Spatiotemporal Contrastive Video Representation Learning

Rui Qian[* 1,2,3]   Tianjian Meng[* 1]   Boqing Gong[1]   Ming-Hsuan Yang[1]
Huisheng Wang[1]   Serge Belongie[1,2,3]   Yin Cui[1]

[1]Google Research   [2]Cornell University   [3]Cornell Tech

Abstract

We present a self-supervised Contrastive Video Representation Learning (CVRL) method to learn spatiotemporal visual representations from unlabeled videos. Our representations are learned using a contrastive loss, where two augmented clips from the same short video are pulled together in the embedding space, while clips from different videos are pushed away. We study what makes for good data augmentations for video self-supervised learning and find that both spatial and temporal information are crucial. We carefully design data augmentations involving spatial and temporal cues. Concretely, we propose a temporally consistent spatial augmentation method to impose strong spatial augmentations on each frame of the video while maintaining the temporal consistency across frames. We also propose a sampling-based temporal augmentation method to avoid overly enforcing invariance on clips that are distant in time. On Kinetics-600, a linear classifier trained on the representations learned by CVRL achieves 70.4% top-1 accuracy with a 3D-ResNet-50 (R3D-50) backbone, outperforming ImageNet supervised pre-training by 15.7% and SimCLR unsupervised pre-training by 18.8% using the same inflated R3D-50. The performance of CVRL can be further improved to 72.9% with a larger R3D-152 (2× filters) backbone, significantly closing the gap between unsupervised and supervised video representation learning. Our code and models will be available at https://github.com/tensorflow/models/tree/master/official/.

1. Introduction

Representation learning is of crucial importance in computer vision tasks, and a number of highly promising recent developments in this area have carried over successfully from the static image domain to the video domain. Classic hand-crafted local invariant features (*e.g.*, SIFT [44])
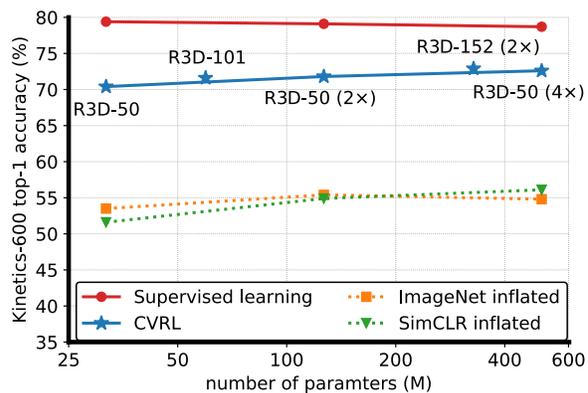

Figure 1. **Kinetics-600 top-1 linear classification accuracy** of different spatiotemporal representations. CVRL outperforms ImageNet supervised [33] and SimCLR unsupervised [10] pre-training using the same 3D inflated ResNets, closing the gap between unsupervised and supervised video representation learning.

for images have their counterparts (*e.g.*, 3D SIFT [55]) in videos, where the temporal dimension of videos gives rise to key differences between them. Similarly, state-of-the-art neural networks for video understanding [63, 9, 31, 71, 17, 16] often extend 2D convolutional neural networks [33, 36] for images along the temporal dimension. More recently, unsupervised or self-supervised learning of representations from unlabeled visual data [32, 10, 26, 7] has gained momentum in the literature partially thanks to its ability to model the abundantly available unlabeled data.

However, self-supervised learning gravitates to different dimensions in videos and images, respectively. It is natural to engineer self-supervised learning signals along the temporal dimension in videos. Examples abound, including models for predicting the future [58, 43, 28], changing temporal sampling rates [73], sorting video frames or clips [42, 39, 72] and combining a few tasks [5]. Meanwhile, in the domain of static images, some recent work [32, 10, 26, 7] that exploits spatial self-supervision has reported unprecedented performance on image representation learning.

The long-standing pursuit after temporal cues for self-

---

supervised video representation learning has left self-supervision signals in the spatial subspace under-exploited for videos. To promote the spatial self-supervision signals in videos, we build a Contrastive Video Representation Learning (CVRL) framework to learn spatiotemporal representations from unlabeled videos. Figure 2 illustrates our framework, which contrasts the similarity between two positive video clips against those of negative pairs using the InfoNCE contrastive loss [48]. Since there is no label in self-supervised learning, we construct positive pairs as two augmented video clips sampled from the same input video.

We carefully design data augmentations to involve both spatial and temporal cues for CVRL. Simply applying spatial augmentation independently to video frames actually hurts the learning because it breaks the natural motion along the time dimension. Instead, we propose a temporally consistent spatial augmentation method by fixing the randomness across frames. It is simple and yet vital as demonstrated in our experiments. For temporal augmentation, we take visual content into account by a sampling strategy tailored for the CVRL framework. On the one hand, a pair of positive clips that are temporally distant may contain very different visual content, leading to a low similarity that could be indistinguishable from those of the negative pairs. On the other hand, completely discarding the clips that are far in time reduces the temporal augmentation effect. To this end, we propose a sampling strategy to ensure the time difference between two positive clips follows a monotonically decreasing distribution. Effectively, CVRL mainly learns from positive pairs of temporally close clips and secondarily sees some temporally distant clips during training. The efficacy of the proposed spatial and temporal augmentation methods is verified by extensive ablation studies.

We primarily evaluate the learned video representations on both Kinetics-400 [38] and Kinetics-600 [8] by training a linear classifier following [10, 32] on top of frozen backbones. We also study semi-supervised learning, downstream action classification and detection to further assess CVRL. We next summarize our main findings.

