

Cross-Architecture Self-supervised Video Representation Learning

Sheng Guo^{1*} Zihua Xiong^{1*} Yujie Zhong² Limin Wang³ Xiaobo Guo¹ Bing Han¹ Weilin Huang^{4†}

¹MYbank, Ant Group ²Meituan Inc.

³State Key Laboratory for Novel Software Technology, Nanjing University, China ⁴Alibaba Group

{guosheng.guosheng, xiongzihua.xzh, jefflittleguo.gxb, hanbing.hanbing}@mybank.cn

jaszhong@hotmail.com, lmwang@nju.edu.cn, weilin.hwl@alibaba-inc.com

Abstract

In this paper, we present a new cross-architecture contrastive learning (CACL) framework for self-supervised video representation learning. CACL consists of a 3D CNN and a video transformer which are used in parallel to generate diverse positive pairs for contrastive learning. This allows the model to learn strong representations from such diverse yet meaningful pairs. Furthermore, we introduce a temporal self-supervised learning module able to predict an Edit distance explicitly between two video sequences in the temporal order. This enables the model to learn a rich temporal representation that compensates strongly to the video-level representation learned by the CACL. We evaluate our method on the tasks of video retrieval and action recognition on UCF101 and HMDB51 datasets, where our method achieves excellent performance, surpassing the state-of-the-art methods such as VideoMoCo [23] and MoCo+BE [34] by a large margin.

1. Introduction

Video representation learning is a fundamental task for video understanding, as it plays an important role on various tasks, such as action recognition [25, 30, 36, 37, 44], video retrieval [39, 43], and video temporal detection [18, 27, 46]. Recent efforts have been devoted to improving its performance by using deep neural networks in a supervised learning manner, which often requires a large-scale video dataset with very expensive human annotations, such as Sports-1M [1], Kinetics [12], HACS [45], and MultiSports [17]. The large annotation cost inevitably limits the potential of deep networks on learning video representation. Therefore, it is of great importance to improve this task by leveraging unlabeled videos which are easily accessible at a large scale.

Recently, self-supervised learning has made significant progress on learning strong image representation, by constructing various supervised learning signals from images

themselves. It has also been extended to video domain, where contrastive learning has been widely-applied. For example, in recent works such as [23, 28, 35] contrastive learning was introduced to capture the discrimination between two video instances, which enables it to learn an invariant representation within each video instance. However, the contrastive learning mainly focuses on learning a global spatio-temporal representation of videos in these approaches, while it is difficult to capture meaningful temporal details which often provide important cues for discriminating different video instances, e.g., human actions. Therefore, different from learning image representation, modelling temporal information is critical to video representation. In this work, we present a new self-supervised video representation method able to perform both video-level contrastive learning and temporal modelling *simultaneously* in a unique framework, as shown in Figure 1.

The supervised signal of learning temporal information can be created by exploring the sequential nature of videos, allowing for performing self-supervised learning. Recent methods, such as pace predictions [11, 41] and playback speeds perception [4, 42], followed this line of research by creating a pretext task that implements self-supervised temporal predictions. In this work, we introduce a new self-supervised temporal learning by predicting an approximate Edit distance between a video (i.e. a sequence of frames) and its temporal shuffle. This allows us to *explicitly* measure the degree of temporal difference quantitatively in Edit distance, setting it apart from the existing self-supervised methods which are often limited to estimate a rough difference of two videos in the temporal domain. For example, they often created a pretext task to predict whether two video sequences are in the same pace or playback speeds, but ignore details in such temporal difference.

While most self-supervised contrastive learning methods generate positive pairs using various data augmentations which provide different views of an instance, we develop a new method able to learn stronger representation from diverse architectures via contrastive learning. The family of

*Equal contribution.

†Corresponding author.

3D CNNs has achieved remarkable performance in various video tasks, including C3D [30], R3D [8], R(2+1)D [31], etc. They are capable of capturing local dependencies in the temporal domain due to the intrinsic property (i.e., convolutions) of CNNs. But the effective receptive fields of CNNs might limit their ability to modelling long-range dependencies. On the other hand, such long-range dependencies can be naturally captured by the transformer architecture [32] using a self-attention mechanism, where each token is able to learn an attention to the whole sequence, and thus encodes meaningful context information into video representations. Moreover, the inductive biases of CNNs may limit their performance when trained on sufficiently large data, while this limitation may not happen in the transformers due to the dynamic weighting of self-attention [7].

We argue that modelling both local and global dependencies are essential for video understanding; and the inductive biases of CNNs and the capacity of transformers can compensate strongly to each other. In this work, we present a new cross-architecture contrastive learning (CACL) framework for self-supervised video representation learning. Our CACL is able to learn from diverse yet more meaningful contrastive pairs generated by a 3D CNN and a video transformer. We demonstrate that a video transformer can strongly enhance the video representation generated by a 3D CNN. It produces rich high-level contextual features and encourages the 3D CNN to capture more detailed information. This allows the two architectures to work collaboratively, which is the key to boost the performance. Our main contributions can be summarized as follows:

- We design a new cross-architecture contrastive learning (CACL) framework for self-supervised video representation learning. CACL uses a 3DCNN and a Transformer to collaboratively generate diverse yet meaningful positive pairs, which allow for more effective contrastive representation learning.

