Benchmarking self-supervised video representation learning

Anonymous ICCV submission

Paper ID 6452

Abstract

Self-supervised learning is an effective way for label-free model pre-training, especially in the video domain where labeling is expensive. Existing self-supervised works in the video domain use varying experimental setups to demonstrate their effectiveness and comparison across approaches becomes challenging with no standard benchmark. In this work, we first provide a benchmark that enables a comparison of existing approaches on the same ground. Next, we study five different aspects of self-supervised learning important for videos; 1) dataset size, 2) complexity, 3) data distribution, 4) data noise, and, 5) feature analysis. To facilitate this study, we focus on seven different methods along with seven different network architectures and perform an extensive set of experiments on 5 different datasets with an evaluation of two different down-stream tasks. We present several interesting insights from this study which span across different properties of pretraining and target datasets, pretext-tasks, and model architectures among others. We further put some of these insights to the real test and propose an approach that requires a limited amount of training data and outperforms existing state-of-the-art approaches which use 10x pretraining data. We believe this work will pave the way for researchers to a better understanding of self-supervised pretext tasks in video representation learning.

1. Introduction

Deep learning models require large amount of labeled data for their training. Obtaining annotations at large-scale needs a lot of effort and it becomes even more challenging as we shift from image to video domain. There are several interesting directions focusing on this issue such as domain adaptation [61], knowledge distillation [17], semi-supervised learning [64], self-supervision [26] and weakly-supervised learning [47], which attempts to rely on the knowledge learned from existing source datasets and transfer it to new target datasets with minimal labels. Among these approaches, self-supervised learning use pretext task as supervisory signal and does not require any labels on source datasets which makes it more favorable.

In recent years, we have seen a great progress in self-supervised learning (SSL) in video domain [62, 27, 10, 58, 41, 8]. More recently, the focus is more towards context-based learning which involves modifying input data such that to derive a classification [60, 11, 62, 27], reconstruction [10, 8] or generative [56, 49, 21, 53, 38] signal which can be used as a learning objective. The main focus of these works is designing a pretext task which is computationally inexpensive and which provides strong supervisory signal such that the model learns meaningful *spatio-temporal* features.

Despite this great progress, it is non-trivial to compare these approaches against each other due to lack of standard protocols. These methods are evaluated under different conditions and there is no standard benchmark to evaluate the fair effectiveness of these methods. A recent study [52] attempts to take a step towards this direction, but it is mainly focused on down-stream learning, without exploring the self-supervision aspect which is one of the main goals in our study. In this work, we present a benchmark where important self-supervised pre-training parameters are kept consistent across methods for a fair comparison. With the help of this benchmark, we study several critical aspects which are important for self-supervised learning; *1) effect of pretraining dataset size, 2) task complexity, 3) generalization under distribution shift, 4) robustness against data noise, 5) properties of learned features.*

The proposed benchmark includes a large-scale assessment of context-based representative self-supervised methods for video representation learning. We analyze two different aspects: 1) *learning objective* which includes *contrastive* vs *non-contrastive*, and 2) *data transformation* that comprises of three categories namely, *spatial*, *temporal*, and *spatio-temporal*. We study seven different pretext tasks with seven different model architectures and perform our experiments on five different video action recognition datasets and evaluate these approaches on two different down-stream tasks, action recognition and video retrieval.

We observe some interesting insights in this benchmark.

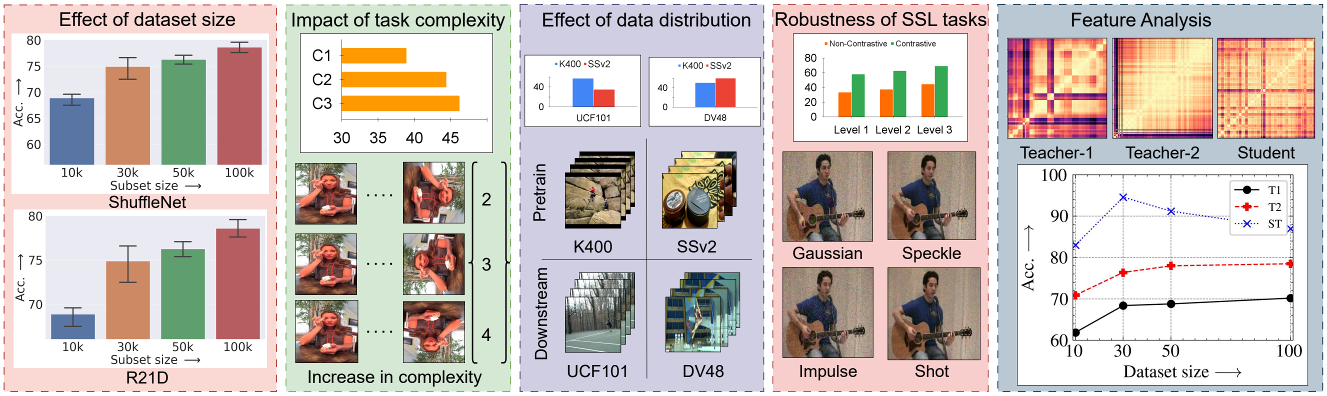


Figure 1: Overview of proposed benchmark. We study five different aspects in this benchmark study. Starting from left, 1) we show the analysis of *effect of dataset size vs training time*. As the dataset size increases, variation in performance decreases even with longer training time, 2) We show effect of task complexity. Bottom figure shows one use case of how complexity increases for RotNet task, and, top figure shows how the performance varies for R21D network, 3) With different data distribution shifts, third sub-figure shows the impact of *target* data distribution on the *source* data, 4) We look into another data distribution shift due to introduction of noise. We see how *non-contrastive* tasks are more robust than *contrastive* ones even with increasing level of severity of noise. Bottom part shows an example for each type of noise. Clips are provided in supplementary, and, 5) Finally, we further analyze whether the features learn complimentary information or not. In this sub-figure, we show that using different architectures as teachers, we can substantially improve the performance even in low-data regime.

