A Large-Scale Study on Unsupervised Spatiotemporal Representation Learning

Christoph Feichtenhofer Haoqi Fan Bo Xiong Ross Girshick Kaiming He

Facebook AI Research (FAIR)

Abstract

We present a large-scale study on unsupervised spatiotemporal representation learning from videos. With a unified perspective on four recent image-based frameworks, we study a simple objective that can easily generalize all these methods to space-time. Our objective encourages temporally-persistent features in the same video, and in spite of its simplicity, it works surprisingly well across: (i) different unsupervised frameworks, (ii) pre-training datasets, (iii) downstream datasets, and (iv) backbone architectures. We draw a series of intriguing observations from this study, e.g., we discover that encouraging long-spanned persistency can be effective even if the timespan is 60 seconds. In addition to state-of-the-art results in multiple benchmarks, we report a few promising cases in which unsupervised pre-training can outperform its supervised counterpart. Code will be made available at `https://github.com/facebookresearch/SlowFast`.

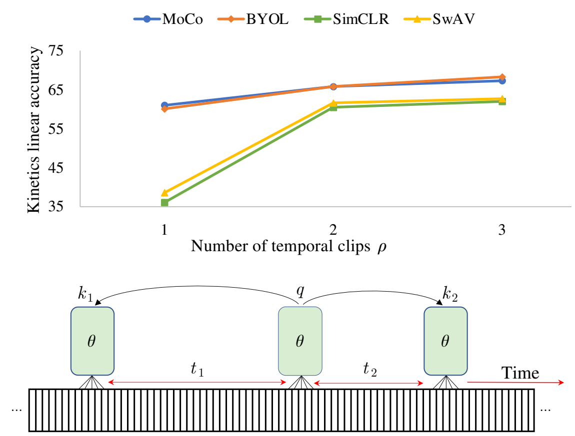


Figure 1. Learning to maximize the similarity between different temporal clips of the same video encourages feature persistency over time. A query clip (q) is matched to multiple key clips ($k_1, k_2, \ldots$) that are temporally shifted. This method can be incorporated into several unsupervised learning frameworks (MoCo [36], SimCLR [12], BYOL [32], SwAV [9]). The figure on the top shows that increasing the number (ρ) of temporal clips improves representation quality for all these frameworks.

1. Introduction

A series of recent methods on unsupervised representation learning from images [36, 12, 32, 9] are based on maximizing a similarity objective for different views of the same image under data augmentations [18, 90]. In addition to the *artificial* augmentations on images, videos can provide *natural* augmentations of visual content under various changing factors, such as motion, deformation, occlusion, and illumination. This work aims to generalize these image-based methods [36, 12, 32, 9] into space-time.

We study a simple objective that can be easily incorporated into these image-based methods. Our hypothesis is that the visual content is often *temporally-persistent* along a timespan in the video. This persistency may involve an action (*e.g.*, a person dancing), an object (*e.g.*, an individual person, who transitions from running to walking), and a scene (*e.g.*, a room with people moving), covering short to long spans, with different levels of visual invariance (action, object, scene). Our objective simply encourages *the visual representations in different clips of the same video to be similar*. We empirically find that this objective works well across different unsupervised frameworks (MoCo [36], SimCLR [12], BYOL [32], SwAV [9]), either *with* or *without* using dissimilar (negative) samples.

Our objective is a natural generalization of *crops* in images [18, 90] to *clips* in videos. This allows us to make use of the recent unsupervised learning frameworks with minimal modifications. We aim to learn a high-level representation of the categorical semantics present in a video by enforcing persistency of the representation over space-time. We investigate factors such as the effective timespan, t, between positives, and number of temporal clips, ρ, to find that longer timespans (up to a minute) and multiple samples are beneficial for downstream performance (Fig. 1).

Our unsupervised training is performed on large-scale data, including Kinetics [47] (240k videos) and three versions of *million-scale* Instagram sets. In addition to standard linear probing, we evaluate representation quality on multiple classification and detection downstream datasets, *e.g.*, Charades [76], Something-Something [31], and AVA [33].

Our results suggest that unsupervised pre-training can achieve competitive performance in videos, and it can surpass the supervised pre-training counterparts in a few cases. Finally, our study also reveals room for improvement along multiple directions.

In summary, our large-scale study involves the following five aspects:

(i) Four unsupervised learning frameworks (MoCo [36], SimCLR [12], BYOL [32], SwAV [9]) viewed from a unified perspective, and incorporated with a simple temporal persistency objective;

(ii) Three pre-training datasets, including the relatively well-controlled Kinetics [47] and the relatively "in-the-wild" Instagram sets at million-scale;

(iii) Six downstream datasets/tasks for evaluating representation quality;

(iv) Ablation experiments on different factors, such as temporal samples, contrastive objective, momentum encoders, training duration, backbones, data augmentation, curated *vs.* uncurated, trimmed *vs.* untrimmed, *etc.*; and

(v) State-of-the-art results of unsupervised video representation learning on established benchmarks, UCF-101 [78], HMDB51 [50] and Kinetics-400 [47] .

2. Related Work

Unsupervised learning in images has been actively researched recently with approaches focusing on various pretext tasks related to color- or patch-based processing [68, 96, 17, 65], instance discrimination with contrastive objectives [18, 90, 84, 40, 41, 46, 36, 97, 12, 82] and ones that focus on positive pairs [8, 9, 32].

Unsupervised learning in videos has followed a similar trajectory with earlier methods focusing on predictive tasks based on motion, color and spatiotemporal ordering [29, 43, 1, 44, 79, 86, 61, 85, 59, 58, 21, 51, 87, 67, 22, 48, 93, 16, 88, 71, 45], and contrastive objectives with visual [75, 80, 34, 53, 28, 94] and audio-visual input [66, 4, 5, 49, 3, 69, 70].

Several recent ones [28, 35, 3, 69, 2, 72, 94, 63] relate to image-based approaches [36, 8, 12, 90]. With some of them using additional modalities of optical-flow [82, 35], audio [3, 69, 2, 63] and text [80, 2] to transfer supervision from one modality to another.

In relation to these previous efforts, our work studies purely visual unsupervised learning from video and tries to compare the meta-methodologies on common ground.

Evaluation protocols and backbones in most image-based approaches have converged to ResNet-50 [39] encoders with ImageNet linear-classification protocol, and several smaller downstream tasks [36, 12, 32, 9] for evaluation. In video understanding research, the field has not yet converged and is using different backbones with focus on fine-tuning performance on two relatively small datasets [78, 50]. We investigate this aspect by looking at different encoders and 6 different downstream benchmarks for evaluation.

