Benchmarking Self-supervised Learning for Spatio-temporal Representations

Anonymous CVPR submission

Paper ID 3061

Abstract

Self-supervised learning has become the cornerstone in training deep neural networks to alleviate the requirement of a large number of labeled samples, especially for the video domain. Existing works in video domain use varying settings/backbones to show that one pretext is better than the other, but it's challenging with no standard benchmark. We present a benchmark with similar constraints for all pretext tasks and compare existing approaches on the same ground. In our study, with more than ***400+ experiments****, first we propose a new categorization for self-supervised pretext tasks. Then, we perform an in-depth study on different key factors including:* ***pre-training dataset size****,* ***task complexity****,* ***effect of out-of-distribution*** *and* ***noisy datasets****, which are important for self-supervised learning. Furthermore, we analyze the complementary nature of features learned under such scenarios. More specifically, we utilize knowledge distillation to study this aspect and demonstrate how this can be used to train better models. We observe that* ***models learn complementary knowledge*** *under different conditions and we integrate this knowledge in a model which shows* ***state-of-the-art performance*** *for activity recognition on UCF-101 dataset. Our work will pave the way for researchers for a better understanding of self-supervised pretext tasks in video representation learning.*

1. Introduction

Every year, an exorbitant amount of data is getting published in several domains. The main concern is lack of labels for each specific domain, especially when deep learning requires a lot of annotations. Obtaining annotations for videos is far more challenging than image. There are several research directions addressing this challenge including, domain adaptation [55], knowledge distillation [15], semi-supervised [58], self-supervised [24] and weakly-supervised [44] which attempts to rely on the knowledge learned from existing datasets and use it for new datasets with minimal labels. Among these approaches, self-supervised learning is one which use pretext task as supervisory signal instead of labels to train on large scale datasets which makes it more favorable.

We have seen a great progress with different approaches [8, 9, 25, 40, 52, 56] for self-supervised learning. More recently, the focus is more towards developing a new pretext tasks peculiarly on modifying input data such that to derive a classification [10, 25, 54, 56], reconstruction [8, 9] or generative signals [18, 37, 46, 49, 50] out of it. 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 developed methods are evaluated under different conditions and there is no standard benchmark to evaluate the fair effectiveness of these methods. There are some efforts focusing on image domain [16, 28], but there is no such effort in video domain to best of our knowledge. A recent study [48] 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 the main goal of our study, among many other critical aspects. In this work, we present a benchmark where important pre-training parameters are kept consistent across pretext tasks for a fair comparison. We extend our analysis to several other aspects that are still left unanswered among researchers, particularly in the self-supervised video domain.

There are several critical questions which are important for self-supervised learning to measure the effectiveness of different pretext tasks: 1) does scaling the dataset size directly reciprocate in the performance gain? 2) how does task complexity affect the quality of learned features? 3) how well the features generalize to dataset which have different distribution? 4) what is the behavior of self-supervised pre-trained models with the introduction of noise? 5) does pre-training on different datasets helps in performance improvement on downstream datasets? and more importantly 6) does different pretext tasks learn complementary features? If so, can it be utilized together to learn better spatio-temporal features than model trained on standalone pretext task?

In this study, we attempt to find answers to some of these critical questions. Towards this, we perform a large-scale assessment on these queries pertaining to self-supervised learning on videos. To facilitate our study, first, we carefully select a set of representative pretext tasks such that they cover all the variations. To this end, we propose a new categorization for pretext tasks, specifically dividing them into three different categories, namely, spatial, temporal, and, spatio-temporal. We select two pretext tasks from each of these categories and study six different pretext tasks. We use five different video-based datasets to perform our experiments and evaluate these approaches on two different down-stream tasks, action recognition and video retrieval.