Some of the key insights are; 1) Contrastive tasks are fast learners but are less robust against data noise, 2) there is no benefit of increasing dataset size for smaller models once model capacity is reached, 3) *temporal* based pretext tasks are more difficult to solve than *spatial* and *spatio-temporal*, 5) spatio-temporal task can solve the pretext task independent of data distribution shifts, and finally, 6) we empirically show that these pretext tasks learn complementary features across factors such as model architecture, dataset distributions, dataset size, and pretext task.

Our contributions are threefold:

- We present a benchmark for self-supervised video representation learning to compare different pretext tasks under a similar experimental setup.
- We perform extensive analysis on five important factors for self-supervised learning in videos; 1) dataset size, 2) task complexity, 3) distribution shift, 4) data noise, and, 5) feature analysis.
- Finally, we put some of our insights from this study to test and propose a simple approach which outperforms existing state-of-the-art methods on video action recognition with limited amount of pretraining data.

2. Related work

Self-supervised learning There are several works in the domain of self-supervised learning for video representation learning [26, 46]. These approaches can be grouped into two main categories on the basis of pretext task: 1) context-based [29, 59, 2, 16, 60, 51, 63, 11, 25, 58, 41, 8, 13, 20, 42], and 2) cross-modal [40, 44, 1]. Cross-modal approaches use multiple modalities such as audio, video, optical flow and camera positions, and rely on consistencies across these modalities. Context-based learning exploits data transformations to derive supervisory signals for training the model. Context-based pretraining tasks have evolved a lot in the past few years. Our work explores the domain of how much variation in learned representations under different transformations. In contrast to other approaches, context-based approaches exploit the spatial and temporal information independently by several transformations [36, 16, 62, 6, 60, 41, 58]. Recent works have started to transform the spatial and temporal domain together [29, 35, 51, 10, 8]. Incorporating multiple modalities improves performance, but, it's not available for all datasets, especially large-scale datasets. In this work, we restrict our focus to single-modality (RGB) approaches.

Self-supervised benchmarking There are some prior efforts focusing on benchmarking self-supervised learning in the image domain. In [18], the authors provide a detailed analysis of image-based self-supervised learning approaches and study how dataset size scaling affects the learned representations. Similarly in [30], the authors analyze how different model architectures play a role in visual self-supervised learning. In both these works, the authors did not focus on the importance of various pretext tasks themselves but only showed how certain pretext tasks can be improved. Therefore, their main focus was on down-stream tasks rather than pretext learning. We, on the other hand, study different pretext tasks and analyze how various aspects affect feature learning. Moreover, these works are focused on the image domain, whereas we focus on the video domain. In a recent work, [15], a study was performed to better understand unsupervised learning in the video domain, it basically explored the use of several pre-text tasks from the image domain and applied them to videos. We are not merely focusing on down-stream tasks and our attention is on the self-supervised aspect which includes factors such as data subset size, task complexity, dataset distribution, and noise robustness.

3. Self-supervised configurations

We first describe the pretext tasks used in our study along with their categorization. Then we discuss the details of this benchmark including network architectures, datasets, downstream tasks and evaluations.

3.1. Tasks categorization

We analyze two different aspects of video pretext tasks: 1) transformations applied to data, and 2) learning objective. Data transformations include, *spatial-based (S)*, *temporal-based (T)* and *spatio-temporal (ST)*. *Spatial* transformations include reshuffling of spatial patches, temporal consistent data augmentation, or rotation of images/patches. *Temporal* tasks involve permutation classification of frames/clip, order verification, clips sampling at different paces, or, contrastive learning from temporal triplets. *Spatio-temporal* tasks include those in which we modify both of these parameters simultaneously. This includes dilated sampling and simultaneous frame reconstruction, shuffling spatial and temporal domains, or, speed prediction, and contrastive visual features. Learning objectives can be either *contrastive* [9] or *non-contrastive* such as [53].

Following this categorization, we select at least two representative pretext tasks from each *transformation* category, one *contrastive* and one *non-contrastive*. We study the following pretext tasks in this study; RotNet (Rot) [27], Video Clip Order Prediction (VCOP) [62], Playback Rate Prediction (PRP) [10], Spatiotemporal Contrastive Video Representation Learning (CVRL) [41], Temporal Discriminative Learning (TDL) [58], Relative Speed Perception network (RSPNet) [8], and *V-MAE* [53]. In concise summary, 1) *RotNet* applies geometrical transformation on the data, 2) *VCOP* learns the representation by predicting the permutation order, 3) *PRP* has two branches, discriminative and generative that concentrate on temporal and spatial aspect respectively, 4) *CVRL* learns to cluster the video of the same class with strong temporal coherent augmentations, 5) *TDL* works on temporal triplets and minimizes the gap between anchor and positive on the basis of visual content, 6) *RSP-Net* applies contrastive loss in both spatial and temporal domain, and, 7) *V-MAE* [53] mask tokens of the input video and it tries to reconstruct those missing patches using an encoder-decoder architecture. More details are provided in supplementary.

3.2. Benchmark details

Datasets: We experiment with two different dataset types, 1) where appearance is more important, and 2) where time is more important. For appearance based, we use Kinetics-400 [28], UCF101 [48], and HMDB51 [32], where appearance is more important (recognize activity with a single frame) than temporal aspect, and for temporal aspect, we use Something Something-V2 [19] and Diving48 [33], where temporal information plays a significant role (require few frames to recognize activity). More details are in the supplementary.