3. Approach

The objective of this work is to study several recent unsupervised representation learning methodologies to train a spatiotemporal encoder f_θ, exploring implementation details and comparing them on a common ground to measure their efficacy in video understanding. We focus on two contrastive approaches using positive and negative samples: SimCLR [12] and MoCo [36], as well as two approaches that solely rely on positives, BYOL [32] and SwAV [9] (Sec. 3.2).

These approaches were originally presented for learning image representations, and they all share the objective of learning invariant features across different views (crops/augmentations) of the spatial image input. In this paper, this idea is extended to the temporal domain. Our core idea is to learn an encoder f_θ that produces embeddings which are persistent in space-time, over multiple (ρ) temporally distant clips of the same video. This is related to Slow Feature Analysis [89] where the objective is to minimize the representations' temporal derivative over the input. The general idea of learning temporally persistent features is not new and has been proposed in the past with similar motivation *e.g.*, [6, 62, 29].

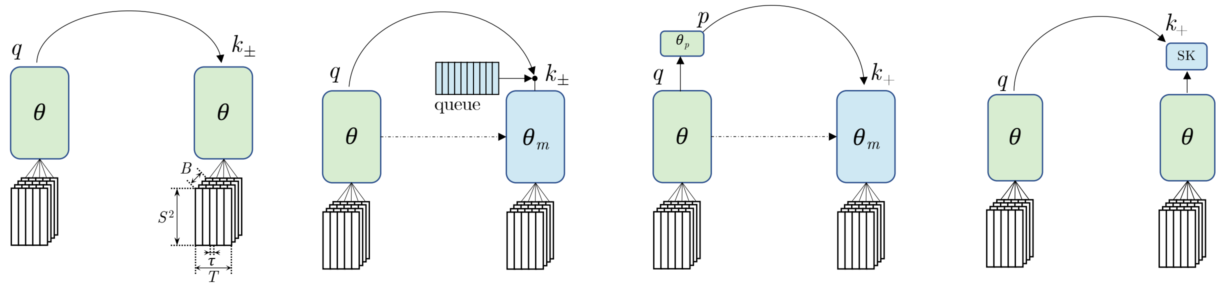
3.1. Persistent temporal feature learning

Our framework takes different augmented clips x of an unlabeled video and passes them through an encoder f_θ with weights θ to obtain corresponding embeddings $q = f_\theta(x)$. The encoder is spatiotemporal ConvNet, by default a ResNet-50 (R-50) [39], Slow-only pathway of SlowFast Networks [20], which is a 3D ResNet-50 [39] without temporal pooling in convolutional feature maps, followed by an MLP projection head, that produces and output of dimension d.

The input clips are stacks of RGB frames of size $3 \times T \times S^2$ for temporal $\times$ spatial dimensions, which are sampled with temporal stride τ , *i.e.*, the encoder processes only one out of τ frames of the raw video. Therefore, $T \times \tau$ define the timespan and resolution of the encoder.

Given a minibatch of B videos, our framework creates a set of ρB positive examples by sampling ρ clips from the videos. The learning methodologies studied in this section maximize similarity of a "query" sample q with a set of positive "key" samples $\{k^+\}$ that are encoded versions of different clips of the same video as q is computed from. Fig. 1 illustrates an example where ρ=3 clips are used.

The next section describes how the contrastive and non-contrastive unsupervised representation learning methodologies are exemplified.



(a) SimCLR (b) MoCo (c) BYOL (d) SwAV

Figure 2. **Conceptual comparison of four unsupervised learning mechanisms applied to video**. The inputs consist of ρ=2 clips from B videos. Each clip is a stack of T frames with temporal stride τ and spatial resolution S^2. Each method trains encoder weights θ by computing a positive loss component w.r.t. to the other clips of the same video. SimCLR (a) and MoCo (b) use a contrastive loss with negatives coming from different videos in the batch or a a queue. respectively. MoCo (b) and BYOL (c) use extra momentum encoders with weights θ_m being moving averages of the trained θ. SwAV (d) uses a Sinkhorn-Knop (SK) transform to generate the positive targets.

3.2. Unsupervised learning frameworks

Contrastive learning maximizes the similarity of a sample q with positive ones $\{k^+\}$ and minimizes similarity to negative ones $\{k^-\}$. The contrastive approaches in this paper use the InfoNCE [84] objective,

$$\mathcal{L}_q = -\log \frac{\sum_{k\in\{k^+\}} \exp\left(\mathrm{sim}(q,k)/\alpha\right)}{\sum_{k\in\{k^+,k^-\}} \exp\left(\mathrm{sim}(q,k)/\alpha\right)}, \tag{1}$$

with α being a temperature hyper-parameter for scaling and $\{k^+\}$ are embedded clips of the same video as q. All the embeddings are ℓ_2 normalized and dot product (cosine) similarity is used to compare them $\mathrm{sim}(q,k) = q^\top k/\|q\|\|k\|$.

SimCLR [12] (Fig. 2a) uses the embeddings of clips of other videos in the minibatch as negatives $\{k^-\}$.

MoCo [36] (Fig. 2b) is a method that uses an explicit momentum encoder which parameters, θ_m, are a moving average $\theta_m \leftarrow m\theta_k + (1-m)\theta$ with m a momentum parameter. In eq. (1) MoCo uses this encoder to compute the positive embeddings $\{k^+\}$ from clips of the same video as q, and negative embeddings $\{k^-\}$ are taken from a queue that stores embeddings of clips from previous iterations. There is no backpropagation into the momentum-encoder weights θ_m.

BYOL [32] (Fig. 2c) can be viewed as a form of MoCo that does not use negative samples, but an extra predictor MLP with weights θ_p, which is stacked on top of f_θ's MLP head. For a sample $q = f_{\theta_p}(f_\theta(x))$, BYOL minimizes negative cosine similarity,

$$\mathcal{L}_q = -\sum_{k\in\{k^+\}} \mathrm{sim}(q,k) = -\sum_{k\in\{k^+\}} q^\top k^+/\|q\|\|k\|, \tag{2}$$

with $\{k^+ = f_{\theta_m}(x^+)\}$ being embedded clips x^+ from the same video as q, encoded with momentum weights θ_m.

