Video Contrastive Learning with Global Context

Haofei Kuang[*,1,3], Yi Zhu[2], Zhi Zhang[2], Xinyu Li[2], Joseph Tighe[2],
Sören Schwertfeger[3], Cyrill Stachniss[1], Mu Li[2]

[1] University of Bonn   [2] Amazon Web Services   [3] ShanghaiTech University

Abstract

Contrastive learning has revolutionized the self-supervised image representation learning field and recently been adapted to the video domain. One of the greatest advantages of contrastive learning is that it allows us to flexibly define powerful loss objectives as long as we can find a reasonable way to formulate positive and negative samples to contrast. However, existing approaches rely heavily on the short-range spatiotemporal salience to form clip-level contrastive signals, thus limit themselves from using global context. In this paper, we propose a new video-level contrastive learning method based on segments to formulate positive pairs. Our formulation is able to capture the global context in a video, thus robust to temporal content change. We also incorporate a temporal order regularization term to enforce the inherent sequential structure of videos. Extensive experiments show that our video-level contrastive learning framework (VCLR) is able to outperform previous state-of-the-arts on five video datasets for downstream action classification, action localization, and video retrieval.

1. Introduction

Self-supervised video representation learning has garnered great attention in the last several years. The allure of this paradigm is the promise of annotation-free ground-truth to ultimately learn a superior data representation. Initial attempts to improve the quality of learned representations were via crafting various pretext tasks [3, 12, 27, 29, 40, 49, 50, 54, 59]. Recently, contrastive learning methods [6, 8, 9, 19] based on instance discrimination [57] have revolutionized the self-supervised image representation learning field. They have consistently outperformed their supervised counterparts on downstream tasks like image classification, object detection and segmentation. Hence, several recent works have started to consider contrastive learning in the video domain [10, 20, 21, 43, 46, 51, 60, 61].

A key component of contrastive learning is to define positive and negative samples to contrast. For example, in the
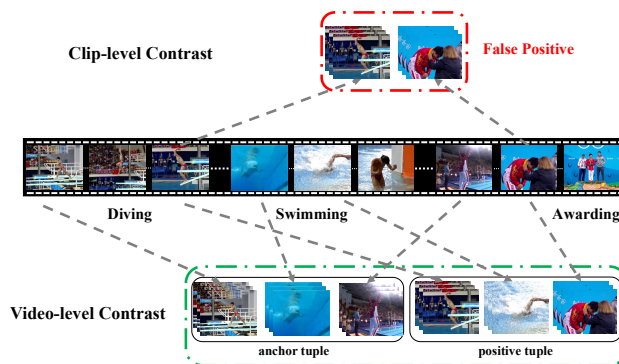

Figure 1: Random sampling of clips from a long video may generate false positive pairs. Our proposed video-level contrastive framework based on segments can generate robust positive training pairs and reason the global information.

image domain, most approaches use random crops from the same image as a positive pair and consider other images in the dataset as negative samples [7, 23, 57]. For video, there are several widely adopted formulations. (i) Instance-based methods [17, 22, 43, 61] use random frames/clips from the same video as a positive pair. (ii) Pace-based methods [51, 60] use clips from the same video but with different sampling rates as a positive pair. (iii) Prediction-based methods [20, 21] use predictions from auto-regressed and ground-truth features at the same spatiotemporal location as a positive pair. However, all previous methods define positive pairs to perform contrastive learning on frame-level or clip-level, which do not capture the global context of a video in a long temporal range. In addition, random sampling of clips from an untrimmed video may generate false positive pairs as shown in Fig. 1 top.

In this paper, we propose a new formulation for generating positive pairs to address the above limitation. As shown in Fig. 1 bottom, we first uniformly divide the video into several segments, and randomly pick a clip from each segment to form the anchor tuple. Then we randomly pick a clip from each segment again to form the positive tuple. We consider these two tuples as a positive pair and tuples from other videos as negative samples. In this way, our formu-

---

lation of positive samples is flexible (i.e., can be trained on any videos regardless of duration or sampling rate) and is able to reason about the global information across a video. However, global context alone is not enough to guide the unsupervised video representation learning since it does not enforce the inherent sequential structure of a video. Hence, we introduce a regularization loss based on the temporal order constraint. To be specific, we shuffle the frame order inside each tuple and ask the model to predict if the tuple has the correct temporal order. Since both proposed techniques work on the video-level, we term our method as *VCLR, video-level contrastive learning*. Our contributions can be summarized as follows.

- We introduce a new way to formulate positive pairs based on segments for video-level contrastive learning.

- We incorporate a temporal order regularization term to enforce the inherent sequential structure of videos.

- Our proposed video-level contrastive learning framework (VCLR) outperforms previous literature on five datasets for downstream action classification, action localization, and video retrieval.

2. Related Work

Self-supervised image representation learning Initial attempts for self-supervised image representation learning are via designing various pretext tasks. Recently, a type of contrastive learning method based on instance discrimination has taken-off as it has been consistently demonstrated to outperform its supervised counterparts on downstream tasks [6, 8, 9, 19]. The core idea behind contrastive learning is to learn representations by distinguishing between similar and dissimilar instances [57]. More specifically, it obtains the supervision signal by designing different forms of positive and negative pairs, such as random crops from the same image [7, 23], different views of the same instance [46], etc.
Self-supervised video representation learning Compared with images, videos have another axis, temporal dimension, which we can use to craft pretext tasks for self-supervised representation learning. There are many previous attempts including predicting the future [12, 38, 48, 49], predicting the correct order of shuffled frames [15, 40] or video clips [32, 59], predicting video rotation [27], solving a space-time cubic puzzle [29], predicting motion and appearance statistics [50], predicting speed [3, 62] and exploring video correspondence [14, 25, 54, 55]. There is another line of work using multi-modality signals as supervision, such as audio [1, 2, 30, 41, 42] and language [1, 39].