- We introduce a new self-supervised temporal learning method by explicitly measuring an edit distance between a video and its temporal self-shuffle. This helps to learn rich temporal information complementary to the representation learned from our CACL.

- We verify our method on two downstream video tasks: action recognition and video retrieval. Experimental results on UCF101 [26] and HMDB51 [14] show that the proposed CACL can significantly outperform the state-of-the-art methods, such as VideoMoCo [23] and MoCo+BE [34].

2. Related Work

In this section, we briefly introduce two groups of self-supervised methods for video representation learning: contrastive learning methods and pretext-task approaches. Then we briefly review recent studies on applying transformers in the related video tasks.

Contrastive learning methods. Self-supervised contrastive learning has received extensive attention. The common approach is to maintain a relative consistency between the representations of an instance and its augmented view. It has achieved remarkable success in image-level representation learning, such as SimCLR [5] and MoCo [9], and recent studies extend the contrastive learning paradigm to video applications. For example, VideoMoCo [23] extends image-based MoCo framework to video representation learning by enhancing the temporal robustness of the encoder, and at the same time, modeling the temporal decay of the keys. In [33], Wang *et al.* proposed DSM able to construct positive and negative sample pairs, which alleviate the negative impact of the scene and the motion coupling problem. TCLR [6] was developed by introducing two different temporal contrastive losses, in order to learn temporally distinct features across the video. It is combined with the vanilla instance contrastive loss, leading to an increase in the temporal diversity of the learned features. In [24], CVRL was introduced by studying various data augmentations for video self-supervised learning. Then a temporally consistent spatial augmentation and a sampling-based temporal augmentation methods were proposed. These approaches perform both spatial-temporal data augmentations to construct positive samples for contrastive learning. In this work, we generate the positive sample pairs from two different network architectures able to learn two diverse video embeddings that focus on learning 3D local video details or long-range temporal dependencies. We demonstrate that the transformer architecture can provide the positive samples with more diversity for the 3D CNN.

Pretext task based approaches. It is intuitive to exploit temporal dimension to generate pseudo training labels without human annotations. Many recent works rely on investigating the speed property of videos, such as pace predictions [11,41], playback speeds perception [4,42], and speediness of moving objects [2]. On the other hand, some recent studies focus on learning from video temporal order e.g., by verifying the correctness of temporal order of a video. In [20] Misra *et al.* created a sequential verification task by comparing a video sequence with its shuffled version, and then OPN [15] extends the sequential verification to order prediction. This allows it to generate richer learning signals that enables the model to learn more temporal details. In [40], VCOP was proposed to learn the spatio-temporal representation of the video by predicting the order of shuffled clips from a video. However, the degree of a video temporal shuffle has not been considered among all these methods. In this work, we propose to use the degree of temporal order as supervision signals which enable the model to learn more detailed temporal information.

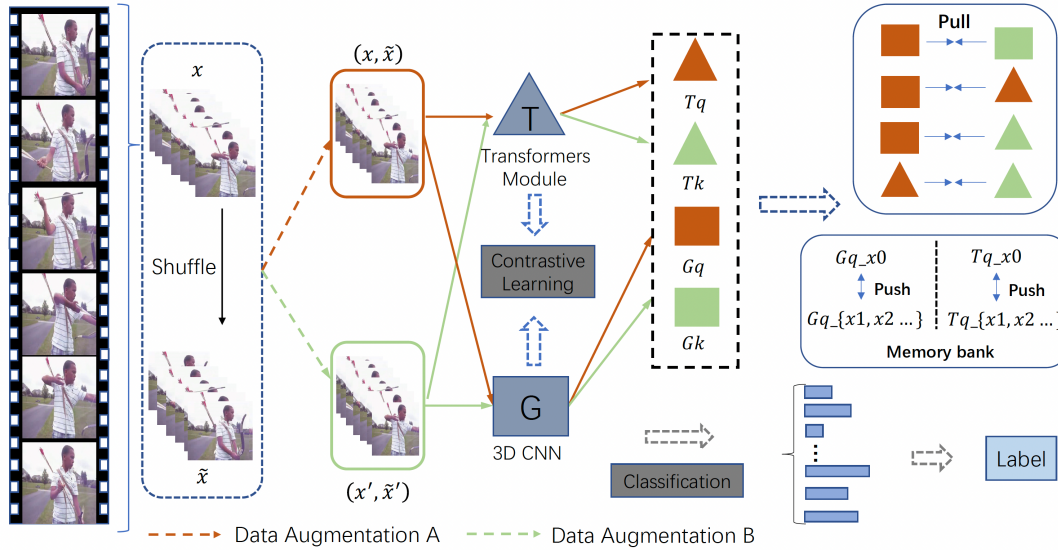


Figure 1. Illustration of the proposed CACL framework. (i) Given a video clip x , we implement random shuffle on temporal dimension to generate a shuffled version \tilde{x} , with random data augmentations applied. (ii) Then the original clip and the shuffled one are concatenated and passed to a transformer encoder, generating a transformer-based video representation. At the same time, x and \tilde{x} are computed by the 3DCNN encoder separately to generate a 3D video representation. In total, the two different encoders output four different representations of the input video, resulting in four positive pairs for constractive learning. (iii) Furthermore, the two 3D representations are concatenated, and are used for self-supervised temporal learning by predicting an Edit distance, which can be reformulated as a classification task. We demonstrate that the temporal representation learned by our method can compensate strongly to our video-level representation (by CACL).