SwAV [9] (Fig. 2d) can be viewed as a form of SimCLR that does not use negative samples. SwAV first performs a linear mapping of the positive embeddings q, k^+ to learned prototypes $\tilde{q}, \tilde{k}^+$ and then transforms the targets with an extra Sinkhorn-Knopp (SK) step. Then the SwAV loss is

$$\mathcal{L}_q = D_{\mathrm{KL}}(\tilde{q}\|SK(\tilde{k}^+)), \tag{3}$$

where D_{KL} is the The Kullback-Leibler divergence and gradients are not back-propagated through the SK operation.

Compared to SimCLR and MoCo, in BYOL and SwAV, q and k are not typical "query" and "key" samples (but rather "source" and "target" samples); however, for consistency we use q, k terminology in notation for all methods.

Implementation specifics. We implement the methods with a *symmetric* loss, as in original SimCLR, BYOL and SwAV, where every input clip is used to produce a loss (and gradient) signal. For each of the $\rho \geq 2$ clips, we compute q, while all *other* $\rho-1$ clips of the same video are used as $\{k^+\}$ to evaluate sub-loss $\mathcal{L}_q$ and the symmetric loss is the average over all ρ sub-losses. Thus, for MoCo and BYOL, every input clip is processed by both encoders.

For MoCo and BYOL, our symmetric loss is aggregated *sequentially* which implies that memory consumption for $\rho > 2$ equals to a single clips' forward and backward pass, since these methods do not backpropagate through the momentum encoder. For SimCLR and SwAV the overall loss is evaluated in *parallel* across all clips and therefore memory consumption grows linearly with the number of clips used.

All details on implementation and pre-training are in §B.1.

4. Experiments

Datasets. Unless otherwise noted, we perform unsupervised pre-training on Kinetics-400 [47] (K400) with ~240k training videos in 400 human action categories.

data	#videos	t_{median}	t_{mean}	t_{std}	t_{min}	t_{max}
Kinetics-400 (K400) [47]	240K	10.0	9.3	1.7	1.0	10.0
IG-Curated [24]	1M	18.9	26.3	19.8	1.5	60.0
IG-Uncurated	1M	29.4	35.3	38.4	0.5	600.0
IG-Uncurated-Short	1M	13.0	13.1	1.6	10.0	15.9

Table 1. **Pre-training data statistics with timings in seconds**.

To study learning from "*in-the-wild*" videos from the web, we pre-train the methods on Instagram videos: IG-Curated [24], a dataset with hashtags similar to K400 classes; IG-Uncurated which has videos taken randomly from Instagram; and IG-Uncurated-Short which is similar, but has constrained duration. Each dataset has 1M videos.

Table 1 shows dataset statistics of all datasets used for unsupervised pre-training. Most of Kinetics videos are of 10 seconds in duration. IG-Curated is a dataset with Instagram videos that have an average duration t_{mean} of 26.3 seconds and a standard deviation t_{std} of 29.8 seconds. The maximum duration t_{max} is 60s. IG-Uncurated contains videos taken randomly from Instagram, with larger deviation in length and maximum duration of 10 minutes (600s). IG-Uncurated-Short is a dataset consisting of random Instagram videos that have a duration between 10 and 16 seconds, to study the effect of a fixed duration and the assumption that short videos may hold more useful information for pre-training.

Evaluation protocols. For evaluation we use two protocols.

The first one is common to evaluate unsupervised image representations [36, 12]. It validates the *linear classifier* performance based on frozen encoder features that are taken from the global average pooling layer. We report top-1 classification accuracy (%) on the K400 validation set.

The second protocol reports *finetuning* accuracy on the first split of the UCF101 dataset [78] which contains 13k videos in 101 human action classes; this is a common procedure used to evaluate unsupervised video representations. Finally, we also report *finetuning* accuracy on AVA [33], Charades [76], Something-Something [31] and HMDB51 [50].

Architecture. By default, we use a R-50 [39] following the Slow pathway in [20] with clips of T=8 frames sampled with stride τ=8 from 64 raw-frames of video. The supervised performance for training 200, 400, 800 epochs on K400 is 74.7%, 74.3% and 72.7%, respectively, and does not improve for training longer due to overfitting.

Implementation details. We follow default settings in video classification [20]. Specifics on the approaches, their training and evaluation and the impact of implementation on performance are provided in §B and §A.3, respectively.

4.1. Persistent temporal learning

Here, we investigate the impact of learning spatiotemporal *vs*. only spatial persistent features. Table 2 shows the accuracy of the four methods when trained for 200 epochs on K400, and evaluated on K400 (linear) and UCF101 (finetuned), *i.e*. our default setting.

	MoCo		BYOL		SimCLR		SwAV	
ρ	K400	UCF101	K400	UCF101	K400	UCF101	K400	UCF101
1	61.0	90.8	60.6	91.2	36.1	84.2	38.6	74.7
2	65.8	91.0	65.8	92.7	60.5	88.9	61.6	87.3
3	67.3	92.8	68.3	93.8	62.0	87.9	62.7	89.4
4	67.8	93.5	**68.9**	**93.8**	out of memory			

Table 2. **Number of temporal clips** ρ. Data: **K400**, 200 epochs. Learning temporally persistent features ($\rho \geq 2$) is effective.

	MoCo		BYOL		SimCLR		SwAV	
ep	K400	UCF101	K400	UCF101	K400	UCF101	K400	UCF101
50	52.6	84.6	30.2	78.5	45.7	79.7	55.9	81.4
100	60.5	89.5	47.6	88.6	57.3	85.6	59.4	85.5
200	65.8	91.0	65.8	92.7	60.5	88.9	61.6	87.3
400	67.4	92.5	66.9	92.8	62.0	87.9	62.9	88.3
800	**67.4**	**93.2**	66.2	93.6	61.8	88.4	63.2	89.5

Table 3. **Training duration in epochs (ep)**: Dataset: **K400**, ρ=2. Training longer brings consistent gains for all methods up to 400 epochs and saturates for K400 but not for UCF101 at 800ep. SwAV is the strongest performer for short training (50ep).

Temporal augmentation. The first row in Table 2, ρ=1, uses two spatial crops at the same temporal instance, while the ρ=2 row uses clips at different temporal locations as positives; therefore, learns persistent features in time. This difference has a large impact on performance, especially for SimCLR (60.5 $\rightarrow$ 36.1) and SwAV (61.6 $\rightarrow$ 38.6) performance degrades significantly when sampling positives from the same temporal instance (ρ=1).

