L}_{adv}$, and the reconstruction loss for another 200 epochs. This two stage learning enables the encoder to select semantically meaningful Top-1 nearest neighbor, hence benefiting the viewpoint-generator to learn high quality 3D world representation.

4. Experiments

In this section, we describe the datasets used in our experimental analysis, implementation details, and evaluation setup. We present ablation studies to illustrate the effectiveness of ViewCLR and also, provide a state-of-the-art comparison.

4.1. Datasets

Our dataset choices are based on multi-camera setups in order to provide cross-view evaluation. So, we do not make use of popular datasets like Kinetics-400 [18], and UCF101 [45] to pre-train ViewCLR as the videos in these datasets do not posses the view-point challenges we are addressing in this paper.

NTU RGB+D (NTU-60 & NTU-120): NTU-60 is acquired with a Kinect v2 camera and consists of 56k video samples with 60 activity classes. The activities were performed by 40 subjects and recorded from 80 viewpoints. For evaluation, we follow the two standard protocols proposed in [41]: cross-subject (CS) and cross-view (CV). NTU-120 is a super-set of NTU-60 adding a lot of new similar actions. NTU-120 dataset contains 114k video clips of 106 distinct subjects performing 120 actions in a laboratory environment with 155 camera views. For evaluation, we follow a cross-subject (CS_1) protocol and a cross-setting (CS_2) protocol proposed in [29].

Northwestern-UCLA Multiview activity 3D Dataset (NUCLA) is acquired simultaneously by three Kinect v1 cameras. The dataset consists of 1194 video samples with 10 activity classes. The activities were performed by 10 subjects, and recorded from three viewpoints. We performed experiments on N-UCLA using the cross-view (CV) protocol proposed in [52]: we trained our model on samples from two camera views and tested on the samples from the remaining view. For instance, the notation $V_{1,2}^3$ indicates that we trained on samples from view 1 and 2, and tested on samples from view 3.

4.2. Implementation Details

For our experiments with ViewCLR, we use 32 RGB frames of resolution 128×128 as input, at 30 fps. For additional data augmentation, we apply clip-wise consistent random crops, horizontal flips, Gaussian blur and color jittering. We also apply random temporal cropping from the same video as used in [14]. For the encoder backbone $f(\cdot)$, we choose S3D [55] architecture. For the third stream, the viewpoint-generator is plugged after 3 blocks in S3D. So, $f_1^g(\cdot)$ stands for the first 3 S3D blocks and $f_2^g(\cdot)$ stands for 2 S3D blocks. The input to the viewpoint-generator $\mathcal{G}(\cdot)$ is a $480 \times 32 \times 28 \times 28$ spatio-temporal tensor. We use an NPL [38] and learn the transformation matrices for each spatial tensor ($480 \times 28 \times 28$). This operation is iterated over 32 temporal slices. The output of NPL from all the time steps are concatenated to obtain F^W (the 3D world feature) and $\mathcal{G}(f_1^g(x))$ for input x.

The auto-encoder in the viewpoint-generator involves pooling F^W temporally followed by flattening the features (denoted by F_{inter}) and fed to two MLPs. The first MLP projects the $(c+3) \times m \times n$ dimensional vector to a lower dimension $d_{low} = 128$ and then another MLP to upsample the former to a $(c+3) \times m \times n$ dimensional vector. We invoke a reconstruction loss to minimize the output of the auto-encoder and F_{inter}.

Hyper-parameters: $\alpha = 1.0$ for mixup, momentum $= 0.999$ for momentum encoder and softmax temperature $\tau = 0.07$. The queue size of MoCo for pre-training is set to 2048. For optimization, we use Adam with 10^{-3} learning rate and 10^{-5} weight decay. All the experiments are trained on 2 V100 GPUs, with a batch size of 32 videos per GPU.