Transformers. The transformer [32] has recently achieved remarkable performance in various computer vision tasks. For example, ViT [7] and DeiT [29] attempted to perform image recognition by using transformers. They split an image into a set of image patches, generating a sequence of patch embeddings which are used as an input to the transformer. Both methods achieve excellent results compared to state-of-the-art convolutional networks. Besides, Carion *et al.* proposed DETR [3], where a direct set prediction approach is equipped with a transformer encoder-decoder architecture, yielding excellent results on par with the well-established CNN-based object detector.

With the sequential nature of videos, it is natural to apply the transformer for better modelling video related tasks. Recently, Wang *et al.* developed VisTR [38] which is a simpler and faster video instance segmentation framework based on transformers. In [21], VTN was presented, consisting of a transformer module to process spatial-temporal information. This started a new line of research in video recognition domain. Our CACL further explore the potential of transformer in modeling long-range dependencies, and the architecture discrepancy between the transformer and convolutional networks. This allows us to build a transformer video encoder that provides rich compensatory features for 3D CNN in contrastive learning.

3. Methodology

We tackle video representation learning problem in a self-supervised manner. In this section, we first present an overall framework of the proposed method. Then we describe details of the proposed contrastive learning method, and our self-supervised temporal learning based on a prediction of frame-level shuffle degree.

3.1. Overall Framework

Our framework consists of two pathways including a transformer video encoder and a 3D CNN video encoder. The self-supervised learning signals are computed from two tasks: clip-level contrastive learning and frame-level temporal prediction.

3D CNN video encoder. In this work, 3D CNNs (such as C3D [30], R3D [8] and R(2+1)D [31]) are adopted as the main video encoder, which are also used for inference. Notice that any other 3D CNN architectures can also be applied in our framework. The output features of original clip and shuffled clip are concatenated and then passed through the contrast head and the classification head. Both heads are fully connected feed-forward networks.

Transformer video encoder. The transformer encoder consists of a 2D CNN and a transformer architecture, which

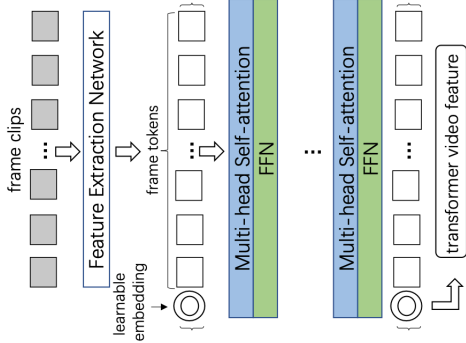


Figure 2. Transformer video encoder architecture.

is motivated by VTN [21]. The pipeline is illustrated in Figure 2. Firstly, individual image frames of a video clip are computed through a 2D CNN which performs feature extraction to obtain a sequence of frame-level tokens. The output CNN features are then projected to 768-D frame tokens through a fully connected layer. Then the frame tokens are concatenated sequentially in a temporal order, and a learnable embedding is prepended to the sequence of frame tokens. Finally, a 6-layer 6-head transformer model takes the clip-level feature sequence as input, and the output of learnable embedding is used as video representation. Notably, the feature extraction network is ResNet50 [10] pretrained with video frames from UCF101 train set, by using self-supervised method MoCo [9]. The pretrain details will be illustrated in 4.1, and its weights are frozen during self-supervised video representation learning.

3.2. Cross-Architecture Contrastive learning

The goal of our self-supervised contrastive learning is to maximize the similarity between the video clips with same context, while minimizing the similarity of clips sampled from different videos. Different from previous contrastive learning methods, our CACL leverages cross-architecture contrastive learning signals to better capture both local and long-range dependencies jointly.

Construction of positive pairs. The fundamental problem of contrastive learning lies in the design of positive and negative samples. Previous works on self-supervised contrastive learning often utilize various data augmentations to generate different transformed versions of a particular instance (e.g. different clips of the same video) which forms the positive pairs. In this work, we enrich the positive pairs from two perspectives: embedding level (using different architectures) and data level. From the perspective of architecture, CACL takes the advantage of a 3D CNN and a transformer. Given an input video clip, each of them produces a video representation separately, which doubles the number of positive samples comparing to previous methods.

For the data level, we apply a random shuffle on temporal dimension on the original clip x and obtain a shuffled video clip \tilde{x} . The two instances are latter concatenated. As illustrated in Figure 1, we maximize the similarities of four positive pairs generated from each video clip, by using different data augmentations and different encoders. We denote different data augmentations as q, k , a transformer encoder as T , a 3D CNN encoder as G , then we can generate four feature representations Gq, Gk, Tq, Tk , for a video clip.

Negative pairs. Clips from different videos are considered as negative samples. We further enhance the contrastive learning by using a momentum encoder and a memory dictionary queue motivated by MoCo [9], providing more meaningful negative samples for improving the performance of contrastive learning.

Data augmentations. Data augmentations are performed in both spatial and temporal domains on the input video clips. Note that the spatial augmentation is performed consistently across all frames within a clip. Therefore, we maximize three kinds of similarities: (1) the similarity between the clips computed by the same network but performed different data augmentations; (2) the similarity between the clips with the same data augmentation but computed by different networks; (3) the similarity between the clips using different networks with different data augmentations.