Mixing spatial and temporal cues boosts the performance. Relying on spatial or temporal augmentation only yields relatively low performance, as shown in Table 9. In contrast, we achieve an improvement of 22.9% top-1 accuracy by combining both augmentations in the manner we proposed above, *i.e.*, temporally consistent spatial augmentation and the temporal sampling strategy.

Our representations outperform prior arts. The linear evaluation of CVRL achieves more than 15% gain over competing baselines, as shown in Figure 1 and Table 2. On Kinetics-400, CVRL achieves 12.6% improvement over ImageNet pre-training, which were shown competitive in previous work [73, 24]. For semi-supervised learning (Table 3), CVRL surpasses all other baselines especially when

there is only 1% labeled data, indicating the advantage of our self-learned feature is more profound with limited labels. For downstream action classification on UCF-101 [57] and HMDB-51 [41], CVRL has obvious advantages over other methods based on the vision modality and is competitive with state-of-the-art multimodal methods (Table 4).

Our CVRL framework benefits from larger datasets and networks. We study the effect of more training data in CVRL. We design an evaluation protocol by first pre-training models on different amounts of data with same iterations, and then comparing the performance on the same validation set. As shown in Figure 4, a clear improvement is observed by using 50% more data, demonstrating the potential of CVRL to scale to larger unlabeled datasets. We also conduct experiments with wider & deeper networks and observe consistent improvements (Table 2), demonstrating that CVRL is more effective with larger networks.

2. Related Work

Self-supervised video representation learning. It is natural to exploit the temporal dimension in self-supervised video representation learning. Some early work predicts the future on top of frame-wise representations [58]. More recent work learns from raw videos by predicting motion and appearance statistics [66], speed [6, 67] and encodings [43, 28, 29]. Aside from predicting the future, other common approaches include sorting frames or video clips [42, 72, 39, 18] along the temporal dimension and learning from proxy tasks like rotation [37]. Yang *et al.* [73] learn by maintaining consistent representations of different sampling rates. Furthermore, videos can often supply multimodal signals for cross-modality self-supervision, such as geometric cues [19], speech or language [60, 59, 45], audio [40, 3, 51, 4], optical flow [30] or combinations of multiple modalities [2] and tasks [52].

Self-supervised image representation learning. Some early work learns visual representations from unlabeled images via manually specified pretext tasks, for instance, the auto-encoding methods [50, 77, 78] that leverage contexts, channels, or colors. Other pretext tasks include but are not limited to relative patch location [14], jigsaw puzzles [47], and image rotations [22]. Interestingly, most of the pretext tasks can be integrated into a contrastive learning framework [32, 10, 76, 26, 48, 35, 61], which maintains relative consistency between the representations of an image and its augmented view. The augmentation could encompass various pretext tasks. Tian *et al.* [62] study what makes a good view in this framework. Clustering can also provide an effective addition to the framework [7]. It is worth noting that the recent wave of contrastive learning shares a similar loss objective as instance discrimination [70].
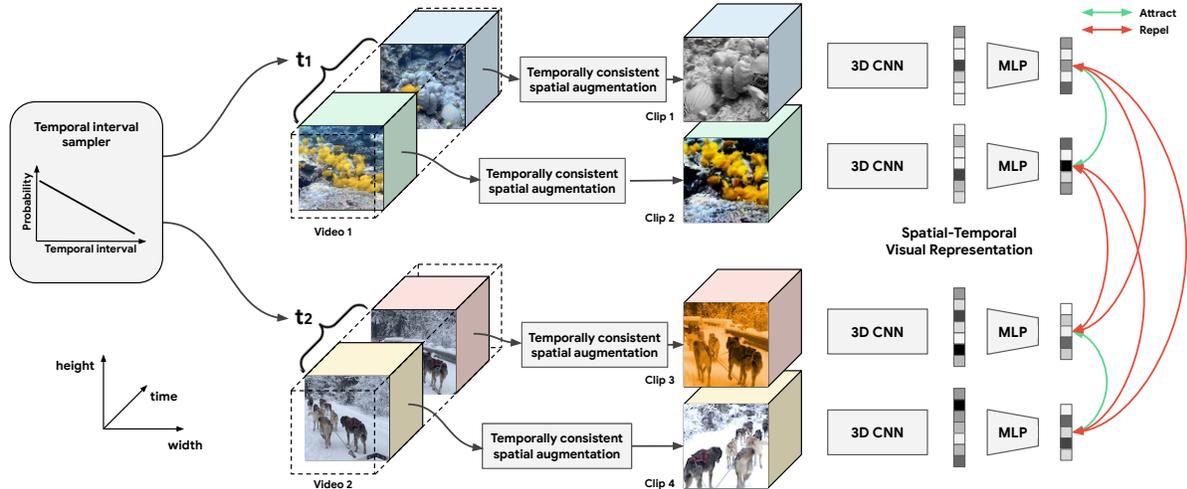
Figure 2. **Overview of the proposed spatiotemporal Contrastive Video Representation Learning (CVRL) framework.** From a raw video, we first sample a temporal interval from a monotonically decreasing distribution. The temporal interval represents the number of frames between the start points of two clips, and we sample two clips from a video according to this interval. Afterwards we apply a temporally consistent spatial augmentation to each of the clips and feed them into a 3D backbone with an MLP head. The contrastive loss is used to train the network to attract the clips from the same video and repel the clips from different videos in the embedding space.

Videos as supervision for images and beyond. Video can help supervise the learning of image representations [68, 49, 65, 24, 53], correspondences [69, 15], and robotic behaviors [56] thanks to its rich content about different views of objects and its motion and tracking cues. On the other hand, Girdhar *et al.* [23] propose to learn video representations by distillation from image representations.