More clips are beneficial. The remaining rows in Table 2 show that accuracy is further increasing with the number of temporal samples per video, *e.g*. at ρ=4 the best accuracy is achieved with **BYOL** at **68.9**% K400 and **93.8**% UCF101.

Negatives do not help but momentum encoders do. When comparing the methods in Table 2, we see that:

(i) There is no clear performance difference between contrastive/non-contrastive methods. This indicates that learning space-time persistence within a video is key for the methods, but learning in-persistence across videos is not.

(ii) There is a clear difference of ~4% on K400 between methods that employ momentum encoders (MoCo, BYOL), *vs*. these that do not (SimCLR, SwAV).

Increasing the number of clips per training iteration increases training cost, so it is reasonable to compare it to training more epochs. Table 3 is studying the base case ρ=2 for various number of epochs (ep).

Overall, the results show that there is a clear gain for training longer which has been also observed in image-related tasks [12, 36, 32, 9]. BYOL performs the worst when training short durations. This might be related to hyper-parameter settings which we do not adjust for this experiment (the original implementation [32] uses different hyper-parameters for different number of training epochs).

4.2. Timespan between positives

All experiments with $\rho \geq 2$ so far were using global temporal sampling of positives, which means that the clips can be sampled at unconstrained temporal locations from the input video. This might be counter-productive because if there is a long duration that has passed between a pair of positive clips they might no longer share the same semantic context for learning high-level features corresponding in time.

t_{max} in seconds	0	2	3	4	5	8	10
K400 acc in %	60.6	65.2	65.7	65.8	65.8	65.6	65.8

(a) Dataset: **K400**, 200 epochs training.

t_{max} in seconds	4	8	16	32	60
K400 acc in %	62.7	63.1	63.1	63.9	64.1

(b) Dataset: **IG-Curated-1M**, 50 epochs training.

t_{max} in seconds	12s	24	36	48	600
K400 acc in %	59.3	59.2	59.9	59.6	58.9

(c) Dataset: **IG-Uncurated-1M**, 50 epochs training.

Table 4. **Maximum frame distance for positives**. Method: **BYOL**, $\rho = 2$. Training is surprisingly robust with increasing accuracy for increased distance between samples. Accuracy only (mildly) degrades when sampling positives that are more than 36 seconds apart when using uncurated (random) videos.

This experiment is concerned with the maximum distance between the positive training samples. We use BYOL pre-training on K400, IG-Curated-1M and IG-Uncurated-1M and report 400 linear readout accuracy in Table 4.

Table 4a shows performance for increasing the maximum temporal distance between positives in **K400** pre-training. It can be seen that using positives from the same time (t_{max}=0) degrades perforance b ~5% but other than that performance is relatively robust up to global sampling of positive clips from the whole video (t_{max}=10s). This is interesting as it seems that a long-temporal correspondence objectives does not hurt performance (but also does not boost it).

Table 4b shows performance for increasing the temporal distance between positive samples on **IG-Curated-1M**. This dataset has a maximum duration of 60 seconds; statistics are in Table 1. Table 4b shows that increasing the maximum duration between positive pairs is beneficial for performance and unrestricted sampling of positives is the best with 64.1% top-1 accuracy for evaluation on K400. This is especially interesting, as it shows that even longer videos benefit from global sampling. There is *no benefit from restricting the time window of positives*, which can be interpreted as the objective of learning extremely-slow features [89] that do not change over 60 seconds of video. Long-temporal-distance samples might also increase robustness of the model by providing "*hard-positive*" samples for learning. Note that here the videos are still sampled according to hashtags related to K400 classes [24]; therefore, the conjecture might be biased.

Finally, we are looking at the **IG-Uncurated-1M** dataset which consists of a random sampling of 1M videos from Instagram. These videos can be between 0.5s and 10 minutes of duration. Most of the videos however are much shorter than 10 minutes, with a mean duration of 35.3 seconds and a standard deviation of 38.4 seconds (Table 1). For this data, Table 4c shows the results of progressively increasing the maximum timespan between positive samples. It can be observed that increasing the maximum distance between positives up to 36 seconds is beneficial and beyond that performance decreases, but only slightly, even when performing global sampling of positives (the default).

4.3. Backbone architectures

So far all experiments were using a R-50, 8×8 Slow pathway [39, 20] as backbone. The next set of ablations studies different architectures for the spatiotemporal encoder.

		training			sup.	**MoCo** (ρ=2)	
backbone	$T \times \tau$	FLOPs	Param	s/iter	K400	K400	UCF101
R-50	8×8	41.7G	31.8M	1.6s	74.7	65.8	91.0
R-18	8×8	20.0G	20.2M	1.2s	68.9	56.2	87.1
R-101	8×8	93.3G	51.4M	2.1s	75.8	67.7	92.4
R-50	16×4	83.5G	31.8M	2.5s	76.1	67.6	93.3
R-50	32×2	167.0G	31.8M	4.6s	76.3	67.8	94.2
R2+1D-18	32×2	48.5G	15.4M	4.0s	71.7	57.2	93.7
S3D-G	32×2	36.0G	9.1M	4.1s	74.7	63.2	94.5

Table 5. **Backbone comparison.** The ResNet [39] backbone (Slow pathway [20]) is used with different depth (R-18, R-50, R-101), input frames T and stride τ. R2+1D [83] and S3D-G [91] are commonly used backbones for unsupervised video representation learning with downstream evaluation on UCF101.

Table 5 compares different backbones for usage with MoCo in our default setting (ρ=2, 200 epoch pre-training on K400). From left to right, the table shows the input duration T, sampling-rate τ, FLOPs (at 224^2 spatial resolution) and parameters of these backbones, as well as the average duration for training one iteration of the MoCo algorithm (measured on a single machine with 8 V100 GPUs in `PySlowFast` [19] and `torchvision` decoder), the supervised performance on K400 and UCF101 (finetuned from K400), as well as the downstream performance for K400 linear evaluation and UCF101 finetuning.

The first observation in Table 5 is that for the Slow architecture [20], using shallower (R-18) or deeper (R-101) networks can influence supervised and downstream performance in a sizable manner, with MoCo, K400 evaluation benefiting from more parameters. Doubling the input framerate (8×8 → 16×4) boosts accuracy on UCF101.