Loss function. Formally, we consider a randomly sampled mini-batch consisting of N different video instances, and then we sample one clip from each video starting a random timestamp with an equal sampling rate. This results in a total of N clips (C) in a mini-batch. We randomly shuffle the order of each clip, yielding a new set of N clips (C_s). Then each clip and its shuffled version are concatenated, and are further processed with data augmentations. This generates two concatenated video clips only having different data augmentations. The generated clips are separately processed by different video encoders: a 3D-CNN based video encoder and a transformer-based encoder. Therefore, we generate four clip-level video representation for each video instance: Gq_i, Gk_i, Tq_i, Tk_i , which are used to construct positive pairs during contrastive learning. We leverage the idea of instance discrimination using InfoNCE [22] based contrastive loss.

$$\mathcal{L}_{\text{NCE}} = -\frac{1}{n} \sum_{i=1}^n \log \frac{e^{\text{cont}(Gq_i, Gk_i)}}{\sum_{j=1}^n e^{\text{cont}(Gq_i, Gk_j)}}, \quad (1)$$

$$\text{cont}(Gq, Gk) = \tau * \frac{G(q)}{\|G(k)\|} \cdot \frac{G(q)}{\|G(K)\|}, \quad (2)$$

where $\text{cont}(Gq, Gk)$ is the similarity metric between two vectors. Gq and Gk are two feature representations. τ is

an adjustable parameter. In this work, we extend the contrastive learning for video representation learning as:

$$S_{\text{pos}}^i = e^{\text{cont}(Gq_i, Gk_i)} + e^{\text{cont}(Gq_i, Tq_i)} + e^{\text{cont}(Gq_i, Tk_i)} + e^{\text{cont}(Tq_i, Tk_i)}, \quad (3)$$

$$\mathcal{L}_{\text{NCE}} = -\frac{1}{n} \sum_{i=1}^n \log \frac{S_{\text{pos}}^i}{S_{\text{pos}}^i + \sum_{j=1}^m e^{\text{cont}(Gq_i, N_j)}}, \quad (4)$$

where N_j is a negative sample from a memory dictionary queue with a queue size of m . As shown in Equation 4, our CACL is able to generate more positive pairs than the standard contrastive learning.

3.3. Temporal Prediction with Edit Distance

We aim to learn video representations that are sensitive to temporal details. To this end, we attempt to train the network by explicitly predicting the temporal difference between a video clip and its shuffled version. We assume that such a temporal prediction task requires both motion and appearance cues. This enables the model to learn meaningful temporal details which in turn benefit downstream tasks. In this work, we propose to use a Minimum Edit Distance (MED) to measure the degree of temporal difference between a video clip and its shuffled version.

MED was originally introduced in [16] by Vladimir Levenshtein in 1965, and it provides a way to measure the dissimilarity between two strings (e.g., words), by counting the minimum number of operations required to transform one string into the other. Mathematically, the Levenshtein distance between two strings a, b is given by $\text{lev}_{a,b}(|a|, |b|)$ where

$$\text{lev}_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0, \\ \min \begin{cases} \text{lev}_{a,b}(i-1, j) + 1 \\ \text{lev}_{a,b}(i, j-1) + 1 \\ \text{lev}_{a,b}(i-1, j-1) + I(a_i \neq b_j) \end{cases} & \text{otherwise.} \end{cases} \quad (5)$$

where $I(a_i \neq b_j)$ is an indicator function which is equal to 0 when $a_i = b_j$, and is 1 otherwise. In this work, the shuffle degree prediction task is formulated as a classification problem, and the 3D CNN model $f(x, \tilde{x})$ is trained with a cross entropy loss. Given a video clip x and its shuffled version \tilde{x} , we have:

$$\mathcal{L}_{\text{cls}} = -\sum_{i=1}^m y_i \log \frac{e^{h_i}}{\sum_{j=1}^m e^{h_j}}, h = f(x, \tilde{x}), \quad (6)$$

where m is the number of all the shuffle degree candidates.

Uniform shuffle-degree sampling. Given a video clip with 16 frames, we random shuffle it and compute a MED between the original clip and the shuffled one. Interestingly, we found that in our case the MED is a discrete integer ranging from 0 to 16 (except 1), which allows us to reformulate the regression problem of MED prediction into a classification task. However, the distribution of such discrete integers is not uniform, which may result in a unbalanced classification, making the training process unstable. Technically, we first randomly sample a MED number from a uniform distribution, and then randomly shuffle the video clip until it satisfies the sampled MED number. This operation allows us to well balance label distribution in classification, which is important to temporal modelling and joint learning.

Compared with other shuffle&learn methods. In contrast to earlier methods such as Shuffle&Learn [20], OPN [15] and VCOP [40]. Our method focuses on degree perception rather than order prediction/verification, which naturally leads to the following characteristics. It can learn more meaningful temporal information by increasing the number of frames, while previous methods are commonly limited to using a very small number of frames, e.g., 3 frames/clips. Because the number of orders overgrows with the increase in the number of frames/clips. Our method can capture more detailed and meaningful difference between video clips, allowing for learning richer temporal features.