3. Methodology

3.1. Video Representation Learning Framework

We build our self-supervised contrastive video representation learning framework as illustrated in Figure 2. The core of this framework is an InfoNCE contrastive loss [48] applied on features extracted from augmented videos. Suppose we sample N raw videos and augment them, resulting in $2N$ clips (the augmentation module is described in Section 3.3). Denote z_i, z_i' as the encoded representations of the two augmented clips of the i-th input video. The InfoNCE contrastive loss is defined as $\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_i$ and

$$\mathcal{L}_i = -\log \frac{\exp(\mathrm{sim}(z_i, z_i')/\tau)}{\sum_{k=1}^{2N}\mathbf{1}_{[k\neq i]}\exp(\mathrm{sim}(z_i, z_k)/\tau)}, \quad (1)$$

where $\mathrm{sim}(u, v) = u^\top v/\|u\|_2\|v\|_2$ is the inner product between two ℓ_2 normalized vectors, $\mathbf{1}_{[\cdot]}$ is an indicator excluding from the denominator the self-similarity of the encoded video z_i, and $\tau > 0$ is a temperature parameter. The loss allows the positive pair (z_i, z_i') to attract mutually while they repel the other items in the mini-batch.

We discuss other components of the framework as follows: (1) an encoder network maps an input video clip to its representation z, (2) spatiotemporal augmentations to construct positive pairs (z_i, z_i') and the properties they induce, and (3) methods to evaluate the learned representations.

3.2. Video Encoder

We encode a video sequence using 3D-ResNets [33, 31] as backbones. We expand the original 2D convolution kernels to 3D to capture spatiotemporal information in videos. The design of our 3D-ResNets mainly follows the "slow" pathway of the SlowFast network [17] with two minor modifications: (1) the temporal stride of 2 in the data layer, and (2) the temporal kernel size of 5 and stride of 2 in the first convolution layer. We also take as input a higher temporal resolution. Table 1 and Section 4.1 provide more details of the network. The video representation is a 2048-dimensional feature vector. As suggested by SimCLR [10], we add a multi-layer projection head onto the backbone to obtain the encoded 128-dimensional feature vector z used in Equation 1. During evaluation, we discard the MLP and use the 2048-dimensional representation directly from the backbone to make the video encoder compatible with other supervised learning methods. We also experiment with $2\times$ and $4\times$ backbones, which multiply the number of filters in the network, including the backbone's output feature dimension and all layers in MLP, by $2\times$ and $4\times$ accordingly.

3.3. Data Augmentation

The flexibility of CVRL allows us to study a variety of desired properties, which are incorporated in the form of data augmentations. We focus on the augmentations in both temporal and spatial dimensions.

Stage	Network	Output size $T \times S^2$
raw clip	-	32×224^2
data	stride **2**, 1^2	16×224^2
conv_1	$\begin{array}{c}\mathbf{5} \times 7^2, 64 \\ \text{stride } \mathbf{2}, 2^2\end{array}$	8×112^2
pool_1	$\begin{array}{c}1 \times 3^2 \text{ max} \\ \text{stride } 1, 2^2\end{array}$	8×56^2
conv_2	$\begin{bmatrix} 1 \times 1^2, 64 \\ 1 \times 3^2, 64 \\ 1 \times 1^2, 256 \end{bmatrix} \times 3$	8×56^2
conv_3	$\begin{bmatrix} 1 \times 1^2, 128 \\ 1 \times 3^2, 128 \\ 1 \times 1^2, 512 \end{bmatrix} \times 4$	8×28^2
conv_4	$\begin{bmatrix} 3 \times 1^2, 256 \\ 1 \times 3^2, 256 \\ 1 \times 1^2, 1024 \end{bmatrix} \times 6$	8×14^2
conv_5	$\begin{bmatrix} 3 \times 1^2, 512 \\ 1 \times 3^2, 512 \\ 1 \times 1^2, 2048 \end{bmatrix} \times 3$	8×7^2
global average pooling		1×1^2

Table 1. **Our video encoder: a 3D-ResNet-50 (R3D-50).** The input video has 16 frames (stride 2) in self-supervised pre-training and 32 frames (stride 2) in linear evaluation, semi-supervised learning, supervised learning and downstream tasks.

Temporal Augmentation: a sampling perspective. It is straightforward to take two clips from an input video as a positive pair, but how to sample the two clips matters. Previous work provides temporal augmentation techniques like sorting video frames or clips [42, 39, 72], altering playback rates [75, 67], *etc*. However, directly incorporating them into CVRL would result in learning temporally invariant features, which opposes the temporally evolving nature of videos. We instead account for the temporal changes using a sampling strategy. The main motivation is that two clips from the same video would be more distinct when their temporal interval is larger. If we sample temporally distant clips with smaller probabilities, the contrastive loss (Equation 1) would focus more on the temporally close clips, pulling their features closer and imposing less penalty over the clips that are far away in time. Given an input video of length T, our sampling strategy takes two steps. We first draw a time interval t from a distribution $P(t)$ over $[0, T]$. We then uniformly sample a clip from $[0, T-t]$, followed by the second clip which is delayed by t after the first. More details on the sampling procedure can be found in Appendix A. We experiment with monotonically increasing, decreasing, and uniform distributions, as illustrated in Figure 3. We find that decreasing distributions (a-c) generally perform better than the uniform (d) or increasing ones (e-f), aligning well with our motivation above of assigning lower sampling probability on larger temporal intervals.