The second observation is that R2+1D [83] has a large gap on Kinetics (71.7% supervised *vs*. 57.2% unsupervised), while being remarkably strong on UCF101 (93.7%). This gap is also observed for S3D-G [91]. The reason for this might be that UCF101 is a small dataset which is easy to overfit and can benefit from fewer parameters.

ep	MoCo K400	MoCo UCF101	BYOL K400	BYOL UCF101	SimCLR K400	SimCLR UCF101	SwAV K400	SwAV UCF101
50	64.8	91.1	64.1	93.5	55.5	86.4	61.0	89.0
200	69.0	93.4	60.2	92.7	56.9	86.6	64.3	91.2

(a) Training on **IG-Curated-1M**.

ep	MoCo K400	MoCo UCF101	BYOL K400	BYOL UCF101	SimCLR K400	SimCLR UCF101	SwAV K400	SwAV UCF101
50	61.8	90.9	58.9	90.1	52.1	85.1	56.0	86.7
200	65.4	91.9	57.9	91.6	51.9	85.3	58.8	87.8

(b) Training on **IG-Uncurated-1M**.

ep	MoCo K400	MoCo UCF101	BYOL K400	BYOL UCF101	SimCLR K400	SimCLR UCF101	SwAV K400	SwAV UCF101
50	61.0	89.6	62.3	91.4	53.61	86.4	55.0	86.2
200	64.5	91.0	57.0	90.9	55.97	86.9	58.4	87.2

(c) Training on **IG-Uncurated-Short-1M**.

Table 6. Training on curated (a), uncurated (b) and short duration video (c) data from the web. Longer training degrades performance for BYOL, possibly due to suboptimal hyper-parameters. ρ=2.

4.4. Uncurated data and video duration

In Table 6 we show the performance of all four methodologies on IG-Curated-1M (a), IG-Uncurated-1M (b) and IG-Uncurated-Short-1M (c) for pre-training with 50 and 200 epochs. We make the following observations:

(i) Among the methods MoCo performs the best with *e.g*. 69.0% *vs*. second-best 64.3% of SwAV on curated data (a).

(ii) MoCo and SwAV scale the best for training longer, gaining roughly 3-4% for 200ep *vs*. 50ep.

(iii) On uncurated data, MoCo and SwAV perform ~1% better on the unconstrained duration videos in Table 6b.

(iv) BYOL and SimCLR show better performance on IG-Uncurated-Short (10-16s videos) in Table 6c, seemingly benefiting from shorter videos, but there is no clear benefit from either longer or shorter duration among all methods.

(v) BYOL degrades performance for training longer which might be due to the requirement of different hyper-parameters for different schedules (as noted in Sec. 4.1).

We will return to this point in §A.1, where we show that increasing clips-size ρ can overcome this issue in BYOL, along with further studies on the trade-off against training more epochs, and dataset scale.

4.5. Data augmentations

Importance of augmentations. Augmentations can have a major impact on visual unsupervised feature learning [12, 14]. In Fig. 3, we ablate spatial cropping (S), temporal clipping (T) and radiometric color (C) augmentations from the four unsupervised learning methods (*e.g*. "T S C" are the baselines using all augmentations and removing "S C" equals ρ=1 in Table 2). We make three main observations:

(i) Among the methods, MoCo and BYOL perform most robust for using fewer augmentations; their advantage over SimCLR and SwAV might be related to the momentum encoder which can provide extra augmentation in training.

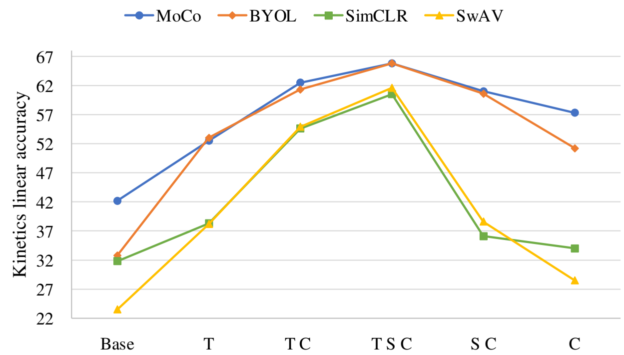


Figure 3. **Ablating augmentations.** We explore temporal (T), spatial (S), and color (C) augmentations to learn persistent features.

(ii) When minimizing the augmentations by resizing the shorter size of the video to the input size of 224 and only cropping along the long side of the video (Base in Fig. 3), MoCo still provides 42.2% K400 linear accuracy, over BYOLs' 32.4%, showing an advantage of the contrastive loss in a weak augmentation scenario.

(iii) Among the augmentations, learning temporal (T) persistency, has the largest impact on performance, except for MoCo which benefits more from color (C) (incl. grayscale) augmentations. Especially SimCLR and SwAV show significant drops in performance when removing T, *i.e*. when extracting positive clips from the same instance in time.

In the remainder of this section, we explore using stronger augmentations than the default ones in previous experiments. We perform the ablations with MoCo in the basic setting of $\rho = 2$, 200 epochs K400 pre-training.

color strength	grayscale probability	temporal difference	fps jitter	accuracy K400	accuracy UCF101
0.5	0.2			65.8	91.0
0.75	0.2			**66.0**	**92.1**
1.0	0.2			65.8	91.2
0.5	0.4			65.5	91.0
0.5	0.2	✓		**66.2**	91.3
0.5	0.2		✓	65.6	91.5

Table 7. **Radiometric augmentation**. Method: **MoCo**, 200 epochs, $\rho = 2$. Dataset: **K400**. Stronger color augmentation in K400 pre-training can especially benefit UCF101 (+1.3%).

Stronger color augmentation. In Table 7 color strength of 0.5 indicates the default one for MoCo [14], 0.75 and 1.0 increase the strength of randomly jittering brightness, contrast, saturation and hue proportionally.

Table 7 shows that increasing it to 0.75 can improve K400/UCF101 accuracy. Increasing the random grayscale probability from 0.2 to 0.4 does not provide an improvement on either of the datasets. However, using a temporal-difference augmentation which randomly (with probability 0.2) first converts the frames to grayscale and then subtracts them across time, can increase K400 accuracy by 0.4%. Finally, using frame-rate jittering of ±50% of the original frame-rate does not improve K400 but UCF101 slightly.

area [min, max]	aspect ratio	accuracy K400	UCF101
default [77, 39, 20]		65.8	91.0
[0.49, 0.76]		64.8	91.7
[0.49, 0.76]	✓	65.4	91.7
[0.20, 0.76]	✓	**66.8**	**91.8**
[0.20, 0.50]	✓	66.3	91.8
[0.20, 1.00]	✓	66.6	91.7
[0.08, 0.50]	✓	64.3	91.6
[0.08, 1.00]	✓	65.3	91.2

Table 8. **Cropping augmentation**. Method: **MoCo**, 200 epochs, $\rho = 2$. Dataset: **K400**. Stronger cropping and aspect ratio augmentation can be beneficial by +1.0% (K400) and 0.7% UCF101.