Spatio-temporal architectures We analyze three different network capacities, 1) small-capacity, 2) medium capacity, and 3) large-capacity. For small capacity, we study the following architectures; ShuffleNet V1 2.0X [65], SqueezeNet [24], and MobileNet [43]. For medium capacity we focus on conventional 3D architectures: C3D [54], R3D [22], and, R(2+1)D [55] (R21D); . And, for big-capacity architectures we study VideoSwin [34], which is a transformer-based model.

Downstream tasks We show results and analysis on two different downstream tasks - action recognition and clip retrieval. These two are the most prominent tasks in the field of self-supervised learning in videos.

Evaluation and analysis We use top-1 accuracy for action recognition which indicates whether the class prediction is correct or not. Clip retrieval calculates the *top-k* hits for nearest neighbor search, where $k = \{1, 5, 10, 20, 50\}$. For robustness performance, we calculate the relative robustness score (R_s) using original accuracy on clean test set (A_c) and perturbed accuracy on noisy test set(A_p) as $R_s = \frac{A_c - A_p}{A_c}$. We also provide qualitative feature analysis with the help of centered kernel alignment (CKA) maps [37]. CKA maps illustrate the model's hidden representations, finding characteristic block structures in models. There are two dominant properties of CKA maps: 1) *Feature similarity:* Lighter regions in map indicates more similar features between layers than darker regions. 2) *Grid patterns:* Two main patterns stand out, a staggering grid, which indicates models are capable of learning more, and, distinctive light/dark block patterns meaning network reached its saturation point.

4. Benchmark analysis

In this section, first, we perform some preliminary experiments to compare each pretext task under identical conditions. Then, we further perform analysis across the following five aspects in the next subsections.

Effect of pretraining dataset size: In self-supervised learning, a natural question to ask is whether dataset size plays any role in the performance of downstream tasks. It is important to study if the increase in the size of the pretraining dataset will proportionally reciprocate in performance improvement. Also, a general trend is to train models for a very long duration at the pre-training stage. We investigate if the longer duration actually impacts the gain in performance. We look across different stages of training for multiple architectures and across different pretext tasks.

Impact of task complexity: Some of the existing works show that increasing complexity leads to better representation learning, and if the complexity is decreased, the network will optimize to suboptimal solutions. We analyze this aspect in more detail with several tasks and different model architectures.

Effect of data distribution: Existing self-supervised methods perform evaluations on K400 and UCF101 datasets. Both these datasets fall into the same visual category with heavy appearance bias. However, we divert our attention towards datasets where the temporal dimension plays an important role such as SSv2 and Diving48.

Robustness of SSL tasks: In this aspect, we study the robustness qualities of SSL methods against data noise [23]. We analyze which factors play a key role in the robustness of these methods against such distribution shifts.

Feature analysis: Finally, we look into feature space and analyze whether the learned representations are complimentary in nature when models are trained under different protocols.

4.1. Preliminary Experiments

First, we perform some preliminary experiments to analyze different architecture backbones, clip length, and evaluation with *linear probing* vs *finetuning*, and, finally layout discussion on the evaluation of different pretext tasks under the same constraints.

Backbone architectures: Looking into smaller and medium capacity networks in Figure 2, ShuffleNet outperforms among smaller networks, whereas considering the trade-off between the number of trainable parameters and performance R21D performs better in medium network category. Among big capacity networks, we look into few recent end-to-end video-based transformer networks [4, 14, 7, 34], and Video Swin [34] outperforms other architectures by a margin of 1-3% on K400.

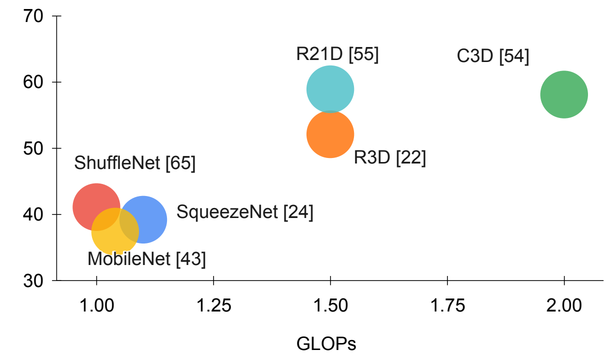


Figure 2: Variation in performance for different architectures. X-axis shows the relative floating point operations and Y-axis shows the Top-1 Accuracy.

	Non-Contrastive				Contrastive		
	Rot (S)	VCOP (T)	PRP (ST)	V-MAE (ST)	CVRL (S)	TDL (T)	RSP (ST)
Shuffle	16.6	40.8	21.9	-	62.3	12.4	**68.8**
R21D	41.2	51.5	46.2	76.2	61.2	31.7	**78.0**
Reported *	72.1	68.4	72.4	91.3	**94.4**	84.9	93.7

Table 1: Comparison across different pretext tasks pretrain on K400-50k subset and finetuned on UCF101 dataset against *reported* results in the original paper.

Clip length: Different pretext tasks take 16 or 32 frames as input clip length. We experimented with both 16 and 32 clips length and observe that 32 frames mostly provide better performance. However, to maintain consistency with most of the approaches and reduce computation costs, we use 16 frames in our experiments.

Linear probe vs finetuning: In the linear probe, we train only the linear layers attached for classification while freezing other network weights, whereas in finetuning the whole network is trained end-to-end. In our preliminary experiments we use Kinetics-400 for pretraining and UCF-101 as the target dataset. On several pretext tasks, we observe an average drop of 25% (ShuffleNet) and 40% (R21D) in performance when comparing linear probe with finetuning. However, we do not usually observe this significant drop when both the pretraining and target datasets are the same [46]. It indicates that *finetuning is important for the model to adapt to downstream dataset* in case it is different. Therefore, some of the existing works [52] rely on finetuning when the source and target datasets are different. Since we are interested in cross-dataset learning, we perform finetuning on all our downstream datasets.