Several recent works have considered contrastive learning in the video domain. Despite their loss objectives are similar (e.g., NCE [47] and its variant), their major differences lie in how they formulate positive and negative samples. CBT [10] adapts masked language modeling [11] to

a video sequence, and uses the video clip and its masked version as a positive pair. DPC [20] and MemDPC [21] use predictions from autoregressed and ground-truth features at the same spatiotemporal location as a positive pair. Pace [51] and VTHCL [60] proposes to use video clips of the same action instance but with different visual tempos as a positive pair. CMC [46] introduces two ways of sampling positive pairs: (1) different frames from the same video and (2) RGB and flow data of the same frame. Similar to CMC, VINCE [17] and SeCo [61] also use different frames from the same video, while IIC [45] and CVRL [43] use different clips from the same video as positive samples. Similar to SeCo, TCGL [36] combine the long-term and short-term snippets with a graph contrastive learning framework to model the multi-scale temporal dependencies.

Different from previous approaches, our work proposes a new way to formulate positive pairs based on segments. Given our formulation, we can perform contrastive learning on video-level rather than frame- or clip-level, and achieve promising results both quantitatively and qualitatively. In addition, our method is generally applicable to various network architectures and orthogonal to other advancements in the self-supervised video representation learning domain.

3. Methodology

3.1. Preliminary

Before diving into our proposed framework, we first revisit the concept of contrastive learning in image domain [7, 23, 57]. Here we use MoCoV2 [9] as an illustrating example. More formally, given a set of images $\mathcal{X}$, an image x_i is sampled from $\mathcal{X}$ and is augmented to generate a positive pair x_i^a and x_i^+. A set of negative samples $\mathcal{N}^-$ are then selected from the rest of $\mathcal{X}$, i.e., $x_j \in \mathcal{N}^-, j \neq i$. Two encoders, query encoder f_q and key encoder f_k, are used to obtain the visual representations, e.g., 2048-dim features from ResNet50 backbone. These visual representations are then projected via MLP heads g_q and g_k to lower dimensional embeddings for similarity comparison. For notation simplicity, we represent anchor embedding as $q = g_q(f_q(x_i^a))$, positive embedding as $p = g_k(f_k(x_i^+))$, and negative embeddings as $n_j = g_k(f_k(x_j^-))$. At this point, the problem is formulated as a (N+1)-way classification task, and can be optimized by InfoNCE loss [47] as

$$\mathcal{L}_{\text{NCE}}(q, p, \mathcal{N}^-) = -\log \frac{e^{\text{sim}(q,p)}}{e^{\text{sim}(q,p)} + \sum_{j=1}^{N} e^{\text{sim}(q,n_j)}}. \quad (1)$$

Here, $\text{sim}(\cdot)$ is the distance metric used to measure the similarity between feature embeddings, e.g., dot product. N denotes the number of negative samples. By optimizing the objective, the model learns to map similar instances closer and dissimilar instances farther apart in the embedding space. For more details, we refer the readers to [9, 23].
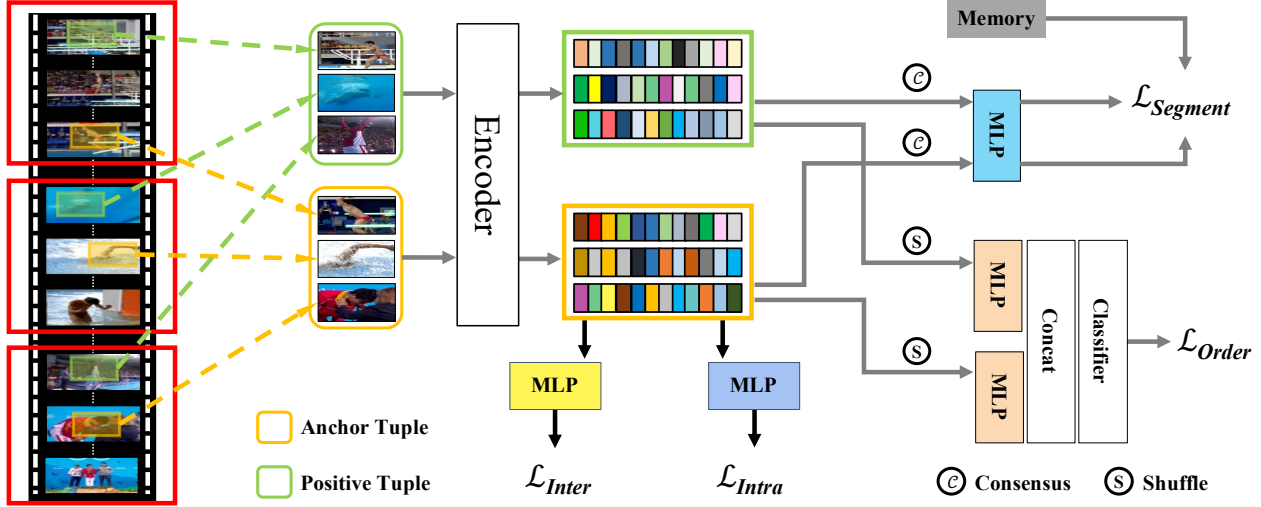
Figure 2: **Overview of the proposed VCLR framework**. We introduce a new way to formulate positive training pairs based on video segments. There are four loss objectives, $\mathcal{L}_{Inter}$ and $\mathcal{L}_{intra}$ following [61], and two new ones $\mathcal{L}_{Segment}$ and $\mathcal{L}_{Order}$ for video-level contrastive learning. Image is best viewed in color.

To adapt contrastive learning into the video domain, we need to find a way to define positive and negative samples. We have mentioned several in Sec. 2. In this paper, we follow a recent state-of-the-art SeCo [61] as our baseline given their strong performance. Specifically, they introduce two ways to formulate positive and negative samples: inter-frame and intra-frame instance discrimination.