Based on our observations, we have summarized the following key findings: 1) scale of increase of subset size doesn't reciprocate to similar performance gain across each pretext task, 2) increase in task complexity doesn't always guarantee that network learning can be maximized to their full potential, 3) performance on out-of-distribution dataset is dependent on the type of pre-training dataset, 4) different pretext tasks have different level of robustness even among separate architectures when it comes to noisy datasets, and finally 6) we empirically show that these pretext tasks learn complementary features and this complementary knowledge can be integrated and transferred to a single model with the help of knowledge distillation to obtain a better model. Our contributions are three fold:

- We setup a benchmark for video representation learning to compare different pretext tasks under similar pre-training setup.
- We perform an extensive analysis on four important factors which are important in self-supervised learning for video domain.
- We show that knowledge distillation demonstrates promising future in the area of self-supervised learning. Our best performing/knowledge distilled model also beats the state-of-the art approaches on action recognition with a margin of **4.4%** on UCF101 and have comparable performance on HMDB51.

Section II walks through previous works in self-supervised area and recent developments in video domain. Section III talks about the training setup, selection of pretext tasks, datasets we use and how we select models for our experiments. Section IV delves deeper into the observations about how different parameters affect the representations and how representations evaluates on downstream tasks and datasets. Then, Section V talks about the use of knowledge distillation along different parameters. Lastly, we provide recommendations based on our key findings.

2. Related work

Self-supervised learning There are several works in the domain of self-supervised learning for video representation learning [24]. The three main categories on the basis of pretext tasks are: 1) Generation-based [18, 46, 50], 2) Context-based, [2, 8, 10, 14, 22, 27, 40, 47, 52–54, 57], and, 3) Cross Modal [1, 39, 42]. Generative pretext tasks generates or predicts future frames. The predicted frames are compared with ground truth frames. Cross-modal approach uses audio, video, optical flow and camera positions. Context-based pretraining tasks has evolved a lot since past few years. Our work explores in the domain of how much is the variation in learnt representations under different transformations. In contrast to other approaches, context-based approaches exploits the spatial and temporal information independently by several transformations [6, 14, 35, 40, 52, 54, 56]. Recent works have started to transform the spatial and temporal domain together [8, 9, 27, 33, 47]. Incorporating multiple modalities improves the performance, but, it's not available for all datasets, especially large-scale. In this work, we restrict to single-modality (RGB) approaches.

Self-supervised benchmarking There are some recent efforts focusing on benchmarking self-supervised learning for visual data. Previous work on images didn't specify a strong reason for the choice of selecting pretext tasks. [16] selected the pretext tasks on the basis of ease of implementation. The authors in [28] also didn't justify the choice of their self-supervised tasks. Whereas we gave a rationale reason in selection of each pretext tasks for our experiments. In a recent work [13], a study was performed to better understand unsupervised learning in video domain, it basically explored the use of several pre-text tasks from image domain and applied them to videos. We are not merely focusing on down-stream tasks and our attention is on self-supervised aspect which includes factors such as data subset-size, task complexity, and, noise robustness compared to [13] where the focus is on requirement of temporal clips, timespan, and, data augmentation. We report scores on small capacity networks [21, 41, 59] and a transformer [32]. Transformers has never been looked in self-supervised learning in video domain. We have also reported a detailed discussion on out-of-distribution datasets and noise robustness.

Knowledge distillation Knowledge distillation (KD) is a technique where information transfers from a bigger model, or from an ensemble of models, to a relatively smaller and compact model without drop in accuracy. It's made up of three core elements: knowledge, distillation algorithm, and teacher–student architecture. It has been mainly explored in supervised settings [19, 34] that too specifically on image-based networks, with few research on limited label settings [23]. However, knowledge distillation in self-supervised learning remains a vast uncharted territory.

3. Self-supervised configurations

We first describe our proposed categorization. Then, we briefly look into how different pretext tasks formalized their problem, followed by network architectures, datasets and downstream tasks used in our study.