Pretext tasks evaluation: A comparison of pretext tasks

	Non-Contrastive			Contrastive		
Subset	Rot	VCOP	PRP	CVRL	TDL	RSPNet
10k	37.6	46.3	17.5	55.9	31.1	70.9
30k	36.2	50.4	42.7	56.9	30.9	76.4
50k	41.2	51.5	46.2	61.2	30.2	78.0

Table 2: Evaluation of different pretext tasks on different subset size on R21D network.

on two different backbones is shown in Table 1. We observe that most of the *contrastive* tasks outperform *non-contrastive* tasks when they are trained under different constraints (row 3). However, that is not the case when we compare them under the same constraints (row 1-2). Similarly, *spatial* and *spatio-temporal* tasks have a similar performance from reported results. However, *spatio-temporal* pretext tasks outperform spatial ones by a large margin when we keep pre-training constraints similar. This supports our hypothesis that it is important to experiment under similar constraints for a fair evaluation of different approaches.

4.2. Effect of dataset-size

We first analyze the effects of pre-training data size variation. The network trains on four subsets of the K400 dataset: 10,000 (10k), 30,000 (30k), 50,000 (50k), and 100,000 (100k). The number of videos per class is the same. The smaller pre-training dataset is a subset of the bigger pre-training dataset size (i.e. $10k \subset 30k$ and so on). We look into three aspects regarding *dependence on pre-train subset size:* a) behavior of different pretext tasks with the increase in pre-train dataset subset, b) performance across the different capacity of backbones, and, c) the effect of training time across different pretext tasks.

Observations: From Table 2, we observe that apart from TDL each pretext task performance improves with an increase in subset size. If we look into specific pretext task transformation category (Table 2), the most gain with an increase in data is for *spatio-temporal* tasks (13%), whereas the least gain is for *temporal* pretext tasks (3%). Looking across different architectures in Figure 3, there's a minimal gain for R21D and ShuffleNet beyond increasing dataset size from 30k subset against VideoSwin which improves with an increase in dataset size which relates to similar behavior like image models discussed in [18]. Analyzing effect of duration of training across different pretext tasks, in Table 3, the performance gain is minimal ($<$1.5%) after training for more than 100 epochs. Comparing contrastive and non-contrastive approaches, the gain in contrastive based approaches is on average 1% compared to 5% for non-contrastive tasks beyond *100 epochs* of training.

Inference: (i) *Spatio-temporal pretext tasks improve most with increment in dataset size and are most dependent on it than others since it involves transformation along both axes: appearance (spatial) and motion (temporal).* (ii) *Benefit of more training data reaches its limitation based on model capacity. Smaller networks saturate according to their learning capability.* (iii) *Contrastive tasks are fast learners against non-contrastive and reach their potential in a relatively shorter duration of training.*

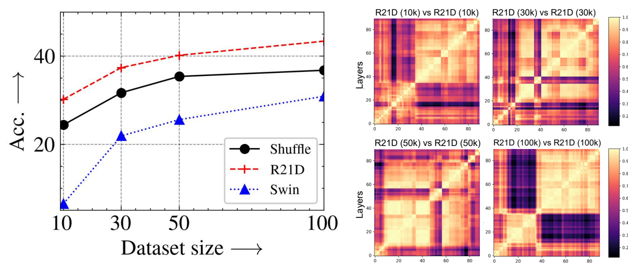


Figure 3: Left: dataset subset performance for three different architectures on RSPNet pretext task (x-axis: subset size, y-axis: Top-1 Accuracy). Here, 10 means 10k dataset subset, 30 means 30k and so on. Right: CKA maps for RSPNet on different subsets with R21D backbone.

	Non-Contrastive			Contrastive		
Epochs	Rot	VCOP	PRP	CVRL	TDL	RSPNet
50	35.4	52.2	24.1	55.7	32.1	75.0
100	37.3	52.3	34.8	58.5	31.3	76.1
150	40.7	51.3	46.7	60.2	31.5	76.5
200	40.9	52.8	45.0	60.5	30.2	77.4

Table 3: Performance of different pretext tasks on R21D over the training with 50k pre-training subset size.

TC↓	S	T	ST
C1	20.1/48.3	41.6/**56.8**	**24.2**/38.9
C2	**20.2/58.3**	**41.8**/54.8	18.1/44.4
C3	16.6/41.2	40.6/55.6	21.9/**46.2**

Table 4: Complexity Variation. TC: Task complexity. Results are shown on UCF101 with ShuffleNet/R21D backbone.

4.3. Impact of change in task complexity

Next, we study the effect of task complexity. In this aspect, we analyze only non-contrastive tasks as it is non-trivial to define task complexity for contrastive-based approaches. We analyze three different complexities (C1, C2, C3) for each task. The variation in complexity for each task is briefly discussed as follows: a) *RotNet*: vary the number of rotations between 2 to 4, b) *VCOP*: increase the number of shuffle clips from 3 to 5, and, c) *PRP*: modify the dilation sampling rates from 2 to 4 classes. We investigate the following aspects here: a) does increase in complexity means better spatio-temporal features learned at pre-training stage? b) does the capacity of architecture plays any role?

Observations: From Table 4, comparing across rows we observe ShuffleNet performance doesn't improve much or degrade significantly if the complexity of the task is increased. CKA maps show the structure transforms from staggering grids to a multi-block pattern indicating saturation with an increase in complexity. In between different categories of transformation, performance improves with complexity for the bigger model in the case of the *spatio-temporal* task. Between ShuffleNet and R21D, R21D gives staggering grids against dark block patterns for ShuffleNet which shows the model can still learn better features. CKA maps are provided in the supplementary.