Inter-frame instance discrimination Unlike approaches in the image domain that take random crops from the same image as a positive pair, inter-frame instance discrimination simply treats random frames from the same video as positive. Suppose we randomly pick three frames from a video, v_1, v_2 and v_3, inter-frame instance discrimination task considers v_1^a as the anchor frame, (v_1^+, v_2, v_3) as the positive samples, and frames from other videos in the dataset as negative samples $\mathcal{N}^-$. Here, v_1^a and v_1^+ are generated from v_1 by different augmentations. Mathematically, the loss objective can be written as

$$\mathcal{L}_{Inter} = \frac{1}{3} \sum \mathcal{L}_{NCE}(q_1^a, \{p_1^+, p_2, p_3\}, \mathcal{N}^-). \quad (2)$$

We represent anchor embeddinig as $q_1^a = g_q^e(f_q(v_1^a))$, and positive embeddings as $p_1^+ = g_k^e(f_k(v_1^+))$, $p_2 = g_k^e(f_k(v_2))$, $p_3 = g_k^e(f_k(v_3))$. g_q^e and g_k^e are MLP heads for inter-frame instance discrimination task. Note that, we have multiple positives in Eq. (2), we compute the contrastive losses separately and take their average.

Intra-frame instance discrimination In order to learn better appearance representation, intra-frame instance discrimination is adopted to explore the inherently spatial changes across frames. Different from inter-frame, we only use v_1^+ as a positive sample, and consider v_2, v_3 as negative sam-

ples. Hence, the loss objective can be written as

$$\mathcal{L}_{Intra} = \mathcal{L}_{NCE}(q_1^a, \{p_1^+\}, \{p_2, p_3\}). \quad (3)$$

Essentially, this task is the same as standard instance discrimination in the image domain except trained with a much smaller negative set. The embeddings in Eq. (3) are actually different from the ones in Eq. (2) due to using different MLP heads g_q^r and g_k^r for intra-frame instance discrimination task, but we reuse them for notation simplicity.

Despite the design of both inter- and intra- frame instance discrimination tasks, SeCo still performs contrastive learning on the frame-level. Simply put, intra-frame is between two crops in the same image, and inter-frame is between two frames in the same video. Some recent work [22, 43] use short video clips as input to perform instance discrimination, which can be considered doing contrastive learning in the clip-level. However, contrastive learning on frame- or clip-level are sub-optimal for video representation learning because they cannot capture the evolving semantics in the temporal dimension. Hence, we ask the question, how can we perform contrastive learning at the video-level?

3.2. Video-level contrastive learning

Long-range temporal structure plays an important role in understanding the dynamics in videos. In terms of supervised learning, there have been a series of work to explore global context [13, 16, 33, 52, 53, 56, 63, 64]. However, in terms of self-supervised learning, few efforts have been done to incorporate global video-level information.

In this work, inspired by [52], we propose to use the segment idea to sample positive pairs for contrastive learning. Formally, given a video V, we divide it into K seg-
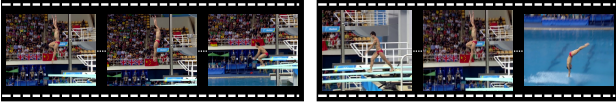
Figure 3: **Left**: It is challenging to identify the temporal order within a local sequence, which will likely confuse the model training. **Right**: With longer temporal context, we can easily tell the evolution of events.

ments $\{S_1, S_2, \cdots, S_K\}$ of the equal duration. Within each segment S_k, we randomly sample a frame v_k to formulate our anchor tuple, i.e., $t^a = \{v_1^a, v_2^a, \cdots, v_K^a\}$. Then, we take a second independent random sample of K frames in the same manner to formulate the positive tuple, $t^+ = \{v_1^+, v_2^+, \cdots, v_K^+\}$. We consider these two tuples as a positive pair, because both of them describe the evolving semantics inside a video. Intuitively, segment-based sampling can be considered as a form of data augmentation, so that the consensus from two tuples can produce two different views of the same video, which is what instance discrimination need to learn an effective representation.

In terms of loss objective, each frame in the tuple will produce its own preliminary frame-level representation. Then a consensus among these representations will be derived as the video-level representation. The final embedding of the anchor and positive tuple can be represented as

$$q_t^a = g_q^s(\mathcal{C}[f_q(v_1^a), f_q(v_2^a), ..., f_q(v_K^a)]) \quad (4)$$

$$p_t^+ = g_k^s(\mathcal{C}[f_k(v_1^+), f_k(v_2^+), ..., f_k(v_K^+)]) \quad (5)$$

where $\mathcal{C}$ denotes the consensus operation, e.g., average. g_q^s and g_k^s are MLP heads for video-level instance discrimination task. Once we have the embeddings q_t^a and p_t^+, we can compute the video-level contrastive loss as

$$\mathcal{L}_{segment} = \mathcal{L}_{\text{NCE}}(q_t^a, p_t^+, \mathcal{N}^-). \quad (6)$$

Here, we reuse the notation $\mathcal{N}^-$ to generally indicate the negative set, where each negative is a tuple of frames sampled from other videos in the dataset.

We want to point it out that, this formulation of positive samples has several advantages. First, it leads to a more robust training as it sees information from the entire video. Second, we can use this strategy to train a model on any types of video, regardless of duration or sampling rate. Third, this formulation is not limited to video frames with 2D CNNs, but also ready to be used for video clips with 3D CNNs. In addition, our formulation is orthogonal to other advancements in self-supervised video representation learning [20, 22, 60], which could be incorporated seamlessly.

3.3. Temporal order regularization

Video-level contrastive learning helps to capture global context as it sees information from the entire video, but it is

Method	Network	Top-1 Acc. (%)
VTHCL [60]	R3D-50	37.8
VINCE [17]	R2D-50	49.1
SeCo [61]	R2D-50	61.9
VCLR	R2D-50	**64.1**
Sup-ImageNet	R2D-50	52.3
Sup-Kinetics400	R2D-50	69.9

Table 1: **Linear evaluation on Kinetic400**. Our method outperforms previous methods and is close to supervised upper bound. Sup-ImageNet:supervised ImageNet weights. Sup-Kinetics400: supervised training on Kinetics400.

weak in enforcing the inherent sequential structure. Given temporal coherence is a strong constraint, there has been some work [15, 32, 40, 61] that use it as a supervision signal for self-supervised video representation learning.

