Contrast and Order Representations for Video Self-supervised Learning

Kai Hu[1,3], Jie Shao[2,3], Yuan Liu[3], Bhiksha Raj[1], Marios Savvides[1], Zhiqiang Shen[1]
[1]Carnegie Mellon University, [2]Fudan University, [3]ByteDance
{kaihu, bhiksha, zhiqians, marioss}@andrew.cmu.edu
shaojie@fudan.edu.cn, liuyuan_merry@connect.hku.hk

Abstract

This paper studies the problem of learning self-supervised representations on videos. In contrast to image modality that only requires appearance information on objects or scenes, video needs to further explore the relations between multiple frames/clips along the temporal dimension. However, the recent proposed contrastive-based self-supervised frameworks do not grasp such relations explicitly since they simply utilize two augmented clips from the same video and compare their distance without referring to their temporal relation. To address this, we present a contrast-and-order representation (CORP) framework for learning self-supervised video representations that can automatically capture both the appearance information within each frame and temporal information across different frames. In particular, given two video clips, our model first predicts whether they come from the same input video, and then predict the temporal ordering of the clips if they come from the same video. We also propose a novel decoupling attention method to learn symmetric similarity (contrast) and anti-symmetric patterns (order). Such design involves neither extra parameters nor computation, but can speed up the learning process and improve accuracy compared to the vanilla multi-head attention. We extensively validate the representation ability of our learned video features for the downstream action recognition task on Kinetics-400 and Something-something V2. Our method outperforms previous state-of-the-arts by a significant margin.

1. Introduction

In recent years, self-supervised learning methods have become increasingly popular for a number of problems, including masked language models in natural language processing [7], and jigsaw solving [35], rotation prediction [15], contrastive learning [20, 5], etc. For vision tasks, to ameliorate the dependence on large amounts of manually annotated data required by fully-supervised learning methods. Among the self-supervised methods in visual tasks, contrastive learning [20, 5] has shown great potential on *image* tasks; the transferability of the learned model often even exceeds supervised models for many popular image downstream tasks like detection [38], segmentation [31] and key point estimation [21].

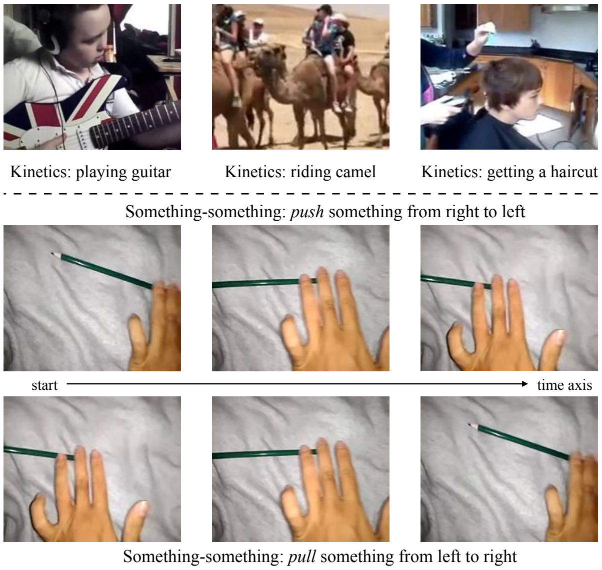
Kinetics: playing guitar
Kinetics: riding camel
Kinetics: getting a haircut
Something-something: *push* something from right to left
start
time axis
Something-something: *pull* something from left to right

Figure 1. Motivation of this work. Some videos can be recognized by the simple appearance information in a single frame, e.g., "playing guitar","riding camel", and "getting a haircut" categories in the Kinetics dataset. Some videos of actions have similar appearance, and require complex temporal-level relation understanding. For example, if we play a video of the action "*push* something from right to left" in reverse, it will not be "*push* something from left to right", but "*pull* something from left to right".

The challenges of labeling are greater for videos, as opposed to static images, since videos include a time dimension, making them more expensive to collect and annotate. Consequently, there is greater need for powerful and practical self-supervised learning algorithms to analyze videos. Recently, [19] and [37] have applied contrastive learning methods to the task of learning representations of videos. In these frameworks, their objective mainly aims to pull representations of two augmented clips from the same video closer in the embedding space, while those from clips originating from different videos are pushed farther apart.

However, we argue that directly employing these approaches (or simple extensions) is not enough for learning sufficiently detailed information from videos through self-supervised learning, since video-analysis tasks are more complex than image tasks as shown in Fig. 1. Besides capturing the static appearance information within each frame (e.g., playing guitar, riding a camel), video learning also needs to understand the relations between multiple frames/-clips (e.g., distinguishing *push* and *pull* actions). [48, 49, 55] shows that temporal relations are vital for video learning, while *current contrastive solutions do not explicitly involve temporal modeling processes*. They only utilize two video clips from the same video and force representations from different timestamps to be similar.

To address this limitation, we propose a *Contrast and Order RePresentation* (**CORP**) framework to incorporate temporal modeling into the self-supervised learning task. Our idea is conceptually simple: given two video clips, our model first learns whether they come from the same video (*contrast*), and then classifies which clip happens earlier if from the same video (*order*). The contrast module extracts the appearance information, such as shapes and edges while the order module models temporal reasoning.

Concretely, we propose our method with two distinct implementations. The first implementation, called CORP$_m$, is illustrated in Figure 2 left (More details will be given later in the paper). Here, we randomly sample $2K$ augmented video clips from two videos (K clips per video) to form $K(2K-1)$ ordered pairs. For each two-clip pair, there are three possible relations between the two clips: 1) they are not from the same videos; 2) they are from the same video, and the first clip in the pair precedes the second in time; 3) they are from the same video, and the second clip in the pair is the one that occurs first in time. Our model is trained to minimize the classification error.