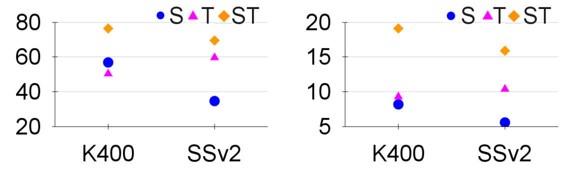
Inference: (i) *Increase in pretext task complexity doesn't always reciprocate to better spatio-temporal feature learning. It is dependent on the pretext task and also the model capacity.* (ii) *If higher complexity improves features learning, the model should also have the capacity, otherwise the task will be too difficult for the model to learn meaningful representations.*

4.4. Effect of dataset distribution

Shifting our focus to datasets which have more hidden cues in the temporal aspect, we add pre-training on SSv2 and finetuning on Diving48 to our experiments. We answer the following questions in this section; a) does the categorization of pretext-task matter on *source (pre-training)* and *target (downstream)* datasets? b) what is the impact of *source* dataset when the pretext task focuses only on a single task either *spatial* or *temporal*?

Observations: Looking into Figure 4, we observe that *spatio-temporal* pretext task outperforms other pretext tasks on both *target* (downstream) datasets UCF101 and DV48 by a margin of 15-40% and 10-13% respectively whether the *source* datasets is K400 or SSv2. Comparing, spatial and temporal-based pretext tasks, we see that they are *majorly* dependent on *source* datasets. Looking at Figure 4, performance is better on both *target* datasets if *source* dataset has the same underlying properties as the pre-text task is trying to learn. Furthermore, the spatial task is more dependent on the *source* dataset, since the relative drop on both UCF101 and DV48 for CVRL is significant (40% and 30% respectively), when the source dataset is SSv2 against K400. However, in the case of the temporal task, the drop is 15% and 10% respectively when the source dataset is K400 against SSv2.

Inference: (i) *Spatio-temporal pretext task learns better features independent of source and target data distribution.* (ii) *Spatial and temporal pre-text tasks are better learners when source data distribution belongs to spatial and temporal respectively.* (iii) *Temporal pretext task prevails when target data is temporal, whereas, in the case of spatial, tasks are dependent upon source data distribution. Spatial pretext doesn't gain much information if source data is SSv2 (temporal) since motion plays a major role, but the temporal task still learns well from K400 (appearance).*



(a) UCF101 (b) DV48

Figure 4: Pretraining on K400 and SSv2 with 30k subset size, finetuning on UCF101/Diving48 using R21D network. Here, S, T, and ST mean spatial(CVRL), temporal(VCOP), and, spatio-temporal(RSPNet) respectively. X-axis shows *source* dataset and Y-axis shows Top-1 accuracy.

	Non-Contrastive			Contrastive			
	Rot	VCOP	PRP	CVRL	TDL	RSP	Avg.
R21D	10.7	19.0	70.1	78.4	26.7	68.8	45.6
Shuffle	28.3	28.4	22.8	51.9	43.5	28.6	33.9

Table 5: Analysis on the relative decrease in % performance across different pretext tasks on noisy UCF101 dataset. The performance is averaged over 4 noises.

4.5. Robustness of SSL tasks

Similar to OOD datasets, introducing noise also shifts the distribution of datasets. We evaluate models on different types of noises introduced in [45] with different severity levels on UCF101 test dataset. Specifically, we probe into four different types of appearance-based noises: Gaussian, Shot, Impulse and Speckle [23]. Here we look into following aspects: a) how robust different categorization of pretext tasks are? b) is the network's architecture dependent on the noise in the dataset? In the main paper, we only discuss one severity level and have provided detailed analysis of multiple severity levels in the supplementary.

Observations: From Table 5, we observe that the relative drop in performance for contrastive tasks is more than non-contrastive tasks for both R21D and ShuffleNet backbone. The most and least robust models are RotNet-R21D and PRP-R21D with 10.7% and 70.1% relative decrease. From Figure 5, we can observe looking across different *severity levels* for each type of noise ShuffleNet is more robust than R21D.

Inference: (i) *Contrastive approaches are less robust to noise when compared with non-contrastive approaches.* (ii) *Looking at the average robustness score, ShuffleNet turns out to be more robust than R21D despite being smaller in terms of the number of parameters.*

4.6. Feature analysis

We further analyze the learned features by these pretext tasks under different configurations. We specifically focus on understanding the complementary nature of these fea-

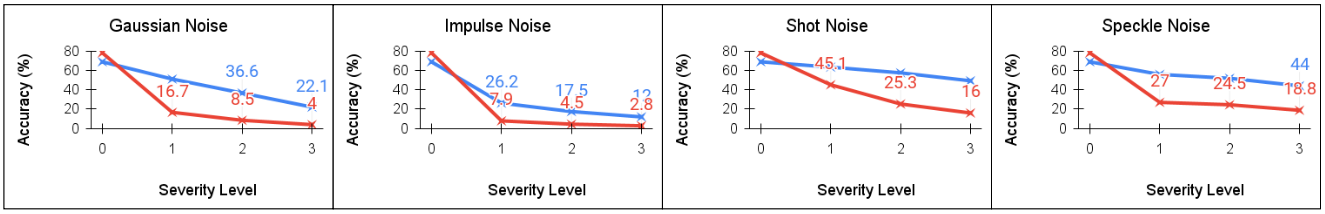


Figure 5: Performance with different types of noises. ShuffleNet and R21D scores are shown by blue and red lines respectively.