In this paper, we also use the temporal order as a regularization term under our contrastive learning framework. To be specific, given tuples t^a and t^+, which are sampled based on segments described in Sec. 3.2, we randomly shuffle the frames inside each tuple with 50% chance. This will lead to a balanced 4-way classification problem: both tuples are in correct temporal order, t^a correct and t^+ shuffled, t^a shuffled and t^+ correct, and both shuffled.

In terms of loss objective, we first compute the embeddings of each tuple as o_t^a and o_t^+,

$$o_t^a = [g_q^o(f_q(v_1^a)), g_q^o(f_q(v_2^a)), ..., g_q^o(f_q(v_K^a))] \quad (7)$$

$$o_t^+ = [g_k^o(f_k(v_1^+)), g_k^o(f_k(v_2^+)), ..., g_k^o(f_k(v_K^+))] \quad (8)$$

g_q^o and g_k^o are MLP heads for temporal order regularization. In order to keep the temporal order within each tuple, we concatenate the embeddings and predict its order type,

$$y^* = h_o([\text{concat}(o_t^a), \text{concat}(o_t^+)]). \quad (9)$$

Here, h_o is a linear classifier to project the concatenated embeddings to logits. y^* is the order type prediction, i.e., class 0, 1, 2 or 3. In the end, we compute the cross-entropy loss between the predictions and pre-defined ground-truth

$$\mathcal{L}_{Order}(y, y^*) = -\sum y \log y^*. \quad (10)$$

Our temporal order regularization may seem similar to [15, 32], but we build it upon our video-level sampling framework. Simply put, previous approaches use local sequence to validate temporal order, while we use global sequence. One drawback of using local sequence is that it may not provide enough cues to predict the correct sequential structure. As shown in Fig. 3 left, any temporal order seems reasonable between these three frames sampled from a local sequence. On the contrary in Fig. 3 right, using our video-level sampling strategy, it is less ambiguous to determine the temporal order inside a tuple.

Method	Venue	Network	Top-1 Acc (%)	
			UCF101	HMDB51
ST-Puzzle [29]	AAAI19	R3D-50	65.8	33.7
MAS [50]	CVPR19	C3D	61.2	33.4
DPC [20]	ICCVW19	R-2D3D	75.7	35.7
SpeedNet [3]	CVPR20	S3D-G	81.1	48.8
VIE [65]	CVPR20	SlowFast	80.4	52.5
MemDPC [21]	ECCV20	R-2D3D	78.1	41.2
PacePred [51]	ECCV20	R(2+1)D	77.1	36.6
TT [26]	ECCV20	R3D-18	79.3	49.8
SeCo[1][61]	AAAI21	R2D-50	83.4	49.7
VCLR	-	R2D-50	**85.6**	**54.1**
Sup-ImageNet	-	R2D-50	81.6	49.0
Sup-Kinetics400	-	R2D-50	88.1	56.1

Table 2: **Downstream action classification on UCF101 and HMDB51**. Sup-Kinetics400: supervised Kinetics400 pretrained weights.

3.4. Overall framework

At this point, we present our video-level contrastive learning method, VCLR, for self-supervised video representation learning. Our overall framework can be seen in Fig. 2. During training, the final loss objective is a summation of previous four loss objectives,

$$\mathcal{L} = \mathcal{L}_{Inter} + \mathcal{L}_{Intra} + \mathcal{L}_{Segment} + \mathcal{L}_{Order} \quad (11)$$

For downstream tasks, we ignore the MLP heads designed for each task, e.g., g^e, g^r, g^s and g^o. We only use pretrained encoder f_q for extracting features or finetuning.

4. Experiments

4.1. Datasets

We conduct experiments on 5 datasets, Kinetics400 [28], UCF101 [44], HMDB51 [31], Something-Something-v2 [18] and ActivityNet [5]. **Kinetics400** consists of approximately 240k training and 20k validation videos trimmed to 10 seconds from 400 human action categories. **UCF101** contains $13,320$ videos spreading over 101 categories of human actions. **HMDB51** contains $6,849$ videos divided into 51 action categories. **Something-Something-v2** consists of 174 action classes and a total of $220,847$ videos. For notation simplicity, we refer this dataset as SthSthv2 for the rest of the paper. **ActivityNet** (V1.3) contains 200 human daily living actions. It has $10,024$ training and $4,926$ validation videos. Both UCF101 and HMDB51 have three official train-val splits, and we report performance on its split 1 for fair comparison to previous work. For other three datasets, we report performance on their validation sets.

---

[1]SeCo [61] reported higher numbers on UCF101 and HMDB51, but we follow the default setting from MemDPC [21] to perform finetuning on all R2D-50 based methods, so the comparisons in Table 2 are fair.

Method	Video Algorithm	Top-1 Acc (%)
VINCE [17]	TSN	31.4
	TSM	50.3
SeCo [61]	TSN	31.9
	TSM	50.7
VCLR	TSN	**33.3**
	TSM	**52.0**
Sup-ImageNet	TSN	33.0
	TSM	59.1

Table 3: **Downstream action classification on SthSthv2**. TSM: temporal shift module network [34]. All methods in this table use R2D-50 as encoder, and follow the same training setting for fair comparison.

4.2. Implementation details of pretraining

In terms of input, we follow our segment-based sampling strategy to choose two tuples from the same video as video-level anchor and positive sample. We set the number of segments to 3, i.e., each tuple contains three frames. The video-level anchor will also be used to generate frame-level anchor and positive sample to compute the inter-frame and intra-frame losses. For data augmentation, we apply random scales, color-jitter, random grayscale, random Gaussian blur, and random horizontal flip. In terms of network architecture, we use ResNet50 [24] as our backbone, plus four MLP heads designed for each loss. We use average operation for $\mathcal{C}$. In terms of optimization, we train the model on Kinetics400 dataset with a batch size of 512 for 400 epochs following [61]. We use SGD to optimize the network with an initial learning rate of 0.2, and annealed to zero with a cosine decay scheduler.