4.3. Evaluation setup

For downstream task, we evaluate the pre-trained ViewCLR models for the task of action classification. We evaluate on (1) **linear probe** where the entire encoder is frozen and a single linear layer followed by a *softmax* layer is trained with cross-entropy loss, and (2) **finetune** where the entire encoder along with a linear and *softmax* layer is trained with cross-entropy loss. Note that the encoder $f(\cdot)$ is initialized with the ViewCLR learned weights. More details for training the downstream action classification framework is provided in the Supplementary.

4.4. Ablation Study

In this section, we empirically show the effectiveness of our viewpoint-generator and the associated loss functions introduced in ViewCLR. Our baseline model is MoCo trained with only InfoNCE loss. In Table 1, we provide the linear-probe and finetuned action classification results on NTU-60 and NUCLA datasets. For our ablation studies, we follow the evaluation protocol proposed in [49] as it better represents the cross-view challenge. In this protocol, we take only 0° viewpoint from the CS split for training, and

Method	Linear Probe				Fine-tune			
	NTU-60			NUCLA	NTU-60			NUCLA
	CVS1	CVS2	CVS3	$V_3^{1,2}$	CVS1	CVS2	CVS3	$V_3^{1,2}$
InfoNCE	28.9	20.0	20.4	37.6	82.5	74.4	73.8	81.0
ViewCLR	**42.5**	**32.5**	**30.6**	**46.6**	**84.2**	**77.0**	**75.8**	**84.6**
ViewCLR-$\mathcal{L}_{3D}$	37.5	27.1	24.9	40.3	83.2	75.7	74.5	82.9
ViewCLR-$\mathcal{L}_{Adv}$	39.2	30.3	28.1	45.8	83.8	76.5	75.4	84.1
ViewCLR-$\mathcal{L}_{Adv}$-$\mathcal{L}_{3D}$	32.6	24.9	23.8	43.9	82.9	75.1	74.1	82.2
ViewCLR-$\mathcal{T}''$	42.1	31.9	30.2	45.9	84.1	76.8	75.5	84.3

Table 1: Ablation Study of ViewCLR by evaluating on NTU-60 (CVS protocol) and NUCLA datasets for the task of action classification. All the baseline MoCo models with InfoNCE loss are trained for 500 epochs. Whereas, the ViewCLR models are initialized with InfoNCE models trained for 300 epochs and then trained with the additional losses for 200 epochs. The results are provided for Linear-probe and fine-tuned evaluation setup. All the methods are trained in an unsupervised manner.

we test on the 0°, 45°, 90° views of the cross-subject test split. We call this protocol crossview-subject CVS1, CVS2, and CVS3 respectively. *Our focus is mainly to improve for the unseen and distinct view of 45° and 90°*. The models trained on NUCLA are initialized with NTU-60 pre-trained weights.

In Table 1, we show that ViewCLR outperforms traditional contrastive model (MoCo) on linear-probe evaluation by a significant margin for both seen (CSV1) and unseen scenarios. The relative improvement on seen camera view-point is 47% whereas it is upto 62.5% on unseen camera viewpoint on NTU-60. The improvement is also consistent for the fine-tuned models. Next, we provide a full diagnosis of ViewCLR to understand the driving force of this improvement.

We remove the 3D Loss that encourages the representation of the videos to project in the same 3D world coordinate system. This model is indicated by ViewCLR-$\mathcal{L}_{3D}$. By default, we also remove the autoencoder and consequently the reconstruction loss from this model. Thus, the viewpoint-generator computes feature map $\mathcal{G}(f_1(x))$ with the adversarial loss to minimize the mixed contrastive loss. Although the performance of this model is superior to the baseline MoCo model but we show that the absence of the proposed 3D Loss significantly hampers the performance. Thus, the learned representations from the viewpoint-generator highly relies on our proposed 3D Loss. Conversely, we remove the adversarial loss and retain the 3D loss in ViewCLR (referred to as ViewCLR-$\mathcal{L}_{adv}$). This experiment further confirms the effectiveness of our proposed 3D loss. The performance gap of ViewCLR-$\mathcal{L}_{adv}$ model w.r.t. ViewCLR is owing to the adversarial loss that enables $\mathcal{G}$ to generate latent viewpoint dissimilar to the original viewpoint. Finally, we pre-train ViewCLR by removing the viewpoint-generator in the third stream. So, the third stream only provides feature to perform manifold mixup of the features. This model (ViewCLR–$\mathcal{L}_{adv}$–$\mathcal{L}_{3D}$) is trained only with the mixup contrastive loss $\mathcal{L}_{CL}^{\text{mix}}$. This model further corroborates the effectiveness of the viewpoint-generator. Thus, we show that all the components of ViewCLR along with the proposed losses are instrumental for learning viewpoint invariant video representation.