3.1. Proposed categorization

We propose a new set of categorization of video pretext tasks on the basis of transformations applied to data during pre-training stage: *spatial-based*, *temporal-based* and *spatiotemporal*. *Spatial-based* transformations includes random crops, reshuffling of spatial patches, temporal consistent data augmentation or rotation of images/patches. *Temporal-based* tasks involves permutation classification of frames/clip, order verification, clips sampling at different paces, or, contrastive learning from temporal triplets. *Spatiotemporal-based* tasks includes those in which we modify both of these parameters simultaneously. Like dilated sampling and frame reconstruction together, shuffling spatial and temporal domain, or, speed prediction and contrastive visual features.

According to our categorization, we select two pretext task from each category, one *contrastive* and one *non-contrastive*, that makes it six different pre-text tasks in total. a) Non-contrastive: RotNet [25], Video Clip Order Prediction (VCOP) [56] and Playback Rate Prediction (PRP) [9], and b) Contrastive: Spatiotemporal Contrastive Video Representation Learning (CVRL) [40], Temporal Discriminative Learning (TDL) [52], and, Relative Speed Perception netowrk (RSPNet) [8]. We will explain each pretext task briefly: 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 concentrates on temporal and spatial aspect respectively, 4) *CVRL* learns to cluster the video of 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, and, 6) *RSPNet* applies contrastive loss in both spatial and temporal domain. More details in supplementary.

3.2. Benchmark details

Datasets: Our work uses following datasets, Kinetics-400 [26], UCF101 [45], and, HMDB51 [30], where appearance is more important (recognize activity with a single frame) than temporal aspect, and Something Something-V2 [17] and Diving48 [31], where temporal information plays a significant role (require few frames to recognize activity). More details are provided in the supplementary.

Spatio-temporal architectures For our analysis, we use three different capacity of networks: 1) Small-capacity: utilizes point-wise group convolutions (ShuffleNet V1 2.0X [59]), reduction in filter size (SqueezeNet [21]) and depthwise convolution (MobileNet [41]); 2) Medium-capacity: Conventional 3D architectures: C3D, R3D, and, R(2+1)D (R21D); 3) Big-capacity: Transformer-based architecture: VideoSwin [32] backbone.

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 on which all the approaches are evaluated. Action recognition is evaluated using Top-1 accuracy, whether the class prediction is accurate or not. Clip retrieval calculates the *top-k* hits for nearest neighbor search, where $k = \{1, 5, 10, 20, 50\}$.

4. Representation analysis

In this section, first, we perform some preliminary experiments to put down the reasoning for selection of backbone architectures, and, pretext tasks. It also helps in narrowing down the number of experiments. We perform analysis in the first section of our study examining the video representation learning across following axes: (i) *Dataset pretraining size:* Self-supervised learning is dependent upon the amount of available videos while pretraining. A natural question arises, is it necessary that increasing the dataset size will replicate the improvement in performance? Also, models perform pre-training for a very long duration. We investigate the performance of networks at different stages of training for multiple architectures and across different pretext tasks. *(ii) Task complexity:* Most of the works show that increasing complexity leads to better representation learning, and, if the complexity is decreased, network will optimize to suboptimal solutions. Is it really the case? *(iii) Out-of-Distribution (OOD) analysis:* Self-supervised approaches mostly perform evaluations on K400 and UCF101 datasets. On the basis of appearance bias, they can be categorized into same category. However, we divert our attention towards datasets where temporal dimension plays an important role such as SSv2 and Diving48. *(iv) Noise Robustness:* In this direction, we apply noise on UCF101 testing dataset, and, then compare the performance of finetuned models. Each analysis is divided into three parts: 1) experimental setup, 2) quantitative and qualitative observations, and, 3) inference from that analysis.