4. Experiments and Results

4.1. Implementation Details

Datasets. UCF101 [26] is an action recognition dataset which consists of 13,320 videos from 101 realistic action categories on YouTube. Following previous works [2, 40], we use training split 1 for our self-supervised learning, training/testing split 1 for evaluating video retrieval task, and all training/testing splits for action recognition. HMDB51 [14] dataset consists of 6,849 clips from 51 action classes, collected from a variety of sources such as digitized movies and YouTube. Following [2, 40], we use training/testing split 1 to perform video retrieval, and all training/testing splits for action recognition.

Pre-training feature extraction network. We train our feature extraction network applied in video transformer by using self-supervised method MoCo [9]. We follow the training policy with the same practice as in MoCo. To avoid introducing additional data, we form our training dataset by sampling frames from UCF101 training split 1. Specifically, we sample frames from a video with 10 intervals in order to enlarge inter-frame differentiation. With around 1.8M total frames in UCF101 training split 1, we can collect a training dataset consisting of approximately 180k frames.

Method	1	5	10	20	50
VCOP (C3D) [40]	12.5	29.0	39.0	50.6	66.9
PRP (C3D) [42]	23.2	38.1	46.0	55.7	68.4
SDP*	17.2	34.6	45.7	57.8	74.3
SDP	22.1	40.6	50.3	61.2	75.5
V-MoCo	30.3	47.4	57.0	66.3	78.9
SDP+V-MoCo	36.2	55.0	64.5	72.8	83.2
V-MoCo+T	39.1	53.3	61.8	72.2	85.0
SDP+V-MoCo+T	43.2	61.1	69.9	78.2	88.2

Table 1. Ablation Studies on video retrieval (on UCF101). SDP refers to shuffle degree prediction, SDP* means SDP without uniform sampling, and V-MoCo is a video-level contrastive learning by simply implementing MoCo with a 3D CNN encoder. +T refers to a joint video transformer encoder.

positive pairs	1	5	10	20	50
(Gq, Gk)	36.2	55.0	64.5	72.8	83.2
$(Gq, Gk), (Gq, Tq)$	37.8	54.9	64.2	73.8	84.4
$(Gq, Gk), (Gq, Tq), (Gq, Tk)$	39.3	58.2	67.0	75.8	85.6
$(Gq, Gk), (Gq, Tq), (Gq, Tk), (Tq, Tk)$	43.2	61.1	69.9	78.2	88.2

Table 2. Video retrieval with different positive pairs.

Self-supervised learning. We train the network with our CACL framework for 300 epochs, and adopt SGD as our optimizer with a momentum of 0.9 and a weight decay of $5e-4$. We set a batch size as 64. A learning rate is initialized as 0.001 with a cosine learning rate schedule used. We sample 16 frames with 2 intervals from a video. Video frames are first resized to 128×171 , and then randomly cropped to 112×112 . Meanwhile, we apply consistent data augmentations, such as horizontal flipping, color jittering, and random gaussian blur to all frames within a same video clip.

Supervised fine-tuning and evaluation. After self-supervised learning, we transfer the weights of 3D CNN encoder network to action recognition. We fine-tune it on each dataset for 150 epochs. The momentum is set as 0.9, and the batch size is 128. The learning rate is initialized as 0.1 with a cosine learning rate schedule applied. For evaluation, by following common practice, the final result of a video is the average of the results of 10 clips uniformly sampled from the video. As the self-supervised training, we sample 16 frames with 2 intervals to form a clip.

4.2. Ablation Study

To evaluate our CACL on self-supervised video representation learning, we conduct video retrieval experiments on UCF101 to verify the effectiveness of each individual component developed in the CACL. We apply C3D as 3D CNN in this experiment. Results show that all the developed components make clear contributions.

Net	(Gq, Gk)	(Gq, Tq)	(Gq, Tk)	(Tq, Tk)
C3D	0.8383	0.7844	0.6487	0.8534
R(2+1)D	0.8174	0.7773	0.6374	0.843
R3D	0.8128	0.7639	0.628	0.8427

Table 3. Average similarities on UCF101.

Shuffle degree prediction. We investigate the capability of our shuffle degree prediction (SDP) on learning temporal information from a video, and compare it with recent VCOP [40] and PRP [42] which were developed specifically for self-supervised temporal learning. Results are compared in Table 1, where our SDP outperforms VCOP considerably and achieves comparable results with PRP. This verifies the effectiveness of our SDP by predicting an Edit distance between two videos. This enables it to identify more differences in temporal details which naturally aggregate more meaningful temporal information. The result of SDP* which means SDP without uniform sampling, demonstrates the importance of uniform shuffle-degree sampling.

We further perform video-level contrastive learning by simply implementing MoCo with C3D networks (referred as V-MoCo), which is able to learn spatio-temporal features for video-level representation. As shown in Table 1, it obtains higher performance than SDP which focuses on learning temporal information from videos, demonstrating that video-level contrastive learning is able to learn stronger video-level representation in general. Interestingly, by integrating our SDP with V-MoCo, the performance can be further improved considerably, suggesting that the temporal representation learned by our SDP can compensate strongly to general video-level representation. Therefore, SDP can work collaboratively with general video representation learning architecture, and plays an important role by capturing temporal features like an optical flow branch used in the widely-applied two-stream design [25].