The second implementation CORP$_f$ is a twin of CORP$_m$ with the SimCLR [5] design (Figure 2 right). Given a batch of B videos, we sample two augmented video clips for each video ($2B$ video clips in total). For each clip, the SimCLR-based method aims to solve a *contrastive pretest*, i.e. find the clip that is derived from the same video from the remaining $(2B-1)$ clips. Our model further predicts whether the found clip occurs earlier in time than the given clip using an additional objective. Generally, SimCLR framework optimizes $(2B-1)$-way classification, while our model converts it to a more challenging $(4B-2)$-way classification task.

For different fractions of mismatched pairs (not from the same video) in the training data, the two models learn different patterns, thus work on different scenarios. In the CORP$_f$ (*f*ewer clips per video) model, the primary task of $(2B-1)$ classification is more challenging than the within-class ordering. Therefore, the CORP$_f$ model pays more attention to appearance patterns (similar to SimCLR) that enable it to disambiguate same-clip entries, with lower emphasis on temporal reasoning patterns. On the other hand, CORP$_m$ (*m*ore clips per video) model focuses more on temporal relation patterns. The positive and negative pairs for the contrast task are sampled equally, and can hence learn more of temporal patterns from the order task on videos.

Our self-supervised models are validated on two popular benchmark datasets, Kinetics400 and Something-something V2. We evaluate the learned video representations by linear evaluation [5, 20] following conventional practice, i.e., training a linear classifier with features extracted by the frozen backbone. As shown in Figure 1, the two datasets are different in terms of appearance and temporal relation. Kinetics400 dataset is the scenario where appearance information is vital while Something-something is the other scenario where temporal clues are more important. On Kinetics400, our CORP$_m$ model performs worse than the contrastive based method CVRL [37], however, our CORP$_f$ model can outperform CVRL with a clear margin. On Something-something V2, our CORP$_f$ model achieves 41.7% top-1 accuracy, which is a 10% improvement over the contrastive-based method. The CORP$_m$ model achieves an even higher accuracy of 48.8%, minimizing the performance gap with supervised learning (58.4%). Our extensive ablation studies verify the effectiveness of our methods, especially that both appearance and temporal relations are learned by our method and they are both vital for video tasks.

2. Related Work

Self-supervised video representation learning. Temporal information is a natural supervision signal for self-supervised learning from video. [40] proposes an encoder-decoder LSTM to reconstruct the input frames or predict future frames. Inspired by two-stream approaches, [43] proposes to learn both motion and appearance statistics along spatio-temporal dimensions. [47], [9] and [50] use another important cue, cycle-consistency, to make full use of video correspondence. [3] and [44] studies the "speediness" of moving objects in videos as cues for video self-supervised learning. [37] applies the image contrastive learning method, SimCLR [5], to video tasks, and [19] proposes to use optical flow to co-train the contrastive learning framework. Cross-modality itself is also widely studied for video self-supervision, such as geometry [14], language [41], narrations [32], audio [26, 1], and multi-modal tasks [2].

Sorting sequences. Our work is related to a series of studies on sorting frames or video clips [34, 13, 28, 51, 25]. [34] learns to verify whether a sequence of video clips are in the correct order. [13] learns to predict the odd video subsequence from a set of correct-order sequences. These two pretext tasks are relatively easy since the most part of the sequence is correct (information is very sufficient). They

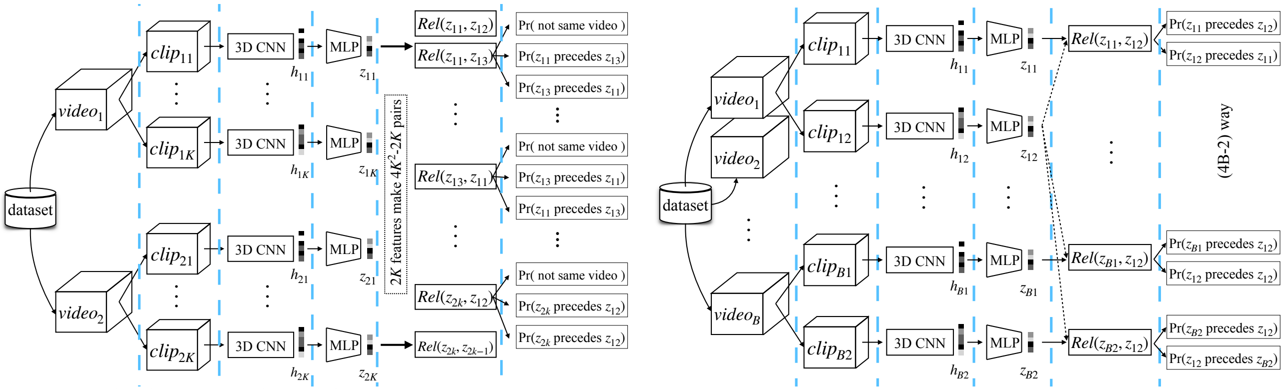

Figure 2. **Left: Overview of the CORP$_m$ model**. In this setting, we first select 2 videos and sample K augmented clips for each video. We then feed the $2K$ clips into the backbone network with a nonlinear head. The output $2K$ features make $2K(K-1)$ pairs (except pairs came from the same feature). The model learns to minimize the number of wrong predictions on pair relation classes (3-way classification).
Right: Overview of the CORP$_f$ model. In this setting, we first select B videos and sample 2 augmented clips for each video. Similarly we get $2B$ features. For each feature (video clip), the model learns to find the other feature that is sampled from the sample video, and whether this clip is earlier or later. Thus it is a $4B-2$-way classification: $2B-1$ clips and each clip has two options.

could be solved without strong temporal modeling. [28] and [51] learn to classify the correct sequence from all possible permutations. Suppose there are N video clips, there can be $N!$ orders. The classification module requires $O((N!)^2)$ parameters (Ablation studies show that a larger N makes a better representation). This limits the maximum number of video clips to be sorted. In our model, a sequence of 8 video clips is sorted. If we use their methods, 10^9 parameters are required for the classification module, while the backbone network contains only $10^7 \sim 10^8$ parameters. [25] solves this problem by using part of the $N!$ orders. This may bring unnecessary bias in the pretext task.