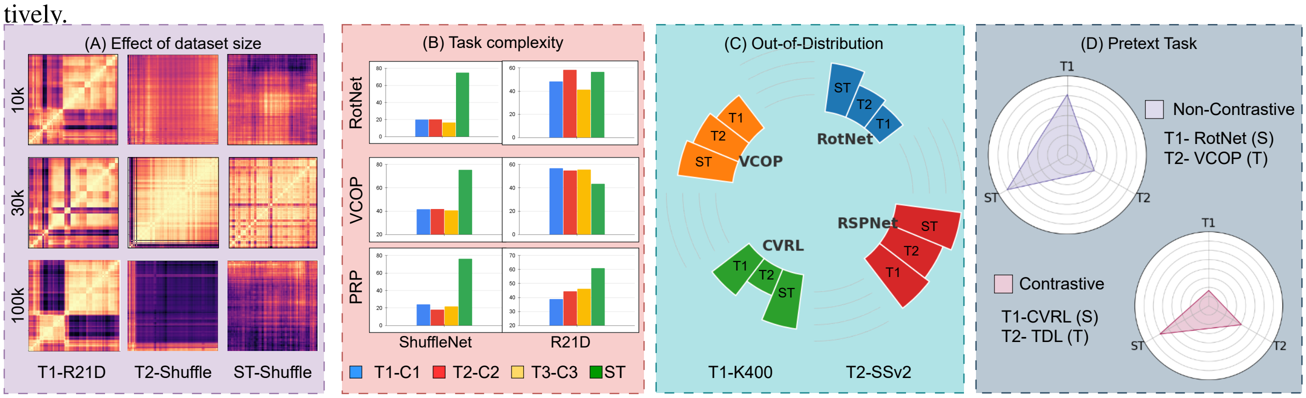
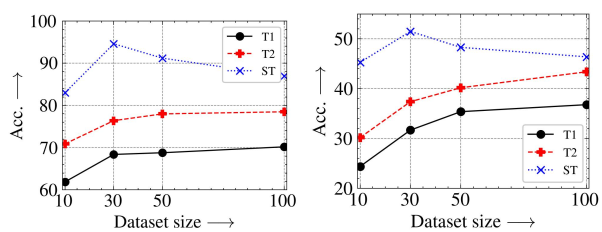


Figure 6: Feature analysis overview. Brief details for each setup: (A) *Effect of dataset size:* Teachers are different architectures for a single subset. (B) *Task Complexity:* Teachers are multiple complexities across the same task. (C) *Out-of-Distribution:* Models from different *source* datasets as teachers. (D) *Pretext Tasks:* Spatial and temporal task networks are teachers.



(a) UCF101

(b) HMDB51

Figure 7: KD using teachers trained on different subset sizes on RSPNet. Student: ShuffleNet UCF101/HMDB51. Here T1 is Teacher -1 (shufflenet) and T2-is teacher 2 (R21D).

tures. We employ knowledge distillation [12] as a tool to study this aspect. It is based on the idea that distilling knowledge from ensemble of teacher networks makes the student model stronger. We use our benchmark models as teachers in different combinations to analyze whether student learns orthogonal information on four different axes: 1) different architectures as teacher within a *dataset size*, 2) teachers with different complexities in a pretext task, 3) models from multiple *source* datasets, and, 4) same architecture as teachers from multiple pretext tasks. Figure 6 summarizes the *observations* for each aspect.

Observations: Although teacher network performance improves with subset, gain in complementary information reduces beyond 30k (Fig. 7). However, distillation does help in the reduction of training time with a significant improvement in performance which is evident from Fig. 6(a). Independent of the pretext tasks category smaller architecture learns complimentary information and outperforms the teacher whereas bigger architecture it's task-dependent. Irrespective of task category whether transformation-based or contrastive, each task learns corresponding features from both source datasets and outperforms the teacher. Student network outperforms standalone spatio-temporal network performance in both contrastive and non-contrastive domains.

Inference: (i) *Knowledge can be distilled from different architectures for a given subset size*, (ii) *Knowledge from different source datasets brings in complementary information*, and (iii) *Orthogonal features are learned across different categories of pretext tasks.*

5. Lessons learned

With all the analysis along studied axes, we learned a few lessons in-between these axes such as: (i) Contrastive tasks are fast learners but are also most susceptible to noise. (ii) An increase in dataset size or complexity does not help smaller models in learning better spatio-temporal features

Approach	NxW/H	Backbone	Dataset	UCF101
Generative				
VIMPAC [50]	10x256	ViT-L	HTM	92.7
VideoMAE [53]	16x224	ViT-B	K400	91.3
VideoMAE † [53]	16x112	R21D-18	K400	76.2
Context				
PacePred [60]	16x112	R21D-18	K400	77.1
TempTrans [25]	16x112	R3D-18	K400	79.3
STS [57]	16x112	R21D-18	K400	77.8
VideoMoCo [38]	16x112	R21D-18	K400	78.7
RSPNet [8]	16x112	R21D-18	K400	81.1
TaCo [5]	16x224	R21D-18	K400	81.8
TCLR[11]	16x112	R21D-18	K400	88.2
CVRL† [41]	32x224	R21D-18	K400	92.9
TransRank [13]	16x112	R21D-18	K200	87.8
Multi-Modal				
AVTS [31]	25x224	I3D	K400	83.7
GDT [39]	32x112	R21D	IG65M	95.2
XDC [3]	32x224	R21D	K400	84.2
Ours *	16x112	R21D-18	K400-30k	**97.3**

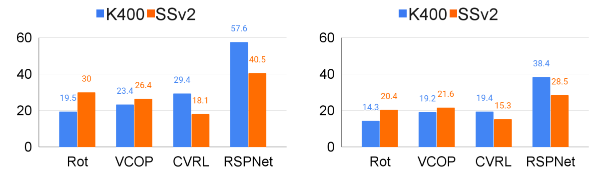
Table 6: Comparison with previous approaches pre-trained on K400 full set. Ours (* best performing) is RSPNet pre-trained on 30k subset of K400. † modified backbone.

but these features are more robust to noise. (iii) Temporal tasks are relatively more difficult to learn since looking at the correlation between time of training, increase in dataset size, and complexity, the performance gain is minimal in each of this axis. It means this category of tasks is actually difficult to solve. (iv) Spatio-temporal pretext tasks improve with the increase in complexity and dataset size (if model permits), and their behavior to learn better spatio-temporal features is independent of data distribution.