Transformer video encoder. In our video-level contrastive learning (V-MoCo), a key improvement is to use a video transformer encoder *jointly* with 3D CNN encoder, which are able to generate more diverse yet meaningful positive pairs. Table 1 shows that our method achieves a large improvement by using the transformer encoder, demonstrating the effectiveness of our cross-architecture design with the transformer encoder. As shown in Table 2, (Gq, Gk) means positive pairs by 3D CNN with different data augmentations, which is equal to performing the original MoCo on videos with our SDP. Using all possible positive pairs $(Gq, Gk), (Gq, Tq), (Gq, Tk), (Tq, Tk)$ is a full implementation of our CACL. *The performance can be improved gradually by adding more groups of the positive pairs.* This demonstrates that our video transformer encoder can provide more meaningful contrast information.

Method	Net	UCF101					HMDB51				
		1	5	10	20	50	1	5	10	20	50
VCOP [40]	C3D	12.5	29.0	39.0	50.6	66.9	7.4	22.6	34.4	48.5	70.1
Pace Pred [11]	C3D	31.9	49.7	59.2	68.9	80.2	12.5	32.2	45.4	61.0	80.7
DSM [33]	C3D	16.8	33.4	43.4	54.6	70.7	8.2	25.9	38.1	52.0	75.0
RSPNet [4]	C3D	36.0	56.7	66.5	76.3	87.7	-	-	-	-	-
CACL	C3D	43.2	61.1	69.9	78.2	88.2	17.3	40.2	53.8	68.0	85.2
CACL*	C3D	44.2	63.1	71.9	80.4	89.4	-	-	-	-	-
VCOP [40]	R3D	14.1	30.3	40.0	51.1	66.5	7.6	22.9	34.4	48.8	68.9
Pace Pred [11]	R3D	23.8	38.1	46.4	56.6	69.8	9.6	26.9	41.1	56.1	76.5
CACL	R3D	41.1	59.2	67.3	75.2	85.9	17.6	36.7	48.4	63.0	81.8
VCOP [40]	R(2+1)D	10.7	25.9	35.4	47.3	63.9	5.7	19.5	30.7	45.8	67.0
Pace Pred [11]	R(2+1)D	25.6	42.7	51.3	61.3	74.0	12.9	31.6	43.2	58.0	77.1
CACL	R(2+1)D	41.5	59.7	68.4	77.6	87.8	16.4	38.0	49.6	63.4	81.1

Table 4. **Recall-at-topK (%)**. Video retrieval performance under different K values on UCF101 and HMDB51. CACL* means pre-trained on Kinetics400 dataset

Method	Year	input size	Pretrained	Architecture	Depth	UCF101	HMDB51
Puzzle [13]	AAAI'19	16 × 112	UCF101	C3D	10	65.0	31.3
VCOP [40]	CVPR'19	16 × 112	UCF101	C3D	10	65.6	28.4
VCP [19]	AAAI'20	16 × 112	UCF101	C3D	10	68.5	32.5
PRP [42]	CVPR'20	16 × 112	UCF101	C3D	10	69.1	34.5
RSPNet [4]	AAAI'21	16 × 112	Kinetics400	C3D	10	76.7	44.6
DSM [33]	AAAI'21	16 × 112	UCF101	C3D	10	70.3	40.5
MoCo+BE [34]	CVPR'21	16 × 112	UCF101	C3D	10	72.4	42.3
CACL	-	16 × 112	UCF101	C3D	10	77.2	43.5
CACL	-	16 × 112	Kinetics400	C3D	10	77.5	-
Puzzle [13]	AAAI'19	16 × 112	UCF101	R3D	18	65.8	33.7
VCOP [40]	CVPR'19	16 × 112	UCF101	R3D	10	64.9	29.5
VCP [19]	AAAI'20	16 × 112	UCF101	R3D	10	66.0	31.5
PRP [42]	CVPR'20	16 × 112	UCF101	R3D	10	66.5	29.7
IIC [28]	ACMMM'20	16 × 112	UCF101	R3D	10	74.4	38.8
RSPNet [4]	AAAI'21	16 × 112	Kinetics400	R3D	18	74.3	41.8
VideoMoCo [23]	CVPR'21	16 × 112	Kinetics400	R3D	18	74.1	43.6
CACL	-	16 × 112	UCF101	R3D	10	77.5	43.8
VCOP [40]	CVPR'19	16 × 112	UCF101	R(2+1)D	10	72.4	30.9
VCP [19]	AAAI'20	16 × 112	UCF101	R(2+1)D	10	66.3	32.2
PRP [42]	CVPR'20	16 × 112	UCF101	R(2+1)D	10	72.1	35.0
Pace Pred [35]	ECCV'20	16 × 112	UCF101	R(2+1)D	10	75.9	35.9
Pace Pred [35]	ECCV'20	16 × 112	Kinetics400	R(2+1)D	10	77.1	36.6
RSPNet [4]	AAAI'21	16 × 112	Kinetics400	R(2+1)D	10	81.1	44.6
VideoMoCo [23]	CVPR'21	16 × 112	Kinetics400	R(2+1)D	18	78.7	49.2
CACL	-	16 × 112	UCF101	R(2+1)D	10	82.5	48.8

Table 5. The top-1 accuracy (%) on UCF101 and HMDB51 dataset. The accuracy is computed by averaging over three splits.