Self-supervised image representation learning. Some early works explore many pretext tasks for self-supervised learning, such as patch location [8], jigsaw puzzles [35], auto-encoding [24], and rotation prediction [15]. Many recent studies focus on discriminative contrastive learning [5, 20, 18, 23, 33, 36]. Most contrastive learning methods target instance discrimination, while relations between different parts in the instance are less studied.

Appearance and relation learning. [45] proposes a two-branch network for video classification: the appearance branch for spatial modeling and the relation branch for temporal modeling. [48] introduces attention mechanisms [42] to video tasks for non-local relation learning. TSN [46] is a simple and efficient baseline for video classification, but can only average the appearance information of different video stages. [29] proposes the temporal shift module (TSM) to capture temporal relationships, and greatly improve the performance of TSN. [10] also finds that temporal order matters more on SthSthV2 dataset than K400 dataset.

3. Methodology

In this section, we begin by introducing the common components of our CORP method for self-supervised contrast-and-order learning on videos. Then we present two settings CORP$_m$ and CORP$_f$ of our CORP method for learning both appearance and temporal relations. After that, we introduce a decoupling attention module that can model pairwise relations and enhance the representation ability. Finally, we compare the two models under different scenarios and provide detailed discussions for practical usage.

3.1. Basic Components

An overview of our CORP$_m$ and CORP$_f$ models (f represents sampling "few" clips in each video and m represents "more" sampling) is shown in Fig 2. Specifically, in CORP$_f$, we sample many videos (e.g., 512) but only two clips in each video to compute the loss as in CVRL [37]. In CORP$_m$, we sample more clips from each video to compute the loss. They both consist of several major modules/components including data processing, clip pairs design, loss function, etc. We begin with the common modules then introduce the unique components which are related to each design of them.

Backbone. Following [48, 49], our backbone network is based on the ResNet-50 Inflated 3D (I3D) architecture, and the down-sampling strategy is adjusted so that every stage's space-time resolution is the same as [37]. The video representation is a 2048-dimensional feature vector ($\boldsymbol{h}_{11}, \cdots \boldsymbol{h}_{2K}$ as shown in Fig. 2).

Nonlinear projection. Following the practice and design of prior related works, instead of using this representation directly for self-supervised tasks, we also add a multi-layer projection head following the backbone to obtain a new d-dimensional feature vector ($\boldsymbol{z}_{11}, \cdots \boldsymbol{z}_{2K}$ as shown in Fig. 2). The number of hidden layers is a hyper-parameter (we choose 3 following previous practice), and the dimension of the hidden layers is 2048.

3.2. CORP$_m$ Model

Video sampling. For one video, we randomly sample K video clips at a time (K is a hyper-parameter, typically $2 \sim 4$). With a batch of B videos, $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_B$, we can have $(B^2 - B)/2$ **video-level pairs**:$\{\boldsymbol{x}_i, \boldsymbol{x}_j\}_{i<j}$. Thus there are $2K$ video clips and the corresponding feature vectors in each video pair. In practice, the video-level pairs are counted within each GPU for communication efficiency.

Clip-level pairs. Using the above video sampling strategy, for each video pair, we will have $(4K^2 - 2K)$ ordered clip-level pairs. For each clip-level pair, there are three possibilities/categories:

$\mathcal{P}_1$: The two clips are sampled from different videos.

$\mathcal{P}_2$: The two clips are sampled from the same video, and the first clip precedes the second one;

$\mathcal{P}_3$: The two clips are sampled from the same video, and the second clip precedes first one.

Features for 3-way classification. For every clip-level pair, we have two projected features $\boldsymbol{z}_i, \boldsymbol{z}_j$. The objective of CORP$_m$ model is to learn the categories given two projected features. We first use multi-head attention as a baseline to represent the relation of $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$:

$$r(\boldsymbol{z}_i, \boldsymbol{z_j}) = [\langle \boldsymbol{U}_1\boldsymbol{z}_i, \boldsymbol{V}_1\boldsymbol{z}_j\rangle, \cdots, \langle \boldsymbol{U}_h\boldsymbol{z}_i, \boldsymbol{V}_h\boldsymbol{z}_j\rangle] \in \mathbb{R}^h. \tag{1}$$

Here $\boldsymbol{U}_i, \boldsymbol{V}_i \in \mathbb{R}^{\frac{d}{h}\times d}$ are model parameters and we set $h = 128$. Next we use a simple two-layer perceptron on top of it to represent the three possibilities (3-way classification):

$$\phi(\boldsymbol{z}_i, \boldsymbol{z}_j) = mlp(r(\boldsymbol{z}_i, \boldsymbol{z_j})) \in \mathbb{R}^3. \tag{2}$$

Loss function. Cross entropy loss is used for each clip-level pair and the final loss function for a video-level pair in this setting is the summation of the 3-way classification loss over all clip-level pairs' drawn from them[‡].

Data processing and augmentation. In the video sampling stage, we need to sample K clips within one video. These clips cannot be too close on temporal dimension of the video, otherwise the order prediction will be too simple. To avoid this, we propose a simple but effective rule to sample clips. Suppose the video has L frames. Every clip is T frames with a dilation of D (roughly covers DT frames in the video). We set a minimal offset:

$$\Delta = \min(\max(\frac{L - DT}{K}, 1), DT) \tag{3}$$

To obtain our samples, we first sample K clips independently, and then check whether the offset between every two clips is greater than this minimal offset. If it is not, we repeat the procedure until the condition is satisfied.