Method	NTU-60		NUCLA	NTU-120	
	CS	CV	$V_3^{1,2}$	CS_1	CS_2
Supervised (S3D) [55]	85.1	86.9	81.3	77.6	80.9
ρ-BYOL (SSL) [11]	87.1	89.7	87.1	80.9	82.4
ViewCLR (ρ-BYOL)	89.5	92.9	**89.1**	83.8	85.7
MoCo (SSL) [17]	87.5	91.3	87.2	81.1	83.3
ViewCLR (MoCo)	89.7	**94.1**	**89.1**	84.5	86.2
K400 pre-trained S3D (SSL)	**90.1**	92.3	88.6	**85.1**	84.9

Table 2: Comparison of ViewCLR with representative baselines by finetuning the encoders for action classification. All the SSL models on NUCLA are pre-trained with NTU-60. K400 pre-trained S3D is pre-trained on Kinetics-400 using MoCo [17].

In addition to the above ablations, we also perform the ViewCLR experiments with two transformations (ViewCLR-$\mathcal{T}''$) as in MoCo and feed the same encoder data (generated by $\mathcal{T}'$) to the Viewpoint Generator. This experiment shows that a minor performance gain in ViewCLR is also attributed to the additional views w.r.t. MoCo.

4.5. Comparison to the state-of-the-art

Most of our state-of-the-art (SOTA) comparison includes self-supervised approaches using 3D Poses since cross-view datasets like NTU RGB+D and NUCLA are popular for skeleton based action recognition. To provide an extensive SOTA comparison, we also present the supervised approaches using (i) models pre-trained on large datasets like Kinetics-400, and (ii) multi-modal (RGB + Poses) information.

Comparison with representative baselines. In Table 2, we present the action classification performance of ViewCLR along with the representative baselines on NTU-60, NUCLA and NTU-120 datasets. Our representative baselines are (i) S3D [55] encoder trained from scratch with randomly initialized weights, (ii) S3D encoders pre-trained using SSL methods like MoCo [17] and ρ-BYOL [11], and (iii) S3D encoder trained on large-scale Kinetics [18] dataset using SSL method MoCo. All the SSL models on NUCLA are

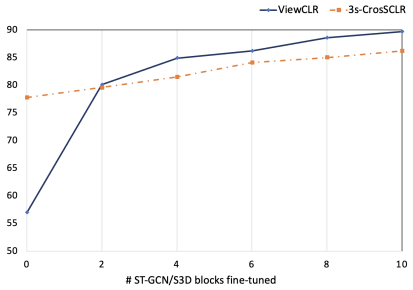

Figure 4: Partial fine-tuning results of ViewCLR vs. 3s-CrosSCLR on NTU-60 (CS). Two ST-GCN blocks in 3s-CrosSCLR are fine-tuned for every one block of S3D in ViewCLR.