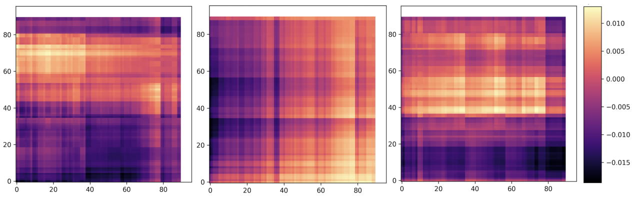


Figure 1. CKA maps for layer representations: pre-training vs linear probe, finetune vs linear probe, and, pretrain vs finetune of R21D network on VCOP 50k subset (Left to right).

4.1. Preliminary Experiments

We perform a small subset of preliminary experiments to explore different architecture backbones, clip length and evaluation with *linear probing* vs *finetuning*, and, discuss briefly about qualitative observations.

Backbone architectures: Looking into smaller and medium capacity networks, ShuffleNet and R21D outperforms other networks in their respective categories. A comparison table is provided in supplementary for different pretext tasks. Therefore, we perform the remaining experiments in this study using R21D and/or ShuffleNet. For big capacity networks, we compare performance of few recent end-to-end video-based transformer networks [4, 7, 12, 32]. Since, training transformer architectures is computationally expensive, we select [32] on the basis of the best performance on K400.

Clip length: Different pretext tasks takes 16 or 32 frames into account. We experimented with both 16 and 32 clips length and observe that 32 frames mostly provides better performance. However, to maintain the consistency with most of the approaches and reduce computation cost, we use 16 frames in our experiments.

Linear probing vs finetuning: In linear probing, we train only the linear layers attached for classification and other weights are frozen, whereas, in finetuning the whole network is trained. Evaluating performance on three different pretext tasks (VCOP, RotNet and PRP) on *linear probing*, there's an average drop of 25% in case of ShuffleNet and 40% in case of R21D network, when the pre-training and downstream datasets are different. In standard settings, existing works [48] also finetune when pre-training and downstream datasets are different. Therefore, in our study we perform finetuning on all our downstream datasets (UCF101, HMDB51 and Diving48) as they all are different from the pre-training datasets (K400 and SSv2).

Pretext tasks: Looking into Table 1, within non-contrastive tasks, VCOP outperforms others by a margin of 5-10% for R21D and almost 2 times in case of Shufflenet architecture. R21D is better than ShuffleNet. In contrastive, RSPNet outperforms all others. ShuffleNet outperforms R21D for CVRL. Amongst contrastive, pretext task focusing on temporal content performs worst compared to spatial and spatiotemporal.

	Non-Contrastive			Contrastive		
	RotNet	VCOP	PRP	CVRL	TDL	RSPNet
Shuffle	16.6	40.8	21.9	62.3	12.4	**68.8**
R21D	41.2	51.5	46.2	61.2	31.7	**78.0**

Table 1. Comparison across different pretext tasks pre-train on K400-50k subset and finetuned on UCF101 dataset.

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

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

Qualitative observations In our work, we discuss qualitative observations using centered kernel alignment (CKA) [36]. CKA maps illustrate model's hidden representations, finding characteristic block structures in models. These block structure indicates how underlying layers preserve and propagate the dominant principal component of their representations. Initially, we look into CKA maps of different pretext tasks. For pretext tasks belonging to the spatio-temporal category (PRP and RSPNet), both the networks produce identical CKA maps, indicating that learning on these tasks happens in a similar fashion, giving rise to similar layer representations. Similar trend is observed for *temporal-based* pretext tasks, where ShuffleNet depicts block type patterns for both VCOP and TDL, while R21D gives more staggering patterns for these instead.

We plot CKA maps for R21D network (Figure 1) to draw out the comparisons and similarities between linear probing and finetuning on VCOP. An excessively purple portion to the left side of the first plot indicates dissimilarity between the first few layers of a linear probe and finetuned network. The other two plots depict the comparison between a pretrained network and a linear probing/finetuned network respectively. In case of a linear probing, there are a few layers in the middle region which have similar layer representations to that of a pretrained network, while in case of finetuning, majority of the layers are different that indicates the network adapts to downstream datasets.