For data augmentation of clips, we follow [37], and use 1) random resized crop, 2) random horizontal flip with 0.5 probability, 3) random color jitter with 0.8 probability, 4) random gray scale with 0.2 probability and 5) Gaussian blur. The random seed for each frame in the same clip is set to the same so that the data augmentation is consistent along time.

3.3. CORP$_f$ Model

Video sampling. If sample B videos as the batch size in each iteration, and further sample two clips from each video, we obtain a total of $2B$ clips. We use these to compose the contrastive loss to learn the appearance information and order classification loss to learn the temporal information.

(4B-2)-way classification. Similar to CORP$_m$ model, we also aim to learn the relationships given two projected features $\phi(\boldsymbol{z}_i, \boldsymbol{z}_j)$. Multi-head attention is used to represent the relation of $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$. Since we have selected B videos and sample 2 augmented clips for each video, we will get $2B$ features. For each feature, the model learns to find the other feature that is sampled from the sample video, also whether this clip happens earlier or later. Thus it will be a ($4B$-2)-way classification problem.

Loss function. Our final loss function in this configuration is a ($4B$-2)-way classification. Suppose we have features $\{\boldsymbol{z}_{2i-1}, \boldsymbol{z}_{2i}\}_{i=1}^B$. Video clips of $\boldsymbol{z}_{2i-1}$ and $\boldsymbol{z}_{2i}$ are from the same video, and $\boldsymbol{z}_{2i-1}$ precedes $\boldsymbol{z}_{2i}$. We use the same idea in CORP$_m$ model to build a 2-way classification:

$$\phi(\boldsymbol{z}_i, \boldsymbol{z}_j) = mlp(r(\boldsymbol{z}_i, \boldsymbol{z_j})) \in \mathbb{R}^2. \tag{4}$$

In Equation 2, one dimension of $\phi(\boldsymbol{z}_i, \boldsymbol{z}_j)$, the 3-way classification output, represents the probability that $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ are from the same video: $\text{Prob}_{i,j}(\mathcal{P}_1)$, and the other two dimensions represent the order probability: $\text{Prob}_{i,j}(\mathcal{P}_2)$ and $\text{Prob}_{i,j}(\mathcal{P}_3)$ (with a softmax activation). In Equation 4, $\phi(\boldsymbol{z}_i, \boldsymbol{z}_j)$ is 2-way, thus it can only learn conditional probabilities of $\text{Prob}_{i,j}(\mathcal{P}_2|\neg\mathcal{P}_1)$, and $\text{Prob}_{i,j}(\mathcal{P}_3|\neg\mathcal{P}_1)$. We use the NCE form to model the probability that $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ from the same video:

$$\text{Prob}_{i,j}(\neg\mathcal{P}_1) = \frac{\exp\{\boldsymbol{z}_i^\top \boldsymbol{z}_j/\tau\}}{\sum_k \mathbf{1}_{[k\neq i]} \exp\{\boldsymbol{z}_i^\top \boldsymbol{z}_k/\tau\}} \tag{5}$$

Here $\tau = 0.1$ following [37]. The $(4B - 2)$-way cross entropy loss can be decomposed to the sum of InfoNCE loss and 2-way classification loss ($\mathcal{L}_{\text{total}} = 1/B \cdot \sum_i \mathcal{L}_i$):

$$\begin{aligned}\mathcal{L}_i &= -\log\left[\text{Prob}_{2i-1,2i}(\mathcal{P}_2)\right] - \log\left[\text{Prob}_{2i,2i-1}(\mathcal{P}_3)\right] \\ &= -\log\left[\text{Prob}_{2i-1,2i}(\mathcal{P}_2|\neg\mathcal{P}_1)\text{Prob}_{2i-1,2i}(\neg\mathcal{P}_1)\right] \\ &\quad -\log\left[\text{Prob}_{2i,2i-1}(\mathcal{P}_3|\neg\mathcal{P}_1)\text{Prob}_{2i,2i-1}(\neg\mathcal{P}_1)\right] \\ &= \mathcal{L}_{\text{NCE}} + \mathcal{L}_{\text{2-way order classification}}\end{aligned} \tag{6}$$

3.4. Decoupling Attention

In both 3-way and ($4B$-2)-way classification modules, we use attention $f(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{U}\boldsymbol{x}, \boldsymbol{V}\boldsymbol{y}\rangle$ to model the relation-

[‡] Some novel classification losses such as A-softmax [30] and ArcFace [6] can improve the performance, but we use cross entropy for simplicity.

ships between two features, which is a common practice in prior literature. The two projections $\boldsymbol{U}$ and $\boldsymbol{V}$ are typically different so that non-symmetric patterns can be modeled.

In our framework, the contrast task requires symmetric patterns: if x is similar to y, y is also similar to x. The order task requires non-symmetric patterns, more specifically, anti-symmetric patterns. If $\boldsymbol{x}$ is *earlier* than $\boldsymbol{y}$, $\boldsymbol{y}$ is *later* than $\boldsymbol{x}$. Many motion related patterns are anti-symmetric, such as left and right, push and pull. Formally, symmetric patterns can be represented as $f_s(x, y) = f_s(y, x)$ and anti-symmetric patterns can be represented as $f_a(x, y) = -f_a(y, x)$.