4.3. Linear evaluation on Kinetics400

In order to quantify the quality of learned representation, the most straightforward way is to treat the pretrained model as a feature extractor and train a classifier on top of the features to see its generalization performance.

Following [61], we uniformly sample 30 frames from each video, resize each frame with short edge of 256 and center-crop it to 224×224. We forward each frame through the frozen ResNet50 backbone, get the frame-level features and average them into video-level feature. A linear SVM is then trained on the video-level features of the Kinetics400 training set, and finally evaluated on its validation set.

As we can see in Table 1, our method outperforms previous literature in terms of linear evaluation on Kinetics400 dataset. Especially for fair comparison when using 2D ResNet50, our VCLR pretrained model is 11.8% higher than ImageNet pretrained weights, and 2.2% higher than previous state-of-the-art SeCo pretrained weights [61]. In addition, we further close the gap between unsupervised feature learning and supervised upper bound (64.1% vs 69.9%). This clearly demonstrates the effectiveness of performing contrastive learning on video-level.

Method	Modality	Network	Pretrain	UCF101				HMDB51			
				R@1	R@5	R@10	R@20	R@1	R@5	R@10	R@20
OPN [32]	V	VGG	UCF101	19.9	28.7	34.0	40.6	-	-	-	-
DRL [4]	V	VGG	UCF101	25.7	36.2	42.2	49.2	-	-	-	-
VCOP [59]	V	R(2+1)D	UCF101	14.1	30.3	40.4	51.1	7.6	22.9	34.4	48.8
VCP [37]	V	R3D-50	UCF101	18.6	33.6	42.5	53.5	7.6	24.4	36.3	53.6
MemDPC [21]	V	R-2D3D	UCF101	20.2	40.4	52.4	64.7	7.7	25.7	40.6	57.7
MemDPC [21]	F	R-2D3D	UCF101	40.2	63.2	71.9	78.6	15.6	37.6	52.0	65.3
CoCLR [22]	VF	S3D	UCF101	53.3	69.4	76.6	82.0	23.2	43.2	53.5	65.5
VCLR	V	R2D-50	UCF101	**46.8**	**61.8**	**70.4**	**79**	**17.6**	**38.6**	**51.1**	**67.6**
SpeedNet [3]	V	S3D-G	Kinetics400	13.0	28.1	37.5	49.5	-	-	-	-
SeCo [61]	V	R2D-50	Kinetics400	69.5	79.5	85	90.1	33.6	57.4	67.6	78.6
VCLR	V	R2D-50	Kinetics400	**70.6**	**80.1**	**86.3**	**90.7**	**35.2**	**58.4**	**68.8**	**79.8**

Table 4: **Downstream video retrieval on UCF101 and HMDB51**. V: RGB frames. F: optical flow.

Method	Classification	Localization	
	Top-1 Acc (%)	AUC (%)	AR@100 (%)
VINCE [17]	60.7	64.6	73.2
SeCo [61]	67.8	65.2	73.4
VCLR	**71.9**	**65.5**	**73.8**
Sup-ImageNet	67.2	64.8	73.4

Table 5: **Downstream action classification and localization on ActivityNet**. All methods use R2D-50 as encoder, and follow the same training setting for fair comparison.

4.4. Downstream action classification

A main goal of unsupervised learning is to learn features that are transferable [23]. In both computer vision and natural language processing, pretraining a network on large datasets and finetuning it on smaller datasets is the de facto way to achieve promising results on downstream tasks.

Following previous work [21, 61], we transfer our pretrained backbone and finetune all its layers on UCF101 and HMDB51 datasets. We compare our method to recent literature in Table 2. Note that there are more related work, but we only list methods that are pretrained on Kinetics400 and using RGB frames as input for fair comparison. As can be seen in Table 2, our method with 2D ResNet50 is able to outperform those using more advanced 3D CNNs [3, 65]. Compared to previous state-of-the-art SeCo [61], we find that video-level contrastive learning brings 2.2% and 4.4% absolute performance improvement on UCF101 and HMD51, respectively. Similarly, our VCLR pretrained models achieve higher accuracy than ImageNet pretrained weights on both datasets, and perform very close to Kinetics400 pretrained weights.

In order to have a complete understanding of our learned representation's transferability, we further evaluate VCLR on another two popular action classification datasets, Sth-Sthv2 and ActivityNet. This is because UCF101 and HMDB51 have small domain gap to Kinetics400, given they are all scene-focused datasets, e.g., most actions can be predicted correctly by using object or background prior. SthSthv2 is a motion-focused dataset and requires strong temporal reasoning because most activities cannot be in-

ferred based on spatial features alone (e.g. opening something, covering something with something). ActivityNet contains long and untrimmed videos, and thus requires strong global context modeling.

In terms of SthSthv2, we compare our learned representations to those from VINCE and SeCo in Table 3. As we can see, our learned representation achieves higher accuracy than both of them. We also show that our pretrained backbone is compatible with different video action recognition algorithms. Besides TSN [52], we adapt our pretrained weights to initialize a popular algorithm TSM [34] on Sth-Sthv2 and obtain promising results. In terms of ActivityNet, we also compare our learned representations to those from VINCE and SeCo in Table 5 and show significant performance improvements.

To summarize, by performing contrastive learning on video-level, our proposed VCLR is able to outperform previous state-of-the-art approaches on 4 downstream action classification datasets, including scene-focused datasets (UCF101 and HMDB51), motion-focused dataset (Sth-Sthv2) and untrimmed video dataset (ActivityNet).

4.5. Downstream temporal action localization

We also evaluate VCLR on temporal action localization task with ActivityNet. The goal of this task is to generate high quality proposals to cover action instances with high recall and high temporal overlap. To be specific, we adopt the popular BMN [35] method for action localization, and only change the input features to see which representation generalizes better. To evaluate proposal quality, Average Recall (AR) under multiple IoU thresholds are calculated. We calculate AR under different Average Number of proposals (AN) as AR@AN, and calculate the Area under the AR vs. AN curve (AUC) as metrics, where AN is varied from 0 to 100. As can be seen in Table 5, our VCLR learned representations not only outperforms previous methods by a large margin on classification task, it also performs better on action localization. Qualitatively, we show two visualizations of our predicted proposals in Fig. 4b.