(a) $P(t) \propto -t + c$ (63.8% acc.)
(b) $P(t) \propto -t^{0.5} + c$ (63.1% acc.)
(c) $P(t) \propto -t^2 + c$ (62.9% acc.)
(d) $P(t) \propto c$ (62.7% acc.)
(e) $P(t) \propto t + c$ (62.4% acc.)
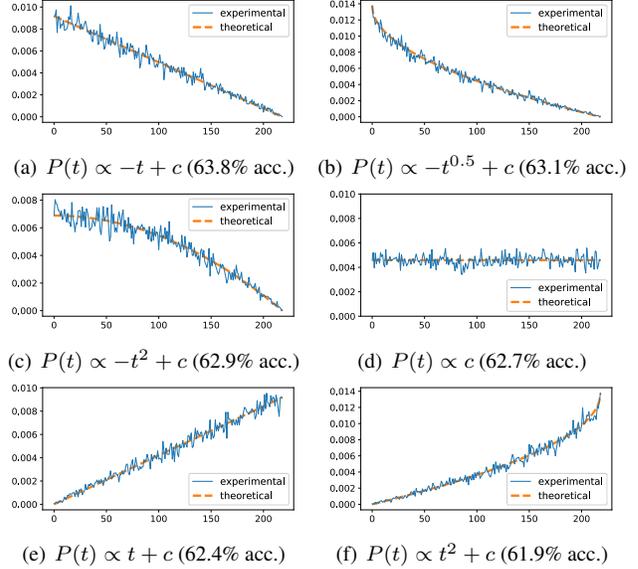(f) $P(t) \propto t^2 + c$ (61.9% acc.)

Figure 3. **Performance of different sampling distributions.** The x-axis is the temporal interval t between two clips in a video, and the y-axis is the sampling probability $P(t)$. We report linear evaluation accuracy upon 200 epochs of pre-training on Kinetics-400.

Spatial Augmentation: a temporally consistent design. Spatial augmentation is widely used in both supervised learning and unsupervised learning for images. Although the question of how to apply strong spatial augmentations to videos remains open, a natural strategy is to utilize existing image-based spatial augmentation methods to the video frames one by one. However, this method could break the motion cues across frames. Spatial augmentation methods often contain some randomness such as random cropping, color jittering and blurring as important ways to strengthen their effectiveness. In videos, however, such randomness between consecutive frames, could negatively affect the representation learning along the temporal dimension. Therefore, we design a simple yet effective approach to address this issue, by making the spatial augmentations consistent along the temporal dimension. With fixed randomness across frames, the 3D video encoder is able to better utilize spatiotemporal cues. This approach is validated by experimental results in Table 9. Algorithm 1 demonstrates the detailed procedure of our temporally consistent spatial augmentations, where the hyper-parameters are only generated once for each video and applied to all frames. An illustration can be found in Appendix C.2.

3.4. Evaluation

As a common practice in self-supervised representation learning [10, 32], we mainly evaluate the learned video representations by fixing the weights in the pre-trained video encoder and training a linear classifier on top of it. We also

Algorithm 1: Temporally consistent spatial augmentation

Input: Video clip $V = \{f_1, f_2, \cdots, f_M\}$ with M frames
Crop: Randomly crop a spatial region with size ratio **S** in range of $[0.3, 1]$ and aspect ratio **A** in $[0.5, 2]$
Resize: Resize the cropped region to size of 224×224
Flip: Draw a flag $\mathbf{F}_f$ from $\{0, 1\}$ with 50% on 1
Jitter: Draw a flag $\mathbf{F}_j$ from $\{0, 1\}$ with 80% on 1
Grey: Draw a flag $\mathbf{F}_g$ from $\{0, 1\}$ with 20% on 1
for $k \in \{1, \ldots, M\}$ **do**
$\quad f'_k = \text{Resize}\big(\text{Crop}(f_k, size = \mathbf{S}, aspect = \mathbf{A})\big)$
$\quad f'_k = \text{Flip}(f'_k)$ if $\mathbf{F}_f = 1$
$\quad f'_k = \text{Color\_jitter}(f'_k)$ if $\mathbf{F}_j = 1$
$\quad f'_k = \text{Greyscale}(f'_k)$ if $\mathbf{F}_g = 1$
$\quad f'_k = \text{Gaussian\_blur}(f'_k)$
end for
Output: Augmented video clip $V' = \{f'_1, f'_2, \cdots, f'_M\}$

assess the learned representations by fine-tuning the entire video encoder network in a semi-supervised learning setting as well as in downstream action classification and detection tasks. More details to come in Section 4.

4. Experiments

We mainly conduct experiments on the Kinetics-400 (K400) [38] and Kinetics-600 [8] (K600) datasets. K400 consists of about 240k training videos and 20k validation videos belonging to 400 action classes. K600 is a super-set of K400 by revising ambiguous classes and adding 200 more classes, containing about 360k training and 28k validation videos from 600 classes. We note that K400 has been extensively used in the literature and hope our additional results on K600 would further demonstrate the effectiveness of CVRL and offer a reference to the field. The videos in Kinetics have a duration of around 10 seconds, with 25 frames per second (*i.e.*, around 250 frames per video). We adopt the standard protocol [10, 32] of self-supervised pre-training and linear evaluation as the primary metric for evaluating the learned representations. We also evaluate the learned representations via semi-supervised learning and downstream tasks.

4.1. Implementation Details

We use SGD as our optimizer with the momentum of 0.9. All models are trained with the mini-batch size of 1024 except for downstream tasks. We linearly warm-up the learning rate in the first 5 epochs [25] followed by the scheduling strategy of half-period cosine learning rate decay [34]. We apply the proposed temporal and spatial augmentations for the self-supervised pre-training. For other tasks, we only use standard data augmentations of cropping, resizing, and flipping. During testing, we densely sample 10 clips from each video and apply a 3-crop evaluation following [17].

Self-supervised pre-training. We sample two 16-frame clips with the temporal stride of 2 from each video for the self-supervised pre-training of video representations. The duration of a clip is 1.28 seconds out of around 10 seconds of a video. We use synchronized batch normalization to avoid information leakage or overfitting [10]. The temperature τ is set to 0.1 in the InfoNCE loss for all experiments. The initial learning rate is set to 0.32.