We can see that attention contains symmetric patterns and anti-symmetric patterns: $f(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{Ux}, \boldsymbol{Vy} \rangle = \boldsymbol{x}^\top (\boldsymbol{U}^\top \boldsymbol{V}) \boldsymbol{y} = \boldsymbol{x}^\top \frac{\boldsymbol{M} + \boldsymbol{M}^\top}{2} \boldsymbol{y} + \boldsymbol{x}^\top \frac{\boldsymbol{M} - \boldsymbol{M}^\top}{2} \boldsymbol{y}$ where $\boldsymbol{M} = \boldsymbol{U}^\top \boldsymbol{V}$. It is easy to verify that the first term is symmetric and the second term is anti-symmetric. Although multi-head attention has the representation ability of both symmetric and anti-symmetric patterns, they are mixed as a black box. In the multi-head attention formula (Equation 1), we cannot know which neurons of $r(\cdot, \cdot)$ represent similarity and which neurons represent temporal order. It is more likely that every neuron contains part information about similarity and part information about temporal modeling. During training, the gradient of similarity supervision signals and the gradient of temporal supervision signals might cancel each other. During deployment, the multi-head attention is also less interpretable due to the non-symmetry.

We solve this problem with decoupling attention. First we present a theorem about matrix decomposition:

Theorem. *Any matrix $\boldsymbol{M} \in \mathbb{R}^{d \times d}$ can be written as:*

$$\boldsymbol{M} = \sum_{i=1}^{n} \boldsymbol{g}_i \boldsymbol{g}_i^\top + \sum_{j=1}^{l} \boldsymbol{p}_j \boldsymbol{q}_j^\top - \boldsymbol{q}_j \boldsymbol{p}_j^\top, \tag{7}$$

where $n, l \leq d$, for some $\boldsymbol{g}_i, \boldsymbol{p}_j, \boldsymbol{q}_j \in \mathbb{R}^d$.

Define $\boldsymbol{G} = [\boldsymbol{g}_1, \cdots, \boldsymbol{g}_n]^T \in \mathbb{R}^{n \times d}$, and $\boldsymbol{P}, \boldsymbol{Q} \in \mathbb{R}^{l \times d}$ follows the same definition. We use $\boldsymbol{x} * \boldsymbol{y} \in \mathbb{R}^d$ to represent the element-wise product of $\boldsymbol{x}$ and $\boldsymbol{y} \in \mathbb{R}^d$, and $\text{sum}(\boldsymbol{x}) \in \mathbb{R}$ to represent element-wise summation of $\boldsymbol{x}$. With the matrix decomposition theorem, we have:

$$\begin{aligned} & \boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{y} \\ = & \sum_{i=1}^{n} (\boldsymbol{x}^\top \boldsymbol{g}_i)(\boldsymbol{y}^\top \boldsymbol{g}_i) + \sum_{j=1}^{l} (\boldsymbol{x}^\top \boldsymbol{p}_j)(\boldsymbol{y}^\top \boldsymbol{q}_j) - (\boldsymbol{x}^\top \boldsymbol{q}_j)(\boldsymbol{y}^\top \boldsymbol{p}_j) \\ = & \text{sum}(\boldsymbol{Gx} * \boldsymbol{Gy}) + \text{sum}(\boldsymbol{Px} * \boldsymbol{Qy} - \boldsymbol{Qx} * \boldsymbol{Py}) \end{aligned} \tag{8}$$

Equation 8 indicates that, if we want the model to learn an attention projection $\boldsymbol{U}$ and $\boldsymbol{V}$, the model can instead learn $\boldsymbol{G}, \boldsymbol{P}$ and $\boldsymbol{Q}$. The parameters $\boldsymbol{G}, \boldsymbol{P}$ and $\boldsymbol{Q}$ are symmetric to the inputs, since the operations on $\boldsymbol{x}$ are the same as the operations on $\boldsymbol{y}$. We introduce a "cross product"-like operation[§] to introduce anti-symmetry. The parameter $\boldsymbol{G}$ learns symmetric patterns, and the parameters $\boldsymbol{P}, \boldsymbol{Q}$ learn anti-symmetric patterns. Similarity supervision signals will only have gradient on $\boldsymbol{G}$ while temporal supervision signals will only have gradient on $\boldsymbol{P}, \boldsymbol{Q}$.

Equation 8 is the case for a single head. Multi-head attention stacks multiple such heads to get a vector and sends it to MLPs. To replace multi-head attention with decoupling attention, we can therefore remove the summation operation in Equation 8 to retain a vector, since the MLPs learn a linear combination of the heads.

Though we introduce our module with multi-head attention in earlier sections for simplicity, we use the decoupling attention for our models. It speeds up the training process (Figure 3) and improves the model interpretability.

3.5. Discussions

On the scene-centric video dataset Kinetics400 [4], a large proportion of video categories can simply be recognized by static appearance [56]. The ability of learning appearance patterns is more crucial than temporal information which is beneficial but not indispensable. In this case, the proposed CORP$_f$ model is more useful. In the motion-centric video dataset Something-something [17], most actions cannot be learned directly from the simple fusion of frame-level features. In this scenario, the appearance information of different action categories are similar, and temporal reasoning turns to be the key for video understanding and classification. The proposed CORP$_m$ model performs better in this scenario. Nevertheless, the general idea of learning both appearance and temporal representations is critical for self-supervised video representation learning tasks.

4. Experiments

We first evaluate our models on the widely-used Kinetics-400 (K400) dataset [4] in the linear evaluation, semi-supervised learning and transfer learning settings. Next, we conduct experiments on Something-Something V2 (SthSthV2) dataset [17]. Many action categories in this dataset share very similar background and object appearance and require strong time modeling. Finally, we make comprehensive ablation studies and case analysis to show the temporal learning ability of our model.

4.1. Implementation Details

Our models are trained with PyTorch from scratch. In the self-supervised pre-training stage, we use LARS [54] with the momentum of 0.9 and the weight decay of 10^{-6} as our

[§] Similar to cross product operation in $\mathbb{R}^3$. Let $\boldsymbol{x} = (x_1, x_2, 0)$, $\boldsymbol{y} = (y_1, y_2, 0)$, the cross product: $\boldsymbol{x} \times \boldsymbol{y} = (0, 0, x_1 y_2 - x_2 y_1)$.

optimizer. The mini-batch size is 64 for CORP$_m$ models and 512 for CORP$_f$ models. The learning rate is computed as batch_size$/256$ for all batch sizes. We use the linear warm-up learning rate for the first 5% epochs and the half cosine learning rate decay scheduling [22] for the remaining epochs. Synchronized batch normalization across all GPUs is used for the backbone and projection heads. On the K400 dataset, We sample 16 frames with a temporal stride of 2 (covers 32 continues frames) as a clip, while on the Sth-SthV2 dataset, the temporal stride is 1 due to the short average video length.