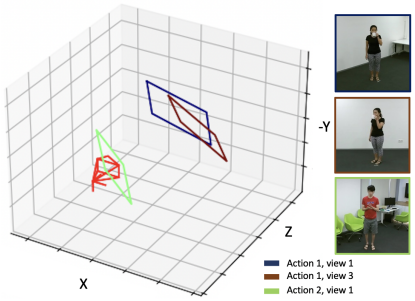

Figure 5: Visualization of the learned Projections for different scenarios. The learned camera view is fixed and treated as the origin of the world coordinate system while transforming the learned projections.

pre-trained with NTU-60. In Table 2, we adapted ViewCLR with another SSL strategy BYOL following the implementation provided in [11]. Details of this adaptation is provided in the supplementary. Although, ρ-BYOL cannot surpass the action classification performance when compared with SSL using MoCo owing to small scale data training. But ViewCLR variant with BYOL outperforms the supervised and SSL baselines. This improvement is even more significant for the ViewCLR with MoCO. Thus, the effectiveness of ViewCLR with BYOL and MoCo shows its robustness to different SSL methods. Moreover, we find that ViewCLR achieves competitive results with encoders pre-trained with large-scale Kinetics [18] dataset. In fact, for challenging cross-view protocols, ViewCLR outperforms encoder using extra data of the scale of Kinetics. This shows the importance of ViewCLR type SSL w.r.t. primitive pre-training schemes with large-scale dataset for learning viewpoint invariant features.

Linear Probe on NTU-60. In Table 3 (at left), we present the linear-probing on NTU-60 where ViewCLR significantly outperforms the MoCo model, however lags behind other self-supervised skeleton based methods [34, 27] and also Motion decoder [25] which leverages optical flow information. Poses and Optical Flow cues characterize highly prominent patterns pertaining to the viewpoint. So, the methods [39, 46, 34, 25, 27] exploiting these cues outperform ViewCLR with RGB input in Table 3. However, both these cues require additional compute and (supervised) pose data labels. Also, they lack appearance information which is encoded by RGB. As a result, all these methods are outperformed by ViewCLR when the pre-trained encoder is fine-tuned (see Table 3). Moreover, a recent article [16] suggested that Linear Probe evaluation misses the opportunity of pursuing strong but non-linear features and thus they proposed a new partial fine-tuning protocol. Following this new protocol, we fine-tune the last S3D block of ViewCLR against two ST-GCN blocks of 3s-CrosSCLR while freezing the other blocks on NTU (CS protocol). We find that finetuning only one S3D block of ViewCLR boosts the accuracy significantly from 57.0% to 81.1% outperforming 3s-CrosSCLR that yields 79.6% accuracy. We present the full partial fine-tuning results of ViewCLR vs. 3s-CrosSCLR in Fig. 4.

Finetuned Results on NTU-60. In Table 3 (at right), we present the comparison to the SOTA methods on NTU-60 dataset in which ViewCLR outperforms all the unsupervised methods. In order to compare with the models using RGB and Poses, we perform a late fusion of ViewCLR with logits obtained from a skeleton based action recognition model, CTR-GCN [24]. Our ViewCLR + Poses model outperforms all the SOTA results. It also indicates the complementary nature of our RGB based ViewCLR model and the skeleton based CTR-GCN model.

Transfer Ability. In Table 4, we present the SOTA results of NTU-60 pre-trained ViewCLR on NUCLA and NTU-120 datasets. The action classification accuracy on both the datasets are on par with the SOTA methods which show the generalization capability of ViewCLR. Skeleton cloud colorization [57] outperforms ViewCLR on NUCLA as it takes into account a high dimensional point cloud input which enables this method to address self-occlusions present in this dataset. However, this method fails to generalize over large-scale dataset like NTU-60. ViewCLR + Poses outperforms methods like STA [8] and VPN [9] which are pre-trained on ImageNet + Kinetics-400 (INK) in contrast to self-supervised NTU-60 ViewCLR pre-training. In Table 4 (at right), we show that the performance of downstream task enhances with increase in the size of pre-trained data. The performance of ViewCLR improves by upto 1.9% on NTU-120 when the size of the pre-trained data increases from 34K to 55k video samples.