Linear evaluation. We evaluate video representations using a linear classifier by fixing all the weights in the backbone. During training, we sample a 32-frame clip with the temporal stride of 2 from each video to train the linear classifier for 100 epochs with an initial learning rate of 32. We ℓ_2 normalize the feature before feeding it to the classifier.

Semi-supervised learning. We conduct semi-supervised learning, namely, by fine-tuning the pre-trained network on small subsets of Kinetics. We sample 1% and 10% videos from each class in the training set, forming two balanced subsets, respectively. The evaluation set remains the same. We use pre-trained backbones to initialize network parameters and fine-tune all layers using an initial learning rate of 0.2 without warm-up. We train the model for 100 epochs on the 1% subset and 50 epochs on the 10% subset.

Downstream action classification. On UCF-101 [57] and HMDB-51 [41], we use the pre-trained backbone on Kinetics to initialize the network parameters. We report results for both fine-tuning (*i.e.*, fine-tune all layers) and linear evaluation (*i.e.*, train a linear classifier by fixing all back-cbone weights). All the models are trained with a mini-batch size of 128 for 50 epochs. We use an initial learning rate of 0.16 for fine-tuning on both datasets, 0.8 for linear evaluation on UCF-101 and 0.2 for HMDB-51.

Downstream action detection. For action detection, we work on AVA [27] containing 211k training and 57k validation videos. AVA provides spatiotemporal labels of each action in long videos of 15 to 30 minutes. Following [17, 73], we adopt a Faster-RCNN [54] baseline with modifications to enable it to process videos. We use pre-trained backbones on Kinetics-400 to initialize the detector, and train with the standard $1\times$ schedule (12 epochs, decay learning rate by $10\times$ at 8-th and 11-th epoch). We use an initial learning rate of 0.2 with 32 videos per batch.

Supervised learning. To understand where CVRL stands, we also report supervised learning results. The setting for supervised learning is the same as linear evaluation except that we train the entire encoder network from scratch for 200 epochs without feature normalization. We use an initial learning rate of 0.8 and a dropout rate of 0.5 following [17].

4.2. Experimental Results

Comparison baselines. We compare our CVRL method with two baselines: (1) **ImageNet inflated** [33]: inflating the 2D ResNets pre-trained on ImageNet to our 3D

Method	Backbone (#params)	Pre-train data (duration)	Mod.	Linear eval.	Top-1 Acc. (%)
VTHCL [73]	R3D-50 (31.7M)	K400 (28d)	V	K400	37.8
SimCLR inflated	R3D-50 (31.7M)	K400 (28d)	V	K400	46.8
VINCE [24]	R-50 (23.5M)	K400 (28d)	V	K400	49.1
ImageNet inflated	R3D-50 (31.7M)	ImageNet (N/A)	V	K400	53.5
SeCo [74]	R-50 (23.5M)	K400 (28d)	V	K400	61.9
CVRL	R3D-50 (31.7M)	K400 (28d)	V	K400	**66.1**
CVRL	R3D-101 (59.7M)	K400 (28d)	V	K400	**67.6**
CVRL	R3D-152 (2×) (328.0M)	K600 (44d)	V	K400	**71.6**
Supervised (K400)	R3D-50 (31.7M)	N/A	V	N/A	76.0
SimCLR inflated	R3D-50 (31.7M)	K600 (44d)	V	K600	51.6
ImageNet inflated	R3D-50 (31.7M)	ImageNet (N/A)	V	K600	54.7
MMV-VA [2]	S3D-G (9.1M)	AS + HT (16y)	VA	K600	59.8
MMV [2]	TSM-50×2 (93.9M)	AS + HT (16y)	VAT	K600	70.5
CVRL	R3D-50 (31.7M)	K600 (44d)	V	K600	**70.4**
CVRL	R3D-101 (59.7M)	K600 (44d)	V	K600	**71.6**
CVRL	R3D-152 (2×) (328.0M)	K600 (44d)	V	K600	**72.9**
Supervised (K600)	R3D-50 (31.7M)	N/A	V	N/A	79.4

Table 2. **Linear evaluation results.** CVRL shows superior performance compared to state-of-the-art methods and baselines, significantly closes the gap with supervised learning. R-50 in the network column represents the standard 2D ResNet-50.

ResNets by duplicating it along the temporal dimension, and (2) **SimCLR inflated** [10]: inflating the 2D ResNets pre-trained with SimCLR on the frame images of Kinetics [1]. SimCLR inflated serves as an important frame-based baseline for our method by directly applying the state-of-the-art image self-supervised learning algorithm to Kinetics frames, where no temporal information is learned. In addition, we present the results of supervised learning as an upper bound of our method.

Notations. We aim at providing an extensive comparison with prior work, but video self-supervised learning methods could be diverse in pre-training datasets and input modalities. For pre-training datasets, we use K400 in short for Kinetics-400 [38], K600 for Kinetics-600 [8], HT for HowTo100M [46], AS for AudioSet [20], IG65M for Instagram65M [21], and YT8M for YouTube8M [1]. We also calculate the total length of the videos in one dataset to indicate its scale, namely duration in the table, by using **y**ears or **d**ays. Following [2], we divide modalities into four types: **V**ision, **F**low, **A**udio and **T**ext.