Positive pairs. To further study the impact of diverse positive pairs to self-supervised contrastive learning, we compute average similarities of the positive pairs on UCF101 test split 1, as reported in Table 3. We can have a number of observations as follows: (1) by comparing (Gq, Gk) with (Gq, Tq) , we found that a same video clip performed with different data augmentations can have a higher similarity than that computed by different encoders. This suggests that different encoders can generate positive pairs with

more diversity, which are often meaningful and compensate to those generated by 3D CNN in contrastive learning. Therefore, our cross-architecture can boost the performance of self-supervised contrastive learning; (2) as expected, (Gq, Tk) has the smallest similarity by using different data augmentations and different encoders, which may indicate the largest diversity and in turn has a larger improvement than that of (Gq, Tq) , as shown in Table 2; (3) by using a same encoder like (Gq, Gk) or (Tq, Tk) , the trans-

former encoder has a higher similarity than that of 3D CNN. But it still obtains a considerable improvement by providing compensatory information to 3D CNN (shown in Table 2).

4.3. Video Retrieval

In this section, we evaluate CACL on video retrieval task. Following VCOP [40], the evaluation is performed on the split 1 of UCF101 dataset. The network is fixed as a feature extractor after performing pre-training on the split 1 training set of UCF101. We sample ten 16-frames clips from each video, and then perform feed-forward computation to generate features from the last pooling layer (p5). We apply spatial max-pooling on each clip to get clip-level features, and perform average-pooling over 10 clips to obtain a video-level representation. We use videos from the test set (query videos) to retrieve the videos from the training set by computing cosine similarities. Once a video having a same category with the query video appears in Top-k retrieval results, we consider it as a correct Top-k hit. For a fair comparison, we apply the Top-k accuracy (k=1, 5, 10, 20, 50) as evaluation metrics. We report our results on UCF101 and HMDB51, and compare our method with recent self-supervised methods in Table 4.

Our method outperforms the state-of-the-art methods on all evaluation metrics by a large margin. For example, with C3D, our method obtains a 43.2% top-1 accuracy on UCF101, which outperforms recent RSPNet [4] by 7.2%. Our CACL also has higher performance than other recent methods with different network architectures, demonstrating the strong generalization ability of our method.

4.4. Action Recognition

We further compare our method with recent self-supervised methods on action recognition in Table 5. For a fair comparison, we conduct experiments on three widely-used 3D CNN backbones applied in previous works. Detailed configurations such as *input size*, *pretrain dataset* and *backbone architecture* are listed in Table 5. We report classification top-1 accuracy on UCF101 and HMDB51 datasets, by computing an average classification accuracy over 3 test splits. Our CACL with C3D and R3D can achieve the state-of-the-art results on both datasets. For C3D, our method outperforms recent DSM [33] considerably by 6.9% and 3% on UCF101 and HMDB51 respectively. By comparing with VCOP, our method increases the accuracy largely from 65.6% to 77.2% on UCF101, and from 28.4% to 43.5% on HMDB51. For R3D and R(2+1)D networks pre-trained on UCF101, our method achieves 77.5% and 82.5% on UCF101, 43.8% and 48.8% on HMDB51, consistently outperforming RSPNet [4] pre-trained on Kinetics400.

The performance on action recognition demonstrates the effectiveness of our method. CACL not only learns more

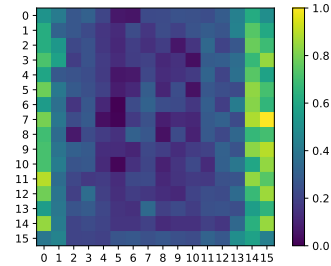


Figure 3. Attention map of the first layer in the trained video transformer. The numbers on both axis denote the indices of the tokens. Note that the attentions are normalized for better visualization.

meaningful spatio-temporal representations by introducing a transformer encoder into self-supervised contrastive learning, but also develops a new temporal prediction of Edit distance which allows it to aggregate rich temporal information that compensates to video-level spatio-temporal representation. Remarkably, UCF101 is a relatively small dataset compared to Kinetics400, CACL pre-trained on UCF101 even beats some recent methods pre-trained on Kinetics400, which indicates a great potential of our methods.

4.5. Transformer Attention Visualization

To justify the motivation of contrasting representations of CNNs and transformers, we study the attention range of the trained video transformer in CACL. We visualize the attention maps of the first layer in the video transformer, averaged over 50 randomly selected video clips. As shown in Figure 3, most frame tokens tend to pay more attention on the first and last positions than the middle ones of the clip. This is in line with our intuition that transformer can capture long-range dependencies, in contrast to the locality of 3DCNNs. Therefore, learning cross-architecture representations can compensate each other and achieve better performance.

5. Conclusion

We have presented a new self-supervised video representation learning framework named CACL. We design a contrastive learning framework by introducing a transformer video encoder, which provide plentiful positive samples for 3D CNN in contrastive learning. We also introduce a new pretext which train a model to predict video shuffle degree. To verify the effectiveness of our approach, we conducted extensive experiments across three network architectures on two different downstream tasks. The experimental results indicates that our shuffle degree prediction and transformer video encoder can encourage model to learn transferable video representations, the learned feature is heterogeneous with contrastive learning based method.