Using these lessons, we further do more analysis in feature space. From there, we observe within an axis of comparison how models learn orthogonal information. Based on those observations, we analyze if we can push the performance for downstream tasks. We look into two downstream tasks: action classification and clip retrieval.

Action Classification For this task, the model is fine-tuned end-to-end on downstream datasets, on UCF101 and HMDB51. In Table 6, we compare our best-performing model with other previous state-of-the-art approaches. ***Observations:*** With only 30k videos compared to 200k+ videos used by other pretext tasks, we show that our model outperforms by a good margin on UCF101 against single and multi-modal approaches. We got competitive results on HMDB51 with a score of 51.5%.

Clip retrieval For this downstream task, we generate the feature vectors using pretraining weights. The nearest neighbor is found by measuring the cosine distance between test and train feature vectors. We show analysis on UCF101 and HMDB51, with different source data distributions, K400 and SSv2. ***Observations:*** Spatio-temporal task still outperform other categories independent of *source* data distribution similar to what we observe earlier. Contrastive learns better *appearance* features during the pre-training stage given both downstream datasets are *appearance* based. Temporal tasks have almost similar performance pre-trained on either of the *source* datasets, which shows even with an appearance-based dataset as a pre-train dataset, the task is not focusing much on spatial features.



(a) UCF101 (b) HMDB51

Figure 8: Top@5 Clip Retrieval - R21D on a) UCF101 and b) HMDB51, pre-trained on K400 and SSv2 - 30k subset.

Recommendations Looking into several factors, here we provide some recommendations to set up the recipe for self-supervised learning: 1) *Training speed:* If training time is a concern, contrastive tasks can help in reducing the pretraining time. The only downside is, they could be less robust against data noise. 2) *Data distribution:* It is always better to use a spatio-temporal pretext task irrespective of the data distribution. However, if that is not an option, pretext task should always be aligned with the nature of pretraining dataset. 3) *Model capacity:* If model capacity is limited, there is no benefit of increasing pretraining dataset size and using complex pretext tasks. 4) *Robustness:* If best performance is the goal we should use a bigger model, otherwise if performance needs to be maintained in noisy data even allowing low performance then a smaller capacity model is preferable. 5) *Performance:* Pretext tasks learn complementary features across model architectures, pretraining datasets, pretext tasks, and tasks complexity, therefore, this complementary knowledge can be distilled to obtain strong spatio-temporal features.

6. Conclusion

In this study, we explore different parameters for self-supervised learning in video domain. We set a benchmark which provides an intuitive task categorization and enables a better comparison of different pretext tasks. Such an analysis has never been explored for video understanding to the best of our knowledge. We presented several interesting insights which will open up new directions for the research community. We also demonstrate the usefulness of some of these insights where we obtain state-of-the-art performance on video action recognition using merely 10% pretraining dataset when compared with existing methods. We believe this benchmark study will help the research community in better understanding of self-supervised learning in video domain. All the results and findings in this benchmark will be publicly released at https://thecodeeagle.github.io/webb/.

References

[1] Triantafyllos Afouras, Andrew Owens, Joon Son Chung, and Andrew Zisserman. Self-supervised learning of audio-visual objects from video. *ArXiv*, abs/2008.04237, 2020. 2

[2] Unaiza Ahsan, Rishi Madhok, and Irfan A. Essa. Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. *CoRR*, abs/1808.07507, 2018. 2

[3] Humam Alwassel, Dhruv Kumar Mahajan, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. *ArXiv*, abs/1911.12667, 2020. 8

[4] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. Vivit: A video vision transformer. *ArXiv*, abs/2103.15691, 2021. 4

[5] Yutong Bai, Haoqi Fan, Ishan Misra, Ganesh Venkatesh, Yongyi Lu, Yuyin Zhou, Qihang Yu, Vikas Chandra, and Alan Loddon Yuille. Can temporal information help with contrastive self-supervised learning? *ArXiv*, abs/2011.13046, 2020. 8

[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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[7] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *ArXiv*, abs/2102.05095, 2021. 4

[8] Peihao Chen, Deng Huang, Dongliang He, Xiang Long, Runhao Zeng, Shilei Wen, Mingkui Tan, and Chuang Gan. Rspnet: Relative speed perception for unsupervised video representation learning. In *AAAI*, 2021. 1, 2, 3, 8

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020. 3

[10] H. Cho, Tae-Hoon Kim, H. J. Chang, and Wonjun Hwang. Self-supervised spatio-temporal representation learning using variable playback speed prediction. *ArXiv*, abs/2003.02692, 2020. 1, 2, 3

[11] I. Dave, Rohit Gupta, M. N. Rizve, and M. Shah. Tclr: Temporal contrastive learning for video representation. *ArXiv*, abs/2101.07974, 2021. 1, 2, 8

[12] Shangchen Du, Shan You, Xiaojie Li, Jianlong Wu, Fei Wang, Chen Qian, and Changshui Zhang. Agree to disagree: Adaptive ensemble knowledge distillation in gradient space. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12345–12355. Curran Associates, Inc., 2020. 7

[13] Haodong Duan, Nanxuan Zhao, Kai Chen, and Dahua Lin. Transrank: Self-supervised video representation learning via ranking-based transformation recognition. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2990–3000, 2022. 2, 8

[14] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *ArXiv*, abs/2104.11227, 2021. 4

[15] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross B. Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3298–3308, 2021. 3

[16] Basura Fernando, Hakan Bilen, E. Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5729–5738, 2017. 2

[17] Jianping Gou, B. Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *ArXiv*, abs/2006.05525, 2021. 1

[18] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. *arXiv preprint arXiv:1905.01235*, 2019. 2, 5