Linear evaluation. Linear evaluation is the most straightforward way to quantify the quality of the learned representation. As shown in Table 2, while some previous state-of-the-art methods [73, 24] are worse than ImageNet inflated, our CVRL outperforms the ImageNet inflated by 12.6% in top-1 accuracy on K400. Compared with the frame-based SimCLR inflated encoder, CVRL has 19.7% improvement, demonstrating the advantage of the learned

Method	Backbone	K400 Top-1 Acc. (Δ vs. Sup.)	
		1% label	10% label
Supervised	R3D-50	3.2	39.6
SimCLR infla.	R3D-50	11.8 (8.6↑)	46.1 (6.5↑)
ImageNet infla.	R3D-50	16.0 (12.8↑)	49.1 (9.5↑)
CVRL	R3D-50	**35.1 (31.9↑)**	**58.1 (18.5↑)**

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

spatiotemporal representation over spatial only ones. Finally, compared with the supervised upper bound, CVRL greatly closes the gap between self-supervised and supervised learning. We also compare CVRL with the very recent state-of-the-art multimodal video self-supervised learning method MMV [2] on K600. CVRL achieves performance that is on par with MMV (70.4% vs. 70.5%), with 133× less pre-training data (44 days vs. 16 years), 3× fewer parameters (31.7M vs. 93.9M) and only a single vision modality (V vs. VAT). With a deeper R3D-101, CVRL is able to show better performance (71.6% vs. 70.5%) with only 60% parameters (59.7M vs. 93.9M). Pre-training and linear evaluation curves can be found in Appendix C.1.

Semi-supervised learning. For semi-supervised learning on K400, as presented in Table 3, CVRL surpasses all other baselines across different architectures and label fractions, especially when there is only 1% labeled data for fine-tuning, indicating that the advantage of our self-supervised CVRL is more profound when the labeled data is limited. Results on K600 can be found in Appendix B.1.

Downstream action classification. Pre-training the network encoder on a large dataset and fine-tuning all layers or conducting linear evaluation on UCF-101 [57] and HMDB-51 [41] is the most common evaluation protocol in the video

---

[1]We find a SimCLR model pre-trained on Kinetics frames slightly outperforms the same model pre-trained on ImageNet released by [10]. This is probably due to the domain difference between ImageNet and Kinetics.

self-supervised learning literature. We organize previous methods mainly by (1) what input modality is used and (2) which dataset is pre-trained on. We provide a comprehensive comparison in Table 4. We first divide all entries by the input modality they used. Inside each modality, we arrange the entries w.r.t. the performance on UCF-101 by ascending order. We notice there is inconsistency in previous work on reporting results with different splits of UCF-101 and HMDB-51, so we report on both split-1 and 3 splits average. For fine-tuning, CVRL significantly outperforms methods using **Vision** modality only. Compared with multimodal methods using **Vision** and **Flow** [29, 30], **Vision** and **Text** [45], CVRL still ranks top. Multimodal methods using **Vision** and **Audio** are able to achieve better performance starting from GDT [51] on AS, while it is worth to point out that the pre-trained dataset is 9× larger than K400 which we pre-train CVRL on with single **Vision** modality. For CVRL pre-trained on K600, it is only worse than the best model of [3, 51, 52, 2] on UCF-101 and outperforms the best model of [3, 52] on HMDB-51, where their pre-training datasets are 108× to 174× larger than K600. For linear evaluation, CVRL is better than all single and multi modal methods with the only exception of MMV [2] on UCF-101. On HMDB-51, CVRL demonstrates very competitive performance, outperforming all methods using vision modality and multimodal methods of [29, 30, 45, 3]. In conclusion, CVRL shows competitive performance on downstream action classification, compared with single and mutilmodal video self-supervised learning methods.

Downstream action detection. We conduct experiments on AVA [27] dataset which benchmarks methods for detecting when an action happens in the temporal domain and where it happens in the spatial domain. Each video in AVA is annotated for 15 to 30 minutes and we consider this as an important experiment to demonstrate the transferability of CVRL learned features. We adopt Faster-RCNN [54] and replace the 2D ResNet backbone with our video encoder in Table 1. Following [17, 73], we compute region-of-interest (RoI) features by using a 3D RoIAlign on the features from the last conv block. We then perform a temporal average pooling followed by a spatial max pooling, and feed the feature into a sigmoid-based classifier for mutli-label prediction. We use pre-trained weights to initialize the video encoder, and fine-tune all layers for 12 epochs. We report mean Average-Precision (mAP) in Table 5, where CVRL shows better performance than baselines.

4.3. Ablation Study

We conduct extensive ablation studies for CVRL based on 200 epochs pre-training on Kinetics-400 and report top-1 linear evaluation accuracy on Kinetics-400.