Qualitative Analysis. In Fig. 5, we provide a visualization of the learned camera for three samples with same action but different viewpoint and different action with same viewpoint. We fix the learned camera position to represent it as the origin of the world coordinate system and transform the

Method	Modality	NTU-60	
		CS	CV
LongTGAN [62]	Poses	39.1	48.1
MS^{2}L [26]	Poses	52.6	-
AS-CAL [39]	Poses	58.5	64.8
P&C [46]	Poses	50.7	76.3
SeBiReNet [34]	Poses	-	79.7
Motion decoder [25]	Flow	77.0	78.8
3s-CrosSCLR [27]	Poses	77.8	83.4
MoCo [17]	RGB	30.5	33.5
ViewCLR	RGB	57.0	60.2

Method	Modality	NTU-60	
		CS	CV
Supervised [9]	RGB + Poses	93.5	96.2
MS^{2}L [26]	Poses	78.6	-
3s-CrosSCLR [27]	Poses	86.2	92.5
Motion Decoder [25]	Depth	68.1	63.9
Colorization [57]	Depth	88.0	94.9
Motion Decoder [25]	Flow	80.9	83.4
Motion Decoder [25]	RGB	55.5	49.3
ViewCLR	RGB	89.7	94.1
ViewCLR+Poses	RGB + Poses	**93.7**	**97.0**

Table 3: Linear Probe (at left) and Fine-tune (at right) action classification results of unsupervised methods on NTU-60. The approaches with Pose and Flow inputs are inherently at an advantage in linear probing as their inputs become more similar across viewpoints with additional compute and training data. Whereas, ViewCLR outperforms all the unsupervised methods when fine-tuned.

	Method	Modality	NUCLA $V_3^{1,2}$
Super.	CTR-GCN [6]	Poses	96.5
	Separable STA [8]	RGB + Poses	92.4
	VPN [9]	RGB + Poses	93.5
Unsuper.	MS^{2}L [26]	Poses	86.8
	Motion Decoder [25]	Depth	62.5
	Colorization [57]	Depth	94.0
	ViewCLR	RGB	89.1
	ViewCLR+Poses	RGB + Poses	**97.2**

	Method	Modality	NTU-120	
			CS_1	CS_2
Super.	CTR-GCN [6]	Poses	88.9	90.6
	Separable STA [8]	RGB + Poses	83.8	82.5
	VPN [9]	RGB + Poses	86.3	87.8
Unsuper.	AS-CAL [39]	Poses	48.6	49.2
	3s-CrosSCLR [27]	Poses	80.5	80.4
	ViewCLR†	RGB	82.1	84.3
	ViewCLR	RGB	84.5	86.2
	ViewCLR+Poses	RGB + Poses	**91.1**	**92.3**

Table 4: Comparison to the SOTA results on NUCLA and NTU-120. The supervised methods using RGB + Poses modalities presented in these tables are pre-trained with ImageNet1K + Kinetics labels in contrast to our unsupervised pre-training. All unsupervised NUCLA models are pretrained on NTU-60 and ViewCLR† is pre-trained on NTU-60.

2D projections accordingly. We notice that this latent viewpoint projection differs for different actions captured from the same viewpoint and similar for the same action captured from different camera viewpoint.

5. Related work

Self-supervised video representation. For learning self-supervised video representation many works have exploited the temporal structure of the videos, such as predicting if frames appear in order, reverse order, shuffled, color-consistency across frames, etc [21, 22, 12, 36, 33, 54, 53, 51, 40]. On the other hand, some methods have been taking advantage of the multiple modalities of videos like audio, text, optical flow, etc by designing pretext tasks for their temporal alignment [7, 20, 2, 35, 37, 32, 1]. Whereas, very less attention is given towards learning viewpoint invariant video representation which is crucial for real-world applications.

View Invariant Action Recognition. With the advancements in the field of Graph Convolutional Networks [19] and the availability of abundant 3D Pose data [41], many works have studied skeleton based action recognition [56, 47, 43, 44, 24, 30, 6]. These skeleton based methods are robust to viewpoint changes due to their extension across depth dimension. Furthermore, to encode appearance information in contrast to the Pose based features, several multi-modal approaches utilizing both RGB and Poses have been proposed in [42, 28, 31, 3, 4, 8, 9]. Recently, several skeleton based self-supervised methods have been proposed in [26, 62, 39, 46, 34]. CrosSCLR [27] performing positive mining across different views (joints, bones, motion) is one of the effective skeleton based self-supervised model till date. However, these methods utilizing 3D poses are limited to indoor scenarios or availability of high quality poses which is impractical for real-world applications. In contrast, ViewCLR learns viewpoint invariant representation using RGB input only, thus encoding appearance information. Most similar to our work, NPL [38] is a geometric based layer to learn 3D viewpoint invariant representation in supervised settings. Different from NPL, ViewCLR can be considered as an augmentation tool for learning self-supervised viewpoint invariant representation.