Spatial cropping. Our default implementation uses VGG-style [77, 39] cropping that randomly resizes the *shorter spatial side* of a video between [256, 320] pixels and takes a random 224^2 crop extended over time to extract a clip [20].

Since unsupervised learning might benefit from more aggressive cropping, we explore Inception-style [81] cropping with aspect ratio augmentation that is commonly used in unsupervised learning from images [36, 12, 32, 9]. This cropping procedure randomly resizes the input *area* between a minimum scale and a maximum scale and jitters aspect ratio between 3/4 to 4/3, before taking a 224^2 crop.

We do not change the cropping for downstream training, as this can drop accuracy significantly (by ∼2% on K400).

In Table 8 we ablate this approach for MoCo (the augmentation in the downstream evaluators are unchanged).

The first ablation shows the comparison of default cropping [77, 39] with a similar version that randomly crops a fraction between $[0.49, 0.76] = [224^2/320^2, 224^2/256^2]$ of the original area, instead of the short-side. The performance degrades by 1% on K400 linear evaluation. Randomly cropping based on area favors larger crops over the short-side resizing and we observe lower training error for this variant.

Next, adding aspect ratio augmentation can recover some of this performance (65.4%), and using a smaller minimum area of 0.2, with the maximum area of 0.76 leads to best performance of **66.8**%. Using the default values for Inception [81] training, [0.08, 1.00], appears to be too aggressive.

	MoCo (ρ=4)		**BYOL** (ρ=4)	
aug+	K400	UCF101	K400	UCF101
	67.8	93.5	68.9	93.8
✓	69.0	93.6	69.8	93.9

Table 9. **Stronger augmentations.** Data: K400, 200 epochs. "*aug+*" combines the best color and cropping augmentations from Table 7 and Table 8, respectively.

Combined augmentations. We pull together the best color and cropping augmentations in Tables 7 & 8, and train MoCo and BYOL with ρ=4 for 200ep on K400. The result shown as "*aug+*" in Table 9 can increase performance on K400 by ∼1%. Training the linear classifier of BYOL (ρ=4) for 100ep instead of 60ep leads to our best accuracy of **70.0**% on K400, which is 4.7% below the supervised R-50, Slow 8×8 accuracy of 74.7%.

4.6. Alternative downstream tasks

The gap between K400 and UCF101 accuracy in Sec. 4.3 question if solely looking at typical evaluation of UCF101 (or the smaller HMDB51) is enough to identify and rank approaches for unsupervised learning in video.

Table 10 studies several new downstream tasks for unsupervised representation learning in video. We use our MoCo, SimCLR, BYOL and SwAV models trained with ρ=3 for 200 epochs on K400 and evaluate their performance by finetuning on Charades [76], AVA [33], or Something-Something [31] (in addition to the K400 linear readout performance and UCF101 performance reported in Table 2). Details on implementation are given in §B.

The first two rows in Table 10 show the two main competitors for this evaluation: (i) training from scratch on the datasets and (ii) K400 pre-training. First, we observe that the supervised pre-trained backbones outperform the train-from-scratch counterpart significantly, as expected.

Downstream datasets. For **K400** pre-training and linear evaluation, its supervised counterpart has an advantage between 12.7% and 6.4% top-1 accuracy among the methods.

On **UCF101** unsupervised pre-training is only 1% lower than the supervised counterpart for BYOL (the strongest).

On **AVA** short-term action detection we observe that the BYOL pre-trained model is able to outperform the supervised counterpart by **+1.2**% mAP, when using the same, fixed region proposals [20]. This result is significant, as *e.g.* switching from K400 to K600 (nearly double the size of K400) pre-training on AVA leads to a smaller gains in performance [20]. Overall this is a surprising result as the tasks in K400 and AVA are similar [52], only that the *temporal* granularity of the actions in AVA is *finer* while their *semantic* granularity is *coarser*; *e.g.* "*shoot*" in AVA *vs.* "*playing paintball*" in Kinetics, which might be better captured by the BYOL objective which solely works on positive temporal samples of a video, without contrasting them to other videos ("*shoot*" might be a positive appearing in many different videos and contrasting them could be harmful to downstream performance). This line of thinking is supported with MoCo's (contrastive objective) performance that is 3.1% worse than BYOL on AVA. Similarly, SimCLR (contrastive) is worse than SwAV (non-contrastive) when benchmarked on AVA.

On **Charades**, long-term action classification, we observe the opposite. Here, the contrastive MoCo is clearly the best performer with 33.5% mAP (close to the supervised pre-training performance of 34.7% mAP), while the non-contrastive BYOL is 12.5% lower. Similarly, now SimCLR (contrastive) is better than SwAV (non-contrastive). Compared to AVA, Charades is a temporally less localized dataset containing activities that need to be recognized from a longer temporal range video, for which contrastive pre-training appears to be outperforming the non-contrastive variants.

method	pre-train	*linear protocol* K400	*finetuning accuracy* UCF101	AVA (mAP)	Charades (mAP)	SSv2
supervised	*scratch*	**74.7**	68.8	11.7	7.4	48.8
supervised	K400-240K	-	**94.8**	22.2	**34.7**	52.8
SimCLR	K400-240K	62.0 (−12.7)	87.9 (−6.9)	17.6 (−4.6)	11.4 (−23.3)	52.0 (−0.8)
SwAV	K400-240K	62.7 (−11.5)	89.4 (−5.4)	18.2 (−4.0)	10.7 (−24.0)	51.7 (−1.1)
BYOL	K400-240K	68.3 (−6.4)	93.8 (−1.0)	**23.4** (+1.2)	21.0 (−13.7)	**55.8** (+3.0)
MoCo	K400-240K	67.3 (−7.4)	92.8 (−2.0)	20.3 (−1.9)	33.5 (−1.2)	54.4 (+1.8)
MoCo	IG-Curated-1M	69.9 (−4.8)	**94.4** (−0.4)	20.4 (−1.8)	**34.9** (+0.2)	54.5 (+1.8)
MoCo	IG-Uncurated-1M	66.0 (−8.7)	92.9 (−2.1)	20.5 (−1.7)	31.3 (−3.4)	53.2 (+0.4)

Table 10. **Downstream benchmarks**: We use linear evaluation on K400 and finetuning accuracy on the other datasets. 200 epochs. ρ=3.