Method	Pre-train data (duration)	Mod.	Top-1 Acc. (%) UCF	HMDB
Fine-Tuning				
MotionPred† [66]	K400 (28d)	V	61.2	33.4
3D-RotNet‡ [37]	K400 (28d)	V	64.5	34.3
ST-Puzzle‡ [39]	K400 (28d)	V	65.8	33.7
ClipOrder† [72]	K400 (28d)	V	72.4	30.9
DPC‡ [28]	K400 (28d)	V	75.7	35.7
PacePred† [67]	K400 (28d)	V	77.1	36.6
MemDPC§ [29]	K400 (28d)	V	78.1	41.2
CBT‡ [59]	K600+ (273d)	V	79.5	44.6
SpeedNet† [6]	K400 (28d)	V	81.1	48.8
VTHCL§ [73]	K400 (28d)	V	82.1	49.2
DynamoNet‡ [13]	YT8M(13y)	V	88.1	59.9
SeCo§ [74]	K400 (28d)	V	88.3	55.6
MemDPC§ [29]	K400 (28d)	VF	86.1	54.5
CoCLR† [30]	K400 (28d)	VF	90.6	62.9
MIL-NCE‡ [45]	HT (15y)	VT	91.3	61.0
AVTS‡ [40]	AS (240d)	VA	89.0	61.6
MMV-VA‡ [2]	AS+HT (16y)	VA	91.1	68.3
XDC‡ [3]	AS (240d)	VA	91.2	61.0
GDT‡ [51]	AS (240d)	VA	92.5	66.1
XDC‡ [3]	IG65M (21y)	VA	94.2	67.4
GDT‡ [51]	IG65M (21y)	VA	95.2	72.8
Elo§ [52]	YT8M (13y)	VFA	93.8	67.4
MMV‡ [2]	AS+HT (16y)	VAT	95.2	75.0
CVRL‡(R3D-50)	K400 (28d)	V	**92.2**	**66.7**
CVRL†(R3D-50)	K400 (28d)	V	**92.9**	**67.9**
CVRL‡(R3D-50)	K600 (44d)	V	**93.4**	**68.0**
CVRL†(R3D-50)	K600 (44d)	V	**93.6**	**69.4**
CVRL‡(R3D-152(2×))	K600 (44d)	V	**93.9**	**69.9**
CVRL†(R3D-152(2×))	K600 (44d)	V	**94.4**	**70.6**
Linear Evaluation				
MemDPC§ [29]	K400 (28d)	VF	54.1	30.5
CoCLR† [30]	K400 (28d)	VF	77.8	52.4
MIL-NCE‡ [45]	HT (15y)	VT	83.4	54.8
XDC‡ [3]	AS (240d)	VA	85.3	56.0
MMV-VA‡ [2]	AS+HT (16y)	VA	86.2	61.5
Elo§ [52]	YT8M (13y)	VFA	-	64.5
MMV‡ [2]	AS+HT (16y)	VAT	91.8	67.1
CVRL‡(R3D-50)	K400 (28d)	V	**89.2**	**57.3**
CVRL†(R3D-50)	K400 (28d)	V	**89.8**	**58.3**
CVRL‡(R3D-50)	K600 (44d)	V	**90.6**	**59.7**
CVRL†(R3D-50)	K600 (44d)	V	**90.8**	**59.7**

Table 4. **Downstream action classification results on UCF-101 and HMDB-51.** CVRL shows competitive performance compared with single and muitl-modal methods, by using only the **V**ision modality on K400 and K600. † indicates split-1 accuracy, ‡ indicates averaged accuracy on 3 splits, § indicates evaluation split(s) not mentioned in paper.

Method	Rand.	ImageNet infla.	SimCLR infla.	CVRL	Sup.
mAP	6.9	14.0	14.2	**16.3**	19.1

Table 5. **Downstream action detection results on AVA.** We report mean Average-Precision (mAP) to assess the performance. CVRL outperforms ImageNet inflated and SimCLR inflated. All methods use R3D-50 as the backbone.

Backbone	Hidden layers	Accuracy (%) top-1	top-5
R3D-50	0	54.7	79.1
	1	62.5	84.5
	2	63.0	84.8
	3	**63.8**	**85.2**

Table 6. **Ablation on hidden layers.**

Backbone	Batch size	Accuracy (%) top-1	top-5
R3D-50	256	60.2	82.5
	512	62.9	84.7
	1024	**63.8**	**85.2**
	2048	61.8	84.2

Table 7. **Ablation on batch size.**

Backbone	Pretrain epochs	Accuracy (%) top-1	top-5
R3D-50	100	58.8	81.8
	200	63.8	85.2
	500	65.6	86.5
	800	**66.1**	**86.8**

Table 8. **Ablation on pre-training epochs.**

Temporal augmentation	Spatial augmentation	Temporal consistency	Accuracy (%) top-1	top-5
✓			33.0	57.3
	✓		40.9	66.6
✓	✓		52.3	76.0
✓	✓	✓	**63.8**	**85.2**

Table 9. **Ablation study on data augmentation.**

Temporal interval sampling distribution. As shown in Figure 3, we experiment with monotonically decreasing distributions (a-c), uniform distribution (d) and monotonically increasing distributions (e-f). We find decreasing distributions are better. We also compare different power functions for decreasing distribution and choose a simple exponent of 1 (*i.e.*, linear) due to its simplicity and best performance.

Spatial and temporal augmentation. We conduct an ablation study on the proposed temporally consistent spatial augmentation. From results in Table 9, we have three major observations. First, both temporal and spatial augmentations are indispensable. Specifically, using both temporal and spatial augmentations yields 52.3% top-1 accuracy, significantly outperforming the same model pre-trained with temporal augmentation only (33.0%) or spatial augmentation only (40.9%). Second, the proposed temporally consistent module plays a critical role in achieving good performance. Adding temporal consistency further improves the top-1 accuracy to 63.8% by a large margin of 11.5% over 52.3%. Third, spatial augmentations, which are ignored to some degree in existing self-supervised video representation learning literature, not only matter, but also contribute more than the temporal augmentations.

More training data. We study whether using more data would improve the performance of CVRL. We design an evaluation protocol by first pre-training models on different amount of data (K600 and K400) with same iterations to remove the advantage brought by longer training, and then comparing the performance on same validation set (K400 val). We verify that the training data of K600 has no overlapping video ids with the validation set of K400. We present results of 46k (200 K400 epochs), 184k (800 K400 epochs) and 284k (800 K600 epochs) pre-training iterations in Figure 4. We find more training data in K600 is beneficial, demonstrating the potential of CVRL's scalability on larger unlabeled datasets.
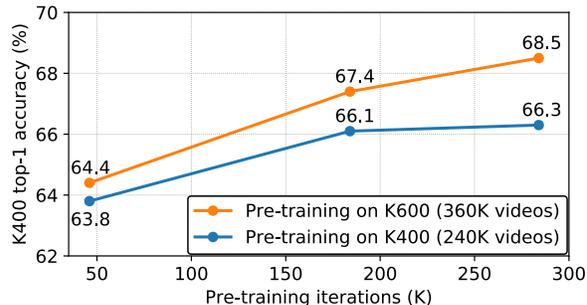

Figure 4. **More data is beneficial for CVRL.** All models are evaluated on the validation set of K400 to provide a fair comparison.