In the linear evaluation stage, we use SGD with the momentum of 0.9 and no weight decay as our optimizer. The batch size is 1024 and the initial learning rate is 0.16 following [5]. The half cosine learning rate decay scheduling is applied without warmup. All layers (including the running statistics in batch normalization) except the last linear layer are frozen with the pre-training backbone (i.e., not trainable). Z-score standardization is used to normalize the features before feeding it into the last linear layer. We sample 32 frames with the same temporal stride as pre-training and train 100 epochs. The data augmentation is the same as in pre-training except that color jittering and Gaussian blur are removed. Unless otherwise stated, the reported top-1 accuracies are obtained by linear evaluation.

In the semi-supervised learning stage, we use the pre-training backbone to initialize the network, and fine-tune all layers on a small subset of the data. We sample 1% and 10% videos from the train set and keep the percentages of each class. The settings are similar to the linear evaluation, except that the initial learning rate is 0.2.

In the inference stage, we uniformly sample 10 clips from the full-length videos, and use 3 crops for each clips. The final prediction is obtained from the average of softmax probabilities of the model outputs of all 30 views.

4.2. Experiments on Kinetics400

K400 dataset contains about 240k training videos and 20k validation videos in 400 video categories[*]. As discussed earlier, a large proportion of the videos in this dataset can be recognized by a single frame [56], appearance learning is more important than temporal modeling in this dataset. Thus CORP$_f$ model is preferred on this dataset for self-supervised learning on this dataset while the CORP$_m$ model is expected to have lower performance.

By default, the CORP$_m$ model uses $K = 4$, and both models are pre-trained for 800 epochs. Table 1 shows the results of linear evaluation results of our models and other state of the art methods on the K400 dataset. The result of CVRL is obtained with a batch size of 1024 while our result is obtained with a batch size of 512 [†]. On the same setting of 1024 batch size, our CORP$_f$ model achieves **66.6**% validation accuracy on Kinetics400 dataset.

Model	Network (#params)	Top-1 Acc
VTHCL[52]	R3D-50 (31.7M)	37.8 %
VINCE[16]	R-50 (23.5M)	49.1 %
SeCo[53]	R-50 (23.5M)	61.9 %
CVRL[37]	R3D-50 (31.7M)	62.9 %
CVRL$_L$[37]	R3D-50 (31.7M)	66.1 %
CORP$_m$	R3D-50 (31.7M)	59.1 %
CORP$_f$	R3D-50 (31.7M)	**66.3** %

Table 1. Linear evaluation results on the Kinetics-400 dataset.

Table 2 shows the ablation studies of batch size and training epochs on K400 dataset using CORP$_f$ models. We can find that the performance can be improved if we increase the batch size or the number of pre-training epochs.

Top-1 Acc	# epochs = 200	# epochs = 500
batch size = 256	60.9%	63.0 %
batch size = 512	64.1%	65.6%

Table 2. Ablation studies of batch size and number of pre-train epochs on Kinetics-400 dataset using the CORP$_f$ model.

Table 3 shows the results of semi-supervised learning results on K400 dataset. In the 1% label setting, results of CORP$_f$ and CVRL are very close, while In the 10% label setting, our method outperforms CVRL.

Model	1% label	10% label
CVRL	35.1%	58.1%
CORP$_f$	34.8%	58.6%

Table 3. Semi-supervised learning results on Kinetics-400.

Table 4 shows the transfer learning ability of our learned representations on two smaller video dataset UCF101 [39] and HMDB51 [27] dataset. We test on two settings: 1) linear evaluation of features extracted from the frozen backbone, 2) fine-tuning all parameters initialized by the pre-training model on the new datasets.

Model	linear evaluation		fine tuning	
	UCF101	HMDB51	UCF101	HMDB51
CVRL	89.8%	58.3%	92.9%	67.9%
CORP$_f$	90.2%	58.7%	93.5%	68.0%

Table 4. Transfer learning comparison of our method and CVRL

[*] Following [37], we use the original size dataset instead of another commonly used downsample version [11].

[†] Large batch sizes result in better performances for contrastive learning. Experiments of a batch size of 1024 requires 64 32GB-V100 GPUs. Due to budget limits, we only run one experiment with a batch size of 1024, and other results are obtained under a batch size of 512.

4.3. Experiments on Something-Something V2

Something-Something V2 (Sth-SthV2) dataset contains about 168k training videos and 24k validation videos in 174 video categories. Many action categories in this dataset shares very similar background and object appearance, thus it would be difficult to classify different action categories.

To the best of our knowledge, no previous related work reports the self-supervised learning performance on Sth-SthV2 (or V1) dataset. However, we argue that the it is necessary to validate video self-supervised models on Sth-SthV2 dataset, especially for temporal learning. To understand this, we compare the performance of our CORP$_m$ model, CORP$_f$ model and 4 other models on the K400 and Sth-SthV2 datasets:

1 **C2D**: Supervised learning of ResNet50 C2D model as described in [48]. No 3D convolution layers are used.

2 **I3D**: Supervised learning of ResNet50 I3D model as described in [48]. A baseline for action recognition.

3 **ImageNet**: Initialize the I3D model with ImageNet pre-training weights, and apply linear evaluation.

4 **SimCLR**: On K400, use the CVRL result of 200 epochs, 512 mini-batch size reported by [37]. On Sth-SthV2, we implement the model with this setting.