[19] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fründ, Peter N. Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5843–5851, 2017. 3

[20] Sheng Guo, Zihua Xiong, Yujie Zhong, Limin Wang, Xiaobo Guo, Bing Han, and Weilin Huang. Cross-architecture self-supervised video representation learning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19248–19257, 2022. 2

[21] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *European Conference on Computer Vision*, 2020. 1

[22] K. Hara, H. Kataoka, and Y. Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 3154–3160, 2017. 3

[23] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ArXiv*, abs/1903.12261, 2019. 4, 6

[24] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡0.5mb model size, 2016. cite arxiv:1602.07360Comment: In ICLR Format. 3

[25] S. Jenni, Givi Meishvili, and P. Favaro. Video representation learning by recognizing temporal transformations. *ArXiv*, abs/2007.10730, 2020. 2, 8

[26] L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. 1, 2

[27] Longlong Jing, Xiaodong Yang, Jingen Liu, and Y. Tian. Self-supervised spatiotemporal feature learning via video

rotation prediction. *arXiv: Computer Vision and Pattern Recognition*, 2018. 1, 3

[28] W. Kay, J. Carreira, K. Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, F. Viola, T. Green, T. Back, A. Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *ArXiv*, abs/1705.06950, 2017. 3

[29] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8545–8552, Jul. 2019. 2

[30] A. Kolesnikov, X. Zhai, and L. Beyer. Revisiting self-supervised visual representation learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1920–1929, 2019. 2

[31] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018. 8

[32] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563, 2011. 3

[33] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *ECCV*, 2018. 3

[34] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3192–3201, 2022. 3, 4

[35] Dezhao Luo, Chang Liu, Y. Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video cloze procedure for self-supervised spatio-temporal learning. *ArXiv*, abs/2001.00294, 2020. 2

[36] I. Misra, C. L. Zitnick, and M. Hebert. Unsupervised learning using sequential verification for action recognition. *ArXiv*, abs/1603.08561, 2016. 2

[37] Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. *ArXiv*, abs/2010.15327, 2021. 3

[38] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11200–11209, 2021. 1, 8

[39] Mandela Patrick, Yuki M. Asano, Ruth Fong, João F. Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal self-supervision from generalized data transformations. *ArXiv*, abs/2003.04298, 2020. 8

[40] Senthil Purushwalkam and Abhinav Gupta. Pose from action: Unsupervised learning of pose features based on motion. *arXiv preprint arXiv:1609.05420*, 2016. 2

[41] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, H. Wang, Serge J. Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6960–6970, 2021. 1, 2, 3, 8

[42] Kanchana Ranasinghe, Muzammal Naseer, Salman Hameed Khan, Fahad Shahbaz Khan, and Michael S. Ryoo. Self-supervised video transformer. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2864–2874, 2021. 2

[43] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[44] N. Sayed, Biagio Brattoli, and Björn Ommer. Cross and learn: Cross-modal self-supervision. In *German Conference on Pattern Recognition (GCPR) (Oral)*, Stuttgart, Germany, 2018. 2

[45] Madeline Chantry Schiappa, Naman Biyani, Shruti Vyas, Hamid Palangi, Vibhav Vineet, and Yogesh Singh Rawat. Large-scale robustness analysis of video action recognition models. *ArXiv*, abs/2207.01398, 2022. 6

[46] Madeline C Schiappa, Yogesh S Rawat, and Mubarak Shah. Self-supervised learning for videos: A survey. *ACM Computing Surveys*. 2, 4

[47] Feifei Shao, Long Chen, Jian Shao, Wei Ji, Shaoning Xiao, Lu Ye, Yueting Zhuang, and Jun Xiao. Deep learning for weakly-supervised object detection and object localization: A survey. *ArXiv*, abs/2105.12694, 2021. 1

[48] K. Soomro, A. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *ArXiv*, abs/1212.0402, 2012. 3

[49] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 843–852, Lille, France, 07–09 Jul 2015. PMLR. 1

[50] Hao Tan, Jie Lei, Thomas Wolf, and Mohit Bansal. Vimpac: Video pre-training via masked token prediction and contrastive learning. *ArXiv*, abs/2106.11250, 2021. 8

[51] Li Tao, Xueting Wang, and Toshihiko Yamasaki. Self-supervised video representation learning using inter-intra contrastive framework. *arXiv preprint arXiv:2008.02531*, 2020. 2

[52] Fida Mohammad Thoker, Hazel Doughty, Piyush Bagad, and Cees G. M. Snoek. How severe is benchmark-sensitivity in video self-supervised learning? In *ECCV*, 2022. 1, 4

[53] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *ArXiv*, abs/2203.12602, 2022. 1, 3, 8

[54] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 4489–4497, USA, 2015. IEEE Computer Society. 3

[55] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *2018 IEEE/CVF Conference on Com-*

puter Vision and Pattern Recognition, pages 6450–6459, 2018. 3

[56] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 1

[57] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Wei Liu, and Yunhui Liu. Self-supervised video representation learning by uncovering spatio-temporal statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:3791–3806, 2022. 8

[58] Jinpeng Wang, Yiqi Lin, Andy Jinhua Ma, and Pong Chi Yuen. Self-supervised temporal discriminative learning for video representation learning. *ArXiv*, abs/2008.02129, 2020. 1, 2, 3

[59] X. Wang, K. He, and A. Gupta. Transitive invariance for self-supervised visual representation learning. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1338–1347, 2017. 2

[60] Jiangliu Watng, Jianbo Jiao, and Yunhui Liu. Self-supervised video representation learning by pace prediction. In *European Conference on Computer Vision*, 2020. 1, 2, 8

[61] Garrett Wilson and Diane Joyce Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11:1 – 46, 2020. 1

[62] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 3

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

[64] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *ArXiv*, abs/2103.00550, 2021. 1

[65] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3