Layers of projection head. We experiment with different number of hidden layers. Unlike [11], we only use different layers in pre-training and perform the linear evaluation on top of the same backbone by removing the entire projection head. In Table 6, we can see using 3 hidden layers yields the best performance and we choose this as our default setting.

Batch size. The batch size determines how many negative pairs we use for each positive pair during training. Our experimental results show that a batch size of 1024 already achieves high performance. Larger batch sizes could negatively impact the performance as shown in Table 7.

Pre-training epoch. As presented in Table 8, we experiment with pre-training epochs varying from 100 to 800 and find consistent improvement with longer pre-training epochs. We choose 800 epochs as our default setting.

5. Conclusion

This work presents a Contrastive Video Representation Learning (CVRL) framework leveraging spatial and temporal cues to learn spatiotemporal representations from unlabeled videos. Extensive studies on linear evaluation, semi-supervised learning and various downstream tasks demonstrate promising results of CVRL. In the future, we plan to apply CVRL to a large set of unlabeled videos and incorporate additional modalities into our framework.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. 6

[2] Jean-Baptiste Alayrac, Adrià Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. In *NeurIPS*, 2020. 2, 6, 7

[3] Humam Alwassel, Dhruv Mahajan, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *NeurIPS*, 2020. 2, 7

[4] Yuki M Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. In *NeurIPS*, 2020. 2

[5] Yutong Bai, Haoqi Fan, Ishan Misra, Ganesh Venkatesh, Yongyi Lu, Yuyin Zhou, Qihang Yu, Vikas Chandra, and Alan Yuille. Can temporal information help with contrastive self-supervised learning? *arXiv preprint arXiv:2011.13046*, 2020. 1

[6] 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, 7

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 1, 2

[8] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. 2, 5, 6

[9] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 1

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1, 2, 3, 4, 5, 6

[11] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *NeurIPS*, 2020. 8

[12] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *NeurIPS*, 2020. 12

[13] Ali Diba, Vivek Sharma, Luc Van Gool, and Rainer Stiefelhagen. Dynamonet: Dynamic action and motion network. In *ICCV*, 2019. 7

[14] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015. 2

[15] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *CVPR*, 2019. 3

[16] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, 2020. 1

[17] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 1, 3, 5, 7

[18] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, 2017. 2

[19] Chuang Gan, Boqing Gong, Kun Liu, Hao Su, and Leonidas J Guibas. Geometry guided convolutional neural networks for self-supervised video representation learning. In *CVPR*, 2018. 2

[20] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017. 6

[21] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, 2019. 6

[22] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 2

[23] Rohit Girdhar, Du Tran, Lorenzo Torresani, and Deva Ramanan. Distinit: Learning video representations without a single labeled video. In *ICCV*, 2019. 3

[24] 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. 2, 3, 6

[25] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 5

[26] 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. In *NeurIPS*, 2020. 1, 2

[27] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 5, 7

[28] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *ICCV Workshops*, 2019. 1, 2, 7

[29] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *ECCV*, 2020. 2, 7

[30] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *NeurIPS*, 2020. 2, 7

[31] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, 2018. 1, 3

[32] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 2, 4, 5

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 3, 5

[34] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *CVPR*, 2019. 5

[35] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019. 2

[36] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 1

[37] Longlong Jing and Yingli Tian. Self-supervised spatiotemporal feature learning by video geometric transformations. *arXiv preprint arXiv:1811.11387*, 2018. 2, 7

[38] 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. 2, 5, 6, 12

[39] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, 2019. 1, 2, 4, 7

[40] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018. 2, 7

[41] 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, 6

[42] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, 2017. 1, 2, 4

[43] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016. 1, 2

[44] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1

[45] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020. 2, 7

[46] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019. 6

[47] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 2

[48] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2, 3

[49] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *CVPR*, 2017. 3

[50] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 2

[51] Mandela Patrick, Yuki M Asano, Ruth Fong, João F Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal self-supervision from generalized data transformations. *arXiv preprint arXiv:2003.04298*, 2020. 2, 7

[52] AJ Piergiovanni, Anelia Angelova, and Michael S. Ryoo. Evolving losses for unsupervised video representation learning. In *CVPR*, 2020. 2, 7

[53] Senthil Purushwalkam1 and Abhinav Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. In *NeurIPS*, 2020. 3

[54] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 5, 7

[55] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *ACM MM*, 2007. 1

[56] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *ICRA*, 2018. 3

[57] 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, 6

[58] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015. 1, 2

[59] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*, 2019. 2, 7

[60] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019. 2

[61] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020. 2

[62] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. In *NeurIPS*, 2020. 2

[63] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 1

[64] Curtis R Vogel. *Computational methods for inverse problems*. SIAM, 2002. 12

[65] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *ECCV*, 2018. 3

[66] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *CVPR*, 2019. 2, 7

[67] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In *ECCV*, 2020. 2, 4, 7

[68] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015. 3

[69] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019. 3

[70] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 2

[71] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018. 1

[72] 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. 1, 2, 4, 7

[73] Ceyuan Yang, Yinghao Xu, Bo Dai, and Bolei Zhou. Video representation learning with visual tempo consistency. *arXiv preprint arXiv:2006.15489*, 2020. 1, 2, 5, 6, 7

[74] 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. 6, 7

[75] 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. 4

[76] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*, 2019. 2

[77] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 2

[78] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017. 2