On **Something-Something v2** (SSv2 in Table 10), all the methods perform strong, with BYOL pre-training showing the largest gain of **+3**% over supervised pre-training on Kinetics (55.8% *vs*. 52.8% top-1 accuracy).

Pre-training sets: Kinetics *vs*. IG. Next, we experiment with pre-training on videos from the web. We first investigate **IG-Curated-1M** [24], which is a dataset that has been collected with hashtags that are similar to Kinetics labels. This data is a 1M subset of the original 65M introduced in [24]. Using this data (penultimate row in Table 10) can excels the performance of MoCo with K400 pre-training, which has a training set of 240K samples (roughly 4.2× smaller), and surprisingly even outperforms pre-training on K400 linear readout itself (69.9% *vs*. 67.3% accuracy).

Second, we ablate the effect of using uncurated videos, with **IG-Uncurated-1M** which are purely random videos taken from the web. On most downstream tasks performance shown in the last row of Table 10 is equal or only slightly lower than pre-training on K400. Specifically, MoCo changes by -1.3% on K400 (as expected), +0.1% on UCF, +0.2% on AVA, -2.2% on Charades and -1.2% on Something-Something v2. This is an encouraging result for unsupervised learning, as only ∼4.2×the number of videos but *random ones* are required to match the performance of supervised K400 pre-training on the UCF101 and AVA.

Overall, our results indicate that unsupervised pre-training can be a new paradigm for all of these downstream tasks, for which supervised pre-training is the de-facto standard to achieve best performance. Further, the large difference in performance for pre-training methodologies and objectives (*e.g*. contrastive/non-contrastive) revealed in the light of these benchmarks signals large room for future work.

4.7. Comparison to previous work

In a final experiment we take the best model from Table 9 and compare it with the state-of-the-art using the commonly used protocols on UCF101 and HMDB51 (across all 3 train/val splits) and K400. In Table 11 we show the results.

The strongest previous approaches are using multi-modal input, Vision "V", Audio "A", Text "T", to train a contrastive objective across modalities; XDC [3] performs DeepCluster [8] on (V+A), CVRL [72], GDT [69] and MMV [2] use an objective similar to SimCLR on (V), (V+A), and (V+A+T), with the latter training on a Audioset (AS) [23] and HowTo100M (HT) [60], and CoCLR [35] can be seen as a variant of MoCo on rgb and optical-flow input.

method	pre-train	backbone	param	T	mod	UCF	HMDB	K400
XDC [3]	K400	R(2+1)D-18	15.4M	32	V+A	84.2	47.1	
GDT [69]	K400	R(2+1)D-18	15.4M	32	V+A	89.3	60.0	
MMV [2]	AS+HT	S3D-G	9.1M	32	V+A+T	92.5	69.6	
SpeedNet [7]	K400	S3D-G	9.1M	64	V	81.1	48.8	
CoCLR [35]	K400	S3D-G	9.1M	32	V	87.9	54.6	
CoCLR [35]	K400	2×S3D-G	9.1M	32	V	90.6	62.9	
VTHCL [94]	K400	R-50	31.8M	8	V	82.1	49.2	37.8
CVRL [72]	K400	R-50	31.8M	32	V	92.2	66.7	66.1
ρ**BYOL**	K400	R-50	31.8M	8	V	**94.2**	**72.1**	**70.0**
ρ**BYOL**	K400	R-50	31.8M	16	V	**95.5**	**73.6**	**71.5**
ρ**BYOL**	K400	R(2+1)D-18	15.4M	32	V	**94.4**	**72.2**	
ρ**BYOL**	K400	S3D-G	9.1M	32	V	**96.3**	**75.0**	

Table 11. **Comparison with state-of-the-art**. "param" indicates the number of parameters, T inference frames, in the backbone. "V" is Vision, "A" is Audio, "T" Text modality. ρ**BYOL** is our best model trained with temporal persistency of ρ=4. We report fine-tuning accuracy on UCF/HMDB and linear accuracy on K400.

In comparisons, our best performing model ρ**BYOL**, which is BYOL trained with temporal persistency over ρ=4 clips, (*cf.* Tables 2 & 9), provides a substantial performance gain over the best published method [35]: **+5.7**% and **+12.1**% top-1 accuracy on UCF101 and HMDB51 (using identical backbone and pre-training data).

On K400 linear evaluation with the same data and R-50, Slow pathway [20] as backbone, our approach outperforms the previous best CVRL [72] by **+5.4**% accuracy.

5. Conclusion

This paper has studied four meta-methodologies for unsupervised learning from video. Our findings include that it is beneficial to sample positives with longer timespans between them, contrastive objectives are less influential than momentum encoders, and training duration, backbones, video augmentation and curation are all critical for good performance. Our resulting models which learn persistent features across augmented spacetime clips set a new state-of-the-art.

We observed that linear readout on Kinetics is a good indicator of the performance on other datasets and that unsupervised pre-training can compete with the supervised counterpart on several datasets, but there is room for improvement. We hope that our baselines will foster research and provide common ground for future comparisons.

References

[1] Pulkit Agrawal, João Carreira, and Jitendra Malik. Learning to see by moving. In *Proc. ICCV*, pages 37–45. IEEE, 2015. 2

[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. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 8

[3] Humam Alwassel, Dhruv Mahajan, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. *NeurIPS*, 2020. 2, 8

[4] Relja Arandjelović and Andrew Zisserman. Look, listen and learn. In *Proc. ICCV*, 2017. 2

[5] Relja Arandjelović and Andrew Zisserman. Objects that sound. In *Proc. ECCV*, 2018. 2

[6] Suzanna Becker. Learning temporally persistent hierarchical representations. In *NeurIPS*, 1997. 2

[7] 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 *Proc. CVPR*, 2020. 8

[8] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proc. ECCV*, 2018. 2, 8

[9] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 1, 2, 3, 4, 7, 14, 15

[10] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. 12, 15

[11] João Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 12, 15

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 1, 2, 3, 4, 6, 7, 14, 15

[13] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in Neural Information Processing Systems*, 33, 2020. 14