6. Conclusions

We have shown that a complementary viewpoint generation of a video while learning self-supervised video representation can significantly improve the learned representation for downstream action classification task. We presented ViewCLR that learns latent viewpoint representation of videos through a viewpoint-generator while optimizing the self-supervised contrastive loss. Our experiments show the importance of each component of ViewCLR and also confirm its robustness to unseen viewpoints. We believe that ViewCLR is a first step towards generalizing the unsupervised video representation for unseen camera viewpoints and hence, will be a crucial takeaway for the vision community.

References

[1] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. *CoRR*, abs/1705.08168, 2017.

[2] Relja Arandjelovic and Andrew Zisserman. Objects that sound. *CoRR*, abs/1712.06651, 2017.

[3] Fabien Baradel, Christian Wolf, and Julien Mille. Human activity recognition with pose-driven attention to rgb. In *The British Machine Vision Conference (BMVC)*, September 2018.

[4] Fabien Baradel, Christian Wolf, Julien Mille, and Graham W. Taylor. Glimpse clouds: Human activity recognition from unstructured feature points. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733. IEEE, 2017.

[6] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13359–13368, 2021.

[7] Joon Son Chung and Andrew Zisserman. Out of time: automated lip sync in the wild. In *Workshop on Multi-view Lip-reading, ACCV*, 2016.

[8] Srijan Das, Rui Dai, Michal Koperski, Luca Minciullo, Lorenzo Garattoni, Francois Bremond, and Gianpiero Francesca. Toyota smarthome: Real-world activities of daily living. In *ICCV*, 2019.

[9] Srijan Das, Saurav Sharma, Rui Dai, Francois Bremond, and Monique Thonnat. Vpn: Learning video-pose embedding for activities of daily living, 2020.

[10] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9588–9597, October 2021.

[11] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3298–3308, 2021.

[12] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[13] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[14] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *Neurips*, 2020.

[15] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv:2111.06377*, 2021.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

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

[19] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[20] Bruno Korbar, Du Tran, and Lorenzo Torresani. Co-training of audio and video representations from self-supervised temporal synchronization. *CoRR*, abs/1807.00230, 2018.

[21] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Kumar Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequence, 2017.

[22] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Kumar Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequence, 2017.

[23] Kibok Lee, Yian Zhu, Kihyuk Sohn, Chun-Liang Li, Jinwoo Shin, and Honglak Lee. i-mix: A domain-agnostic strategy for contrastive representation learning. In *ICLR*, 2021.

[24] Shi Lei, Zhang Yifan, Cheng Jian, and Lu Hanqing. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019.

[25] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Unsupervised learning of view-invariant action representations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[26] Lilang Lin, Sijie Song, Wenhan Yang, and Jiaying Liu. Ms2l: Multi-task self-supervised learning for skeleton based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, page 2490–2498, New York, NY, USA, 2020. Association for Computing Machinery.

[27] Li Linguo, Wang Minsi, Ni Bingbing, Wang Hang, Yang Jiancheng, and Zhang Wenjun. 3d human action representation learning via cross-view consistency pursuit. In *CVPR*, 2021.

[28] Guiyu Liu, Jiuchao Qian, Fei Wen, Xiaoguang Zhu, Rendong Ying, and Peilin Liu. Action recognition based on 3d skeleton and rgb frame fusion. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 258–264, Nov 2019.

[29] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C. Kot. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[30] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 143–152, 2020.