$\mathcal{L}_{Intra}$	$\mathcal{L}_{Inter}$	$\mathcal{L}_{Segment}$	$\mathcal{L}_{Order}$	Top-1 Acc. (%)
✓				NA
	✓			59.0
		✓		62.1
			✓	51.8
✓	✓			60.7
✓	✓		✓	61.2
✓	✓	✓		63.5
✓	✓	✓	✓	**64.1**

Table 6: **Ablation study on loss objectives**. We prertain all configurations on Kinetics400 under the same setting, and report linear evaluation performance on it.

Num. of Segments (K)	Top-1 Acc. (%)
$K = 1$	61.2
$K = 2$	63.0
$K = 3$	64.1
$K = 4$	64.3
$K = 5$	64.4

Table 7: **Ablation study on the number of segments**. We report the linear evaluation performance here.

4.6. Downstream video retrieval

In addition to action classification and localization, we also evaluate VCLR on a common downstream task, video retrieval. For this task, we use the extracted feature from each video to perform nearest-neighbour retrieval, and the goal is to test if the query clip instance and its nearest neighbours belong to same semantic category. Similar to linear evaluation in Sec. 4.3, we treat the pretrained model as a feature extractor, and no finetuning is performed.

Following [22, 59], we use the validation video samples in UCF101 and HMDB51 to search the k nearest video samples from their training sets, respectively. We use Recall at k (R@k) as evaluation metric, that means if one of the top k nearest neighbours is from the same class as query, it is a correct retrieval. As can be seen in Table 4, our VCLR trained representation outperforms other methods with respect to all the R@k regardless of the pretraining dataset. Qualitatively, we show several visualizations of our top-3 video retrievals on UCF101 dataset in Fig. 4a. We note that a recent work CoCLR [22] performs better than us. However, CoCLR has been trained with a heavier network S3D [58] and multiple hard positive samples mined by using optical flow. We conduct another experiment in Sec. 5.3 to fairly compare with CoCLR using the same network architecture, and find that our performance is competitive to theirs even without using optical flow.

5. Discussion

In this section, we present ablation studies and important discussions on VCLR, including ablation study, extend VCLR to 3D network, and visualization.

Method	Modality	Retrieval	Classification	
			Linear	Finetune
$\mathcal{L}_{Inter}$ [22]	V	33.1	46.8	78.4
$\mathcal{L}_{Inter} + \mathcal{L}_{Segment}$	V	49.8	67.4	80.2
$\mathcal{L}_{Inter} + \mathcal{L}_{Segment} + \mathcal{L}_{Order}$	V	**51.7**	**70.9**	**81.7**
CoCLR [22]	VF	51.8	70.2	81.4

Table 8: **VCLR with 3D CNNs**. All models are pretrained on UCF101 under the same setting following [22].

Method	Pretrain	Classification	
		UCF101	HMDB51
DPC [20]	UCF101	60.6	44.9
DPC + $\mathcal{L}_{Segment}$	UCF101	62.0	46.3
DPC + $\mathcal{L}_{Segment} + \mathcal{L}_{Order}$	UCF101	**62.7**	**47.1**

Table 9: **VCLR with DPC** [20]. All models are pretrained under the same setting following [20].

5.1. Ablation study on loss objectives

Our method VCLR has four loss objectives: $\mathcal{L}_{Intra}$, $\mathcal{L}_{Inter}$, $\mathcal{L}_{Segment}$ and $\mathcal{L}_{Order}$. We now dissect their contributions and see how they impact the learned representations.

We have several observations from Table 6. First, video-level instance discrimination $\mathcal{L}_{Segment}$ is a strong supervision signal. By using this loss alone could learn good representations and outperform previous state-of-the-art SeCo (62.1 vs 61.9). Second, temporal order constraint $\mathcal{L}_{Order}$ is relatively weak. Using it alone only achieves 51.8% accuracy. However, when combined with other loss objectives, it can provide further regularization and push the performance to 64.1. Third, using intra-frame instance discrimination $\mathcal{L}_{Intra}$ alone is not enough to stabilize training as the negative set is too small to perform contrasting.

5.2. Ablation study on the number of segments

We set $K = 3$ as default number of segments in our experiments, now we discuss the impact of the number of segments for our method. We present the comparisons with respect to linear evaluation performance on K400 dataset in the Table 7.

We can see that with the increasing number of segments, the performance continue to improve, which indicates the necessity of using global context in training video models. But the performance saturates when using more segments. In our paper, we simply choose $K=3$ for a better training speed-accuracy trade-off.

5.3. General applicability

Our framework VCLR is not limited to any network architecture. For example, we could sample short video clips instead of frames from each segment to formulate positive pairs. In this way, a 3D CNN can be pretrained without using labels. In addition, VCLR is compatible with most loss objectives and is orthogonal to other advancements in

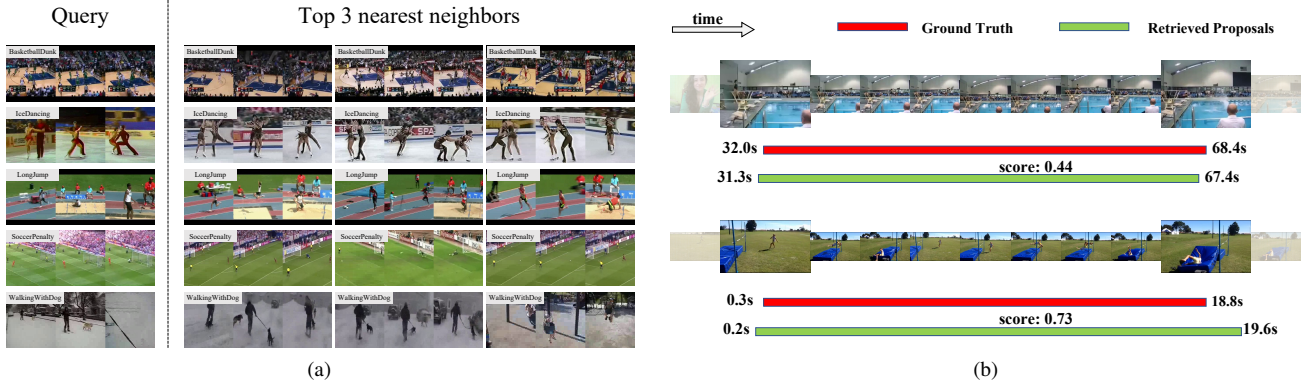| Query	Top 3 nearest neighbors

(a)