Acknowledgements. This work is supported by National Natural Science Foundation of China (No.62076119, No.61921006).

References

- [1] Karpathy Andrej, Toderici George, Shetty Sanketh, Leung Thomas, Sukthankar Rahul, and Fei-Fei Li. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. [1](#)
- [2] 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. [2](#), [5](#)
- [3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020. [3](#)
- [4] Peihao Chen, Deng Huang, Dongliang He, Xiang Long, Runhao Zeng, Shilei Wen, Mingkui Tan, and Chuang Gan. Rspnet: Relative speed perception for unsupervised video representation learning. In *AAAI*, 2021. [1](#), [2](#), [7](#), [8](#)
- [5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICLR*, 2020. [2](#)
- [6] Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. Tclr: Temporal contrastive learning for video representation. *arXiv preprint arXiv:2101.07974*, 2021. [2](#)
- [7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. [2](#), [3](#)
- [8] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, 2018. [2](#), [3](#)
- [9] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. [2](#), [4](#), [5](#)
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. [4](#)
- [11] Wang Jiangliu, Jiao Jianbo, and Liu Yun-Hui. Self-supervised video representation learning by pace prediction. In *ECCV*, 2020. [1](#), [2](#), [7](#)
- [12] 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. [1](#)
- [13] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, 2019. [7](#)
- [14] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011. [2](#), [5](#)
- [15] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, 2017. [2](#), [5](#)
- [16] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966. [5](#)
- [17] Yixuan Li, Lei Chen, Runyu He, Zhenzhi Wang, Gangshan Wu, and Limin Wang. Multisports: A multi-person video dataset of spatio-temporally localized sports actions. In *ICCV*, pages 13516–13525. [1](#)
- [18] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. BMN: boundary-matching network for temporal action proposal generation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 3888–3897. IEEE, 2019. [1](#)
- [19] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video cloze procedure for self-supervised spatio-temporal learning. *AAAI*, 2020. [7](#)
- [20] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016. [2](#), [5](#)
- [21] Daniel Neimark, O. Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021. [3](#), [4](#)
- [22] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. [4](#)
- [23] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *CVPR*, 2021. [1](#), [2](#), [7](#)
- [24] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *CVPR*, 2021. [2](#)
- [25] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. [1](#), [6](#)
- [26] 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. [2](#), [5](#)
- [27] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. In *ICCV*, pages 13506–13515, 2021. [1](#)
- [28] Li Tao, Xueting Wang, and Toshihiko Yamasaki. Self-supervised video representation learning using inter-intra contrastive framework. In *ACM International Conference on Multimedia*, 2020. [1](#), [7](#)
- [29] Hugo Touvron, M. Cord, M. Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2021. [3](#)
- [30] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. [1](#), [2](#), [3](#)
- [31] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. [2](#), [3](#)

- [32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. [2](#), [3](#)
- [33] Jinpeng Wang, Yuting Gao, Ke Li, Xinyang Jiang, Xiao-Wei Guo, Rongrong Ji, and Xing Sun. Enhancing unsupervised video representation learning by decoupling the scene and the motion. In *AAAI*, 2021. [2](#), [7](#), [8](#)
- [34] Jinpeng Wang, Yuting Gao, Ke Li, Yiqi Lin, Andy J Ma, Hao Cheng, Pai Peng, Feiyue Huang, Rongrong Ji, and Xing Sun. Removing the background by adding the background: Towards background robust self-supervised video representation learning. In *CVPR*, 2021. [1](#), [2](#), [7](#)
- [35] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In *ECCV*, 2020. [1](#), [7](#)
- [36] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. TDN: temporal difference networks for efficient action recognition. In *CVPR*, pages 1895–1904, 2021. [1](#)
- [37] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(11):2740–2755, 2019. [1](#)
- [38] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. *arXiv preprint arXiv:2011.14503*, 2020. [3](#)
- [39] Zhenzhi Wang, Limin Wang, Tao Wu, Tianhao Li, and Gangshan Wu. Negative sample matters: A renaissance of metric learning for temporal grounding. In *AAAI*, 2022. [1](#)
- [40] 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. [2](#), [5](#), [6](#), [7](#), [8](#)
- [41] Ceyuan Yang, Yinghao Xu, Bo Dai, and Bolei Zhou. representation learning with visual tempo consistency. In *arXiv preprint arXiv:2006.15489*, 2020. [1](#), [2](#)
- [42] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. Video playback rate perception for self-supervised spatio-temporal representation learning. In *CVPR*, 2020. [1](#), [2](#), [6](#), [7](#)
- [43] Bowen Zhang, Hexiang Hu, Joonseok Lee, Ming Zhao, Sheide Chammas, Vihan Jain, Eugene Ie, and Fei Sha. A hierarchical multi-modal encoder for moment localization in video corpus. *CoRR*, abs/2011.09046, 2020. [1](#)
- [44] Bowen Zhang, Jiahui Yu, Christopher Fifty, Wei Han, Andrew M. Dai, Ruoming Pang, and Fei Sha. Co-training transformer with videos and images improves action recognition. *CoRR*, abs/2112.07175, 2021. [1](#)
- [45] Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan. Hacs: Human action clips and segments dataset for recognition and temporal localization. In *ICCV*, 2019. [1](#)
- [46] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *ICCV*, pages 2933–2942, 2017. [1](#)