Models	Supervised	Kinetics	Something V2
C2D	√	71.8%[48]	45.6%
I3D	√	73.3%[48]	58.4%
ImageNet	×	53.5%[37]	13.3%
SimCLR	×	62.9%[37]	33.9%
CORP$_m$	×	56.1%	**48.8**%
CORP$_f$	×	**63.4**%	41.1%

Table 5. Top 1 accuracies of 6 models on Kinetics and Something-Something V2 dataset. Results without citations are trained by us. Non-supervised results are obtained from linear evaluation.

As shown in Table 5, the performance gap between the C2D model and the I3D model on K400 dataset is very small. However the performance gaps between self-supervised methods and the supervised methods are even bigger than the performance gap between self-supervised methods and fixed ImageNet weights. It supports our argument that temporal modeling is less important on the K400 dataset. It would remain questionable if a model can learn good spatio-temporal representations even if it has a high performance on K400. For example, a self-supervised 2D model may minimize the performance gap with the C2D model, but may not learn temporal representations. However, the performance gap between the C2D model and the I3D model is much larger on the Sth-SthV2 and the ImageNet features is almost not able to recognize actions in Sth-SthV2 since 2D models/features cannot cannot represent temporal patterns which is important in Sth-SthV2 dataset. CVRL does not work well on Sth-SthV2 dataset. However its performance can be improved by simply introducing the order task. Although CORP$_f$'s performance is better than CORP$_m$ on K400 dataset, CORP$_m$ is better on Sth-SthV2 and even better than the supervised C2D model. It is consistent with our theory that CORP$_m$ method focuses more on temporal modeling which is greatly requires on Sth-SthV2 dataset.

Task	Top-1 Acc
Order	42.4%
Contrast	29.7%
CORP$_m$	48.8%

(a) ablation on the tasks.

# clips	Top-1 Acc
2	45.4%
3	47.0%
4	48.8%

(b) number of clips

Attention	Top-1 Acc
Mixed	48.0%
Decoupling	48.8%

(c) relation modeling.

# layers	Top-1 Acc
1	48.4%
2	48.8%

(d) # perceptron layers

Epoch	SimCLR-based	CORP$_f$	CORP$_m$
100	31.1%	38.8%	46.3%
200	33.9%	41.7%	48.8%

(e) ablation on pre-training epochs.

Table 6. Ablation Studies on Something-Something V2 dataset.

Next we show ablation studies on Sth-SthV2 dataset.

The roles of contrast and order learning: Our models learn two tasks 1) contrast learning to distinguish video clips' source, and 2) order learning to distinguish video clips' order. We use the CORP$_m$ model to study the roles of the two tasks as shown in Table 5a. The CORP$_m$ model learns by a 3-way classification. An order-only task (first line) is a 2-way classification that only learns the order of clips from the same video. A contrast-only task (first line) is a 2-way classification that only learns whether two video clips come from the same videos. We can find both order-only and contrast-only tasks are worse than the CORP$_m$ model, and the order-only task is better than the contrast-only task since temporal patterns are more important on Sth-SthV2 dataset. The performance of contrast-only task is similar but lower than the SimCLR-based method in Table **??**tab:ablation]5e since infoNCE loss is more effective than simple classification.

Number of clips sampled in one video: In Table 5b, we study the hyper-parameters K. In our default setting, we choose K=4, i.e., sample 4 video clips in one video. Note that a smaller K indicates smaller computation. We train 200 epochs for $K = 4$, 267 epochs for $K = 3$ and 400 epochs for $K = 400$ to have the same computation cost. A

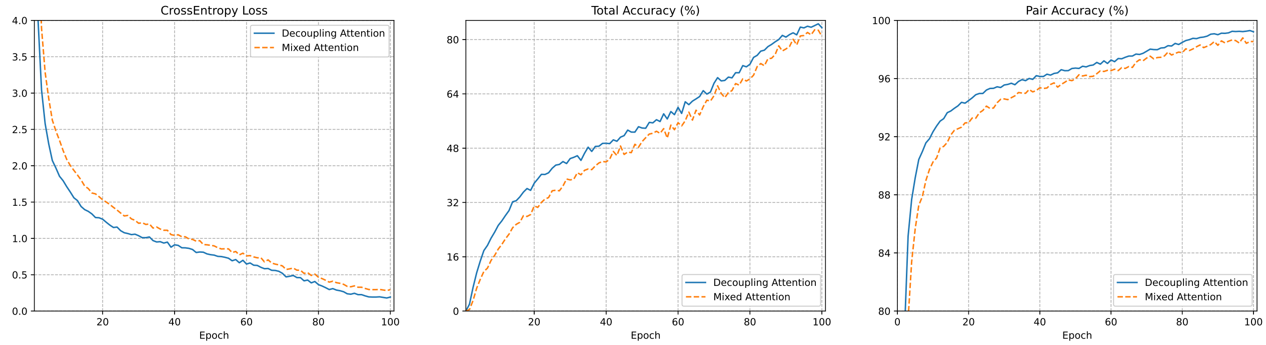

Figure 3. Training curves for CORP$_m$ model on Something Something-V2 dataset. Left figure: CrossEntropy Loss (Lower is Better). Middle figure: Total Accuracy (Higher is Better). Right figure: Pair Accuracy (Higher is Better).

larger K still makes better performance since the number of temporal positive-negative pairs is $O(K^2)$.

Decoupling attention *vs*. Mixed attention: Given two video clips, we need to model their relations given their feature vectors. As discussed in Section 3, a simple idea is to use the mixed attention (multi-head attention). We propose decoupling attention that decouples symmetric and anti-symmetric patterns in the mixed attention. Figure 3 shows the performance during pre-training. Total Accuracy is the ratio of completely correct video pairs ("completely" means that all clip pairs in the video pair are classified correctly) to all video pairs. Table 5c shows the linear evaluation performance of the two kinds of attention. Decoupling attention is better than the original mixed attention in both pre-training and downstream tasks.