(b)

Figure 4: (a) **Video retrieval** results on UCF101. (b) **Action localization** results on ActivityNet. For both experiments, we use the learned representations from our VCLR model pretrained on Kinetics400 dataset.

self-supervised video representation learning.

VCLR with 3D CNNs Here we train a R3D-50 model using VCLR to demonstrate our framework's general applicability. In terms of baseline, we compare to a R3D-50 model pretrained by $\mathcal{L}_{Inter}$ on UCF101 dataset following [22]. We then use three segments and pretrain the model by adding loss objectives $\mathcal{L}_{Segment}$ and $\mathcal{L}_{Order}$. As seen in Table 8, our method significantly outperforms the corresponding baseline with respect to both video classification and retrieval task on UCF101. In addition, we compare to CoCLR [22], which uses optical flow to mine multiple hard positives for contrastive learning. We can see that our method is competitive to CoCLR even without optical flow.

VCLR with DPC Both $\mathcal{L}_{Segment}$ and $\mathcal{L}_{Order}$ are compatible to other algorithms in self-supervised video representation learning. Similarly, we incorporate them to a recent work DPC [20], which uses dense prediction as a pretext task, to train a R3D-18 model on UCF101. As shown in Table 9, a simple addition improves the transfer performance to downstream action classification task on both UCF101 and HMDB51, which demonstrates the necessity of using global context. We believe VCLR can be combined with other methods [43, 60] and show performance improvements.

5.4. Visualization

We visualize the activations of the conv5 output from ResNet50 models pretrained by SeCo [61] and ours. As can be seen in Fig. 5, VCLR is able to focus on the moving region of interest, while SeCo focuses more on the background (e.g., springboard and water). It is also interesting to note that VCLR generalizes better to unseen videos.

6. Conclusion

In this paper, we introduce a new way to formulate positive pairs for video-level contrastive learning. We also incorporate a temporal order regularization to enforce
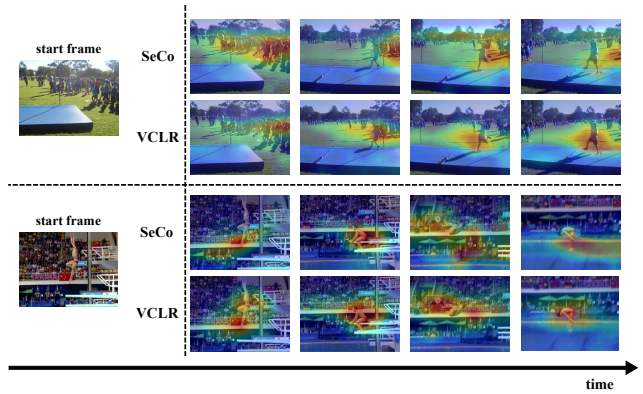

Figure 5: **Attention visualization between SeCo [61] and VCLR.** The upper video is from Kinetics400 validation set, and the bottom one is an unseen video from Internet.

the inherent sequential structure of videos during training. Our final framework VCLR shows state-of-the-art results on five video datasets for downstream action recognition/localization and video retrieval. Furthermore, we demonstrate the generalization capability of our method, e.g., pretrain 3D CNNs and are compatible with other self-supervised learning algorithms. We want to emphasize that our method is applicable to various inputs (i.e., short trimmed or long untrimmed videos), networks (i.e., 2D or 3D CNNs), methods and ready to scale to future untrimmed datasets for video self-supervised representation learning.

Regarding the workshop, learning a robust visual representation from video is crucial for various tasks, such as video understanding and editing. Our method can learn such features through self-supervised manner on unlimited video data, and we show its effectiveness on several widely adopted downstream tasks. We hope VCLR can serve as a strong baseline and facilitate research of using global context for video understanding or other related video tasks.

References

[1] Jean-Baptiste Alayrac, Adrià Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. In *NeurIPS*, 2020.

[2] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *NeurIPS*, 2020.

[3] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T. Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. SpeedNet: Learning the Speediness in Videos. In *CVPR*, 2020.

[4] Uta Buchler, Biagio Brattoli, and Bjorn Ommer. Improving Spatiotemporal Self-Supervision by Deep Reinforcement Learning. In *ECCV*, 2018.

[5] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding. In *CVPR*, 2015.

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*. 2020.

[8] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big Self-Supervised Models are Strong Semi-Supervised Learners. *arXiv preprint arXiv:2006.10029*, 2020.

[9] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved Baselines with Momentum Contrastive Learning. *arXiv preprint arXiv:2003.04297*, 2020.

[10] Chen Sun and Fabien Baradel and Kevin Murphy, Cordelia Schmid. Learning Video Representations using Contrastive Bidirectional Transformer. *arXiv preprint arXiv:1906.05743*, 2019.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Ali Diba, Vivek Sharma, Luc Van Gool, and Rainer Stiefelhagen. DynamoNet: Dynamic Action and Motion Network. In *ICCV*, 2019.

[13] Ali Diba, Vivek Sharma, and Luc Van Gool. Deep Temporal Linear Encoding Networks. In *CVPR*, 2017.

[14] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal Cycle-Consistency Learning. In *CVPR*, 2019.

[15] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-Supervised Video Representation Learning with Odd-One-Out Networks. In *CVPR*, 2017.