[14] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 6, 14, 15

[15] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013. 15

[16] Ali Diba, Vivek Sharma, Luc Van Gool, and Rainer Stiefelhagen. DynamoNet: Dynamic Action and Motion Network. In *Proc. ICCV*, 2019. 2

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

[18] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE PAMI*, 38(9):1734–1747, Sept 2016. 1, 2

[19] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. PySlowFast. `https://github.com/facebookresearch/slowfast`, 2020. 5

[20] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast Networks for Video Recognition. In *Proc. ICCV*, 2019. 2, 4, 5, 7, 8, 15, 16

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

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

[23] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. 8

[24] Deepti Ghadiyaram, Matt Feiszli, Du Tran, Xueting Yan, Heng Wang, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *Proc. CVPR*, 2019. 4, 5, 8, 12

[25] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. `https://github.com/facebookresearch/detectron`, 2018. 16

[26] R. B. Girshick. Fast R-CNN. In *Proc. ICCV*, 2015. 15

[27] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011. 14

[28] 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

[29] Ross Goroshin, Joan Bruna, Jonathan Tompson, David Eigen, and Yann LeCun. Unsupervised learning of spatiotemporally coherent metrics. In *Proc. ICCV*, 2015. 2

[30] 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:1706.02677*, 2017. 16

[31] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "Something Something" video database for learning and evaluating visual common sense. In *ICCV*, 2017. 1, 4, 7, 16

[32] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 1, 2, 3, 4, 7, 13, 14

[33] Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *Proc. CVPR*, 2018. 1, 4, 7, 15

[34] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Workshop on Large Scale Holistic Video Understanding, ICCV*, 2019. 2

[35] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *NeurIPS*, 2020. 2, 8

[36] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. CVPR*, 2020. 1, 2, 3, 4, 7, 14, 15

[37] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proc. ICCV*, 2017. 16

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 14

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 2, 4, 5, 7, 15

[40] Olivier J. Hénaff, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aäron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019. 2

[41] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *Proc. ICLR*, 2019. 2

[42] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*, 2015. 13, 14

[43] Phillip Isola, Daniel Zoran, Dilip Krishnan, and Edward H Adelson. Learning visual groups from co-occurrences in space and time. In *Proc. ICLR*, 2015. 2

[44] Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In *Proc. ICCV*, 2015. 2

[45] Simon Jenni, Givi Meishvili, and Paolo Favaro. Video representation learning by recognizing temporal transformations. *arXiv preprint arXiv:2007.10730*, 2020. 2

[46] Xu Ji, João F. Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proc. ICCV*, pages 9865–9874, 2019. 2

[47] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1, 2, 3, 4, 15

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

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

[50] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *Proc. ICCV*, pages 2556–2563, 2011. 2, 4, 16

[51] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequence. In *Proc. ICCV*, 2017. 2

[52] Ang Li, Meghana Thotakuri, David A Ross, João Carreira, Alexander Vostrikov, and Andrew Zisserman. The ava-kinetics localized human actions video dataset. *arXiv preprint arXiv:2005.00214*, 2020. 7

[53] Tianhao Li and Limin Wang. Learning spatiotemporal features via video and text pair discrimination. *arXiv preprint arXiv:2001.05691*, 2020. 2

[54] Ji Lin, Chuang Gan, and Song Han. Temporal shift module for efficient video understanding. In *ICCV*, 2019. 16

[55] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proc. CVPR*, 2017. 16

[56] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proc. ECCV*, 2014. 16

[57] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv:1608.03983*, 2016. 14

[58] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *Proc. ICLR*, 2017. 2

[59] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016. 2

[60] 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 *Proc. CVPR*, 2019. 8

[61] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *Proc. ECCV*, 2016. 2

[62] Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In *Proc. ICML*, pages 737–744, 2009. 2

[63] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. *arXiv preprint arXiv:2004.12943*, 2020. 2

[64] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. ICML*, 2010. 14

[65] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proc. ECCV*, pages 69–84. Springer, 2016. 2

[66] Andrew Owens, Phillip Isola, Josh H. McDermott, Antonio Torralba, Edward H. Adelson, and William T. Freeman. Visually indicated sounds. In *Proc. CVPR*, pages 2405–2413, 2016. 2

[67] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *Proc. CVPR*, 2017. 2

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

[69] 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, 8

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

[71] Senthil Purushwalkam and Abhinav Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. *arXiv preprint arXiv:2007.13916*, 2020. 2

[72] 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. 2, 8

[73] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2016. 15

[74] Pierre H Richemond, Jean-Bastien Grill, Florent Altché, Corentin Tallec, Florian Strub, Andrew Brock, Samuel Smith, Soham De, Razvan Pascanu, Bilal Piot, et al. Byol works even without batch statistics. *arXiv preprint arXiv:2010.10241*, 2020. 13

[75] Pierre Sermanet et al. Time-contrastive networks: Self-supervised learning from video. In *Proc. Intl. Conf. on Robotics and Automation*, 2018. 2

[76] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 1, 4, 7, 16

[77] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 7, 15

[78] 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, 4, 16

[79] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *Proc. ICML*, 2015. 2

[80] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Contrastive bidirectional transformer for temporal representation learning. *arXiv preprint arXiv:1906.05743*, 2019. 2

[81] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, 2015. 7, 15

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

[83] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proc. CVPR*, 2018. 5

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

[85] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabelled video. In *Proc. CVPR*, 2016. 2

[86] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proc. ICCV*, 2015. 2

[87] Xiaolong Wang, Kaiming He, and Abhinav Gupta. Transitive invariance for self-supervised visual representation learning. In *Proc. ICCV*, 2017. 2

[88] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *Proc. CVPR*, 2019. 2

[89] Laurenz Wiskott and Terrence Sejnowski. Slow feature analysis: Unsupervised learning of invariances. In *Neural Computation*, 2002. 2, 5

[90] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. In *Proc. CVPR*, volume abs/1805.01978, 2018. 1, 2

[91] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. In *Proc. ECCV*, 2018. 5

[92] Weidi Xie. *Deep Neural Networks in Computer Vision and Biomedical Image Analysis*. PhD thesis, University of Oxford, 2017. 16

[93] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proc. CVPR*, 2019. 2

[94] Ceyuan Yang, Yinghao Xu, Bo Dai, and Bolei Zhou. Video representation learning with visual tempo consistency. *arXiv preprint arXiv:2006.15489*, 2020. 2, 8

[95] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017. 13, 14

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

[97] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proc. ICCV*, 2019. 2