[31] Zelun Luo, Jun-Ting Hsieh, Lu Jiang, Juan Carlos Niebles, and Li Fei-Fei. Graph distillation for action detection with privileged modalities. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[32] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *CVPR*, 2020.

[33] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Shuffle and Learn: Unsupervised Learning using Temporal Order Verification. In *ECCV*, 2016.

[34] Qiang Nie, Ziwei Liu, and Yunhui Liu. Unsupervised 3d human pose representation with viewpoint and pose disentanglement. In *European Conference on Computer Vision (ECCV)*, 2020.

[35] Andrew Owens and Alexei A. Efros. Audio-visual scene analysis with self-supervised multisensory features, 2018.

[36] Lyndsey C. Pickup, Zheng Pan, Donglai Wei, YiChang Shih, Changshui Zhang, Andrew Zisserman, Bernhard Scholkopf, and William T. Freeman. Seeing the arrow of time. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2043–2050, 2014.

[37] AJ Piergiovanni, Anelia Angelova, and Michael S. Ryoo. Evolving losses for unsupervised video representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[38] AJ Piergiovanni and Michael S. Ryoo. Recognizing actions in videos from unseen viewpoints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4124–4132, June 2021.

[39] Haocong Rao, Shihao Xu, Xiping Hu, Jun Cheng, and Bin Hu. Augmented skeleton based contrastive action learning with momentum lstm for unsupervised action recognition. *Information Sciences*, 569:90–109, 2021.

[40] Adrià Recasens, Pauline Luc, Jean-Baptiste Alayrac, Luyu Wang, Florian Strub, Corentin Tallec, Mateusz Malinowski, Viorica Patraucean, Florent Altché, Michal Valko, Jean-Bastien Grill, Aäron van den Oord, and Andrew Zisserman. Broaden your views for self-supervised video learning. *CoRR*, abs/2103.16559, 2021.

[41] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[42] Amir Shahroudy, Gang Wang, and Tian-Tsong Ng. Multimodal feature fusion for action recognition in rgb-d sequences. In *2014 6th International Symposium on Communications, Control and Signal Processing (ISCCSP)*, pages 1–4, May 2014.

[43] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[44] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. *IEEE Transactions on Image Processing*, 29:9532–9545, 2020.

[45] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[46] Kun Su, Xiulong Liu, and Eli Shlizerman. Predict & cluster: Unsupervised skeleton based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9631–9640, 2020.

[47] Yansong Tang, Yi Tian, Jiwen Lu, Peiyang Li, and Jie Zhou. Deep progressive reinforcement learning for skeleton-based action recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5323–5332, June 2018.

[48] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 4489–4497, Washington, DC, USA, 2015. IEEE Computer Society.

[49] Gül Varol, Ivan Laptev, Cordelia Schmid, and Andrew Zisserman. Synthetic humans for action recognition from unseen viewpoints. *IJCV*, 2021.

[50] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6438–6447, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[51] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. "tracking emerges by colorizing videos". In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[52] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning, and recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2649–2656, June 2014.

[53] Xiaolong Wang, Kaiming He, and Abhinav Gupta. Transitive invariance for self-supervised visual representation learning. In *ICCV*, 2017.

[54] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019.

[55] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*, volume 11219 of *Lecture Notes in Computer Science*, pages 318–335. Springer, 2018.

[56] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.

[57] Siyuan Yang, Jun Liu, Shijian Lu, Meng Hwa Er, and Alex C. Kot. Skeleton cloud colorization for unsupervised 3d action representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13423–13433, October 2021.

[58] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*, 2019.

[59] Sangdoo Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, and Jinhyung Kim. Videomix: Rethinking data augmentation for video classification. *arXiv preprint arXiv:2012.03457*, 2020.

[60] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*, 2018.

[61] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[62] Nenggan Zheng, Jun Wen, Risheng Liu, , Liangqu Long, Jianhua Dai, and Zhefeng Gong. Unsupervised representation learning with long-term dynamics for skeleton based action recognition. In *AAAI*, pages 2644–2651, 2018.