[16] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. ActionVLAD: Learning Spatio-Temporal Aggregation for Action Classification. In *CVPR*, 2017.

[17] Daniel Gordon, Kiana Ehsani, Dieter Fox, and Ali Farhadi. Watching the World Go By: Representation Learning from Unlabeled Videos. *arXiv preprint arXiv:2003.07990*, 2020.

[18] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" Something Something" Video Database for Learning and Evaluating Visual Common Sense. In *ICCV*, 2017.

[19] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning. *arXiv preprint arXiv:2006.07733*, 2020.

[20] Tengda Han, Weidi Xie, and Andrew Zisserman. Video Representation Learning by Dense Predictive Coding. In *ICCVW*, 2019.

[21] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented Dense Predictive Coding for Video Representation Learning. In *ECCV*, 2020.

[22] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised Co-training for Video Representation Learning. In *NeurIPS*, 2020.

[23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*. 2020.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*. 2016.

[25] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-Time Correspondence as a Contrastive Random Walk. *NeurIPS*, 2020.

[26] Simon Jenni, Givi Meishvili, and Paolo Favaro. Video Representation Learning by Recognizing Temporal Transformations. In *ECCV*, 2020.

[27] Longlong Jing, Xiaodong Yang, Jingen Liu, and Yingli Tian. Self-supervised Spatiotemporal Feature Learning via Video Rotation Prediction. *arXiv preprint arXiv:1811.11387*, 2018.

[28] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The Kinetics Human Action Video Dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[29] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-Supervised Video Representation Learning with Space-Time Cubic Puzzles. In *AAAI*, 2019.

[30] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018.

[31] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: A Large Video Database for Human Motion Recognition. In *ICCV*, 2011.

[32] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised Representation Learning by Sorting Sequences. In *ICCV*, 2017.

[33] Xinyu Li, Yanyi Zhang, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. VidTr: Video Transformer Without Convolutions. In *ICCV*, 2021.

[34] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal Shift Module for Efficient Video Understanding. In *ICCV*, 2019.

[35] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. BMN: Boundary-Matching Network for Temporal Action Proposal Generation. In *ICCV*, 2019.

[36] Yang Liu, Keze Wang, Haoyuan Lan, and Liang Lin. Temporal contrastive graph learning for video action recognition and retrieval. *arXiv preprint arXiv:2101.00820*, 2021.

[37] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video Cloze Procedure for Self-Supervised Spatio-Temporal Learning. In *AAAI*, 2020.

[38] Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei. Unsupervised Learning of Long-Term Motion Dynamics for Videos. In *CVPR*, 2017.

[39] 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.

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

[41] Mandela Patrick, Yuki M Asano, Polina Kuznetsova, Ruth Fong, João F Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal self-supervision from generalized data transformations. *arXiv preprint arXiv:2003.04298*, 2020.

[42] AJ Piergiovanni, Anelia Angelova, and Michael S Ryoo. Evolving losses for unsupervised video representation learning. In *CVPR*, 2020.

[43] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal Contrastive Video Representation Learning. *arXiv preprint arXiv:2008.03800*, 2020.

[44] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *arXiv preprint arXiv:1212.0402*, 2012.

[45] Li Tao, Xueting Wang, and Toshihiko Yamasaki. Self-supervised video representation learning using inter-intra contrastive framework. In *ACM MM*, 2020.

[46] Tian, Yonglong and Krishnan, Dilip and Isola, Phillip. Contrastive Multiview Coding. *arXiv preprint arXiv:1906.05849*, 2019.

[47] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*, 2018.

[48] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Unsupervised Learning of Video Representations using LSTMs. In *ICML*, 2015.

[49] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating Visual Representations from Unlabeled Video. In *CVPR*, 2016.

[50] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-Supervised Spatio-Temporal Representation Learning for Videos by Predicting Motion and Appearance Statistics. In *CVPR*, 2019.

[51] Jiangliu Wang, Jianbo Jiao, and Yunhui Liu. Self-Supervised Video Representation Learning by Pace Prediction. In *ECCV*, 2020.

[52] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In *ECCV*, 2016.

[53] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-Local Neural Networks. In *CVPR*, 2018.

[54] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning Correspondence from the Cycle-Consistency of Time. In *CVPR*, 2019.

[55] Donglai Wei, Joseph J. Lim, Andrew Zisserman, and William T. Freeman. Learning and Using the Arrow of Time. In *CVPR*, 2018.

[56] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-Term Feature Banks for Detailed Video Understanding. In *CVPR*, 2019.

[57] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised Feature Learning via Non-Parametric Instance-level Discrimination. In *CVPR*. 2018.

[58] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-offs in Video Classification. In *ECCV*, 2018.

[59] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-Supervised Spatiotemporal Learning via Video Clip Order Prediction. In *CVPR*, 2019.

[60] Ceyuan Yang, Yinghao Xu, Bo Dai, and Bolei Zhou. Video Representation Learning with Visual Tempo Consistency. In *arXiv preprint arXiv:2006.15489*, 2020.

[61] Ting Yao, Yiheng Zhang, Zhaofan Qiu, Yingwei Pan, and Tao Mei. SeCo: Exploring Sequence Supervision for Unsupervised Representation Learning. In *AAAI*, 2021.

[62] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. Video Playback Rate Perception for Self-supervisedSpatio-Temporal Representation Learning. In *CVPR*, 2020.

[63] Shiwen Zhang, Sheng Guo, Weilin Huang, Matthew R. Scott, and Limin Wang. V4D:4D Convolutional Neural Networks for Video-level Representation Learning. In *ICLR*, 2020.

[64] Yi Zhu, Zhenzhong Lan, Shawn Newsam, and Alexander G Hauptmann. Hidden Two-Stream Convolutional Networks for Action Recognition. In *ACCV*, 2018.

[65] Chengxu Zhuang, Tianwei She, Alex Andonian, Max Sobol Mark, and Daniel Yamins. Unsupervised Learning From Video With Deep Neural Embeddings. In *CVPR*, 2020.