Number of layers in the perceptron: The CORP$_m$ model use a small perceptron given the pair relation to learn the 3-way classification (Equation 2). Table 5d shows that a deeper perceptron still helps even the nonlinear projection head is deep.

Number of pre-training epochs: Table 5e shows the results of training 100 and 200 epochs for SimCLR-based, CORP$_f$ and CORP$_m$ models. The performance comparison is consistent in different number of pre-training epochs.

Classification results analysis: We analyse the category accuracies for SimCLR-based, CORP$_f$ and CORP$_m$ models. We show 1) some categories that all three models have good/similar performances and 2) some categories that CORP$_f$ and CORP$_m$ models have very different performances with the SimCLR-based method.

As shown in Table 7, SimCLR can achieve similar performance in the above 5 categories with our models. From the category name, these categories do not require strong temporal reasoning. A single frame can help recognize "holding something" or "plugging something into something". A linear fusion of several frame features is enough for "tearing something into 2 pieces" since it is not likely to reverse the action and reconstruct 2 pieces into something. Our methods can have limit advantages over SimCLR over the first four categories. However, the next 4 categories cannot be correctly classified without temporal learning since the moving directions along time dimension is the required pattern. If we reverse one video of "Moving something up" in the time dimension, and it turns out to be "Moving something down". The accuracies of SimCLR model is almost half of our models. It shows the strong temporal learning ability of our model, as well as the necessity of evaluation on Sth-SthV2 dataset for video self-supervised learning.

Category	SimCLR	CORP$_f$	CORP$_m$
Tear something into 2 pieces	77.0	84.3	87.4
Approach sth with camera	61.2	88.8	93.1
Show something behind sth	59.6	62.8	65.1
Plug something into sth	58.9	65.4	70.8
Hold something	29.9	26.9	30.5
Move sth and sth closer	41.4	74.1	78.3
Move sth and sth away	30.0	73.7	80.2
Move something up	34.2	51.1	58.9
Move something down	23.5	67.2	67.2

Table 7. Category accuracies of three models. "sth" is short for "something". Three models have similar performance on the above 5 categories, but very different performances on the next 4 categories.

A recent work [12] shows fine-tuning results of Sth-SthV2 dataset. The fine-tuning results of R30-50 models of four popular contrastive framework (SimCLR, BYOL [18], etc) ranges from 52.8% to 55.8%. The CORP$_m$ model achieves 61% fine-tuning accuracy on Sth-SthV2 validation.

5. Conclusion

We introduce a contrast-and-order framework for self-supervised video representations learning on spatial and temporal dimensions. Two implementations CORP$_f$ and CORP$_m$ are proposed for different scenarios whose efficacy is verified on Kinetics400 and Something-something V2 dataset. Our CORP model consistently outperform existing competitors by a significant margin.

References

[1] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. *Advances in Neural Information Processing Systems*, 33, 2020.

[2] Yuki M Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. *Advances in Neural Information Processing Systems*, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9922–9931, 2020.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 2020.

[6] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[7] 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.

[8] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[9] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[10] Debidatta Dwibedi, Pierre Sermanet, and Jonathan Tompson. Temporal reasoning in videos using convolutional gated recurrent units. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1111–1116, 2018.

[11] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. Pyslowfast. https://github.com/facebookresearch/SlowFast/blob/master/slowfast/datasets/DATASET.md, 2020.

[12] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3299–3309, 2021.

[13] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3636–3645, 2017.

[14] Chuang Gan, Boqing Gong, Kun Liu, Hao Su, and Leonidas J Guibas. Geometry guided convolutional neural networks for self-supervised video representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5589–5597, 2018.

[15] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

[16] 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.

[17] 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 *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[18] 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, et al. Bootstrap your own latent: A new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 2020.

[19] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *Advances in Neural Information Processing Systems*, 2020.

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[22] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 558–567, 2019.

[23] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR, 2020.

[24] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[25] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8545–8552, 2019.

[26] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. *Advances in Neural Information Processing Systems*, 2018.

[27] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video

database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE, 2011.

[28] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[29] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019.

[30] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[31] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[32] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020.

[33] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

[34] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016.

[35] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*. Springer, 2016.

[36] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[37] 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.

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[39] 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.

[40] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852. PMLR, 2015.

[41] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[43] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4006–4015, 2019.

[44] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In *European Conference on Computer Vision*, pages 504–521. Springer, 2020.

[45] Limin Wang, Wei Li, Wen Li, and Luc Van Gool. Appearance-and-relation networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1430–1439, 2018.

[46] 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 *European conference on computer vision*, pages 20–36. Springer, 2016.

[47] Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1308–1317, 2019.

[48] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[49] X. Wang and A. Gupta. Videos as space-time region graphs. In *European Conference on Computer Vision (ECCV)*, 2018.

[50] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019.

[51] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10334–10343, 2019.

[52] Ceyuan Yang, Yinghao Xu, Bo Dai, and Bolei Zhou. Video representation learning with visual tempo consistency. *arXiv preprint arXiv:2006.15489*, 2020.

[53] Ting Yao, Yiheng Zhang, Zhaofan Qiu, Yingwei Pan, and Tao Mei. Seco: Exploring sequence supervision for unsupervised representation learning. *arXiv preprint arXiv:2008.00975*, 2020.

[54] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

[55] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[56] Yi Zhu, Xinyu Li, Chunhui Liu, Mohammadreza Zolfaghari, Yuanjun Xiong, Chongruo Wu, Zhi Zhang, Joseph Tighe, R Manmatha, and Mu Li. A comprehensive study of deep video action recognition. *arXiv preprint arXiv:2012.06567*, 2020.