4.2. Pre-training dataset

We first analyze the effects of pre-training data size variation. The network trains on four subsets of K400 dataset: 10,000 (10k), 30,000 (30k), 50,000 (50k) and 100,000 (100k). The number of videos per class are same. Smaller pre-training dataset is a subset of bigger pre-training dataset size (i.e. $10k \subset 30k$ and so on). We try to answer three questions regarding dependence on pre-train subset size: a) Behavior of different pretext tasks, b) How various archi-

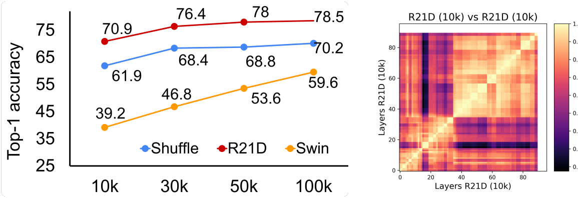


Figure 2. Left: dataset subset performance for three different architectures on RSPNet pretext task (x-axis: subset size). Right: CKA map for RSPNet for 10k subset with R21D backbone.

Epochs	10k	30k	50k	100k
50	59.1/66.8	66.3/71.1	68.1/75.0	68.9/77.2
100	60.3 /69.5	67.6/75.2	68.7/76.1	69.0/80.0
150	61.8/69.5	66.7/76.6	69.4/76.5	69.7/78.8
200	61.5/69.6	68.2/76.6	68.5/77.4	69.9/78.3

Table 3. RSPNet with different subset size on ShuffleNet/R21D.

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

Table 4. Performance of different pretext tasks on R21D with 50k pre-training subset size.

tecture backbone performs?, c) Effect of training time for different architectures and across different pretext tasks.

Quantitative observations: From Table 2, we observe that apart from TDL each pretext task performance improves with increase in subset size, with PRP having the most absolute gain of 28.7%. Looking into different architectures in Figure 2, there's 6-7% improvement in performance with increase in dataset size from 10k to 30k for all architectures. Increasing the subset size from 30k to 100k, shows minimal effect on R21D and ShuffleNet, whereas VideoSwin still improves by 12.8%. Looking into the effect of duration of training across different architectures for different subsets (Table 3), the performance gain is minimal (<1.5%) after training for more than 100 epochs. If we fix the subset size to 50k, apart from PRP, the average gain in performance is less than 2% for all other pretext tasks (Table 4).

Qualitative observations: We observe that with increase in *subset size* used for training from 10k to 100k, block patterns become more distinct for both ShuffleNet and R21D networks . The wider and clearer blocks corroborate with the saturation with increase in subset with minimal gain in performance. At 50 epochs, while comparing R21D CKA maps on 10k, there's a multi-block structure against 100k subset, where, the map shows a grid checkerboard structure. Block patterns relates to highly parameterization with respect to training dataset. With 10k, R21D (Fig.2) becomes relatively over-parameterized with respect to the training set. CKA maps depict how increasing the number of epochs keeping a given subset size fixed leads to development of distinct and darker block structures in the layer representations. Table 4 shows gain in performance reduces as the number of training runs increases from 150 to 200, indicating signs of saturation.

TC↓	RotNet	VCOP	PRP
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 5. Complexity Variation:TC: Task complexity. Results are shown on UCF101 with ShuffleNet/R21D backbone.

Inference: As a general rule, performance do increase with increase in subset size. However, the scale of increase of subset size doesn't reciprocate to gain in performance for each pretext task. Pretext tasks does saturate at certain subset size. Beyond certain point, if we compare the time taken with more data, training becomes less efficient. CKA maps also shows the block structure patterns becomes more distinct as the improvement is small when we move to larger subset. For contrastive tasks, they reach their potential with shorter duration of training as well.

4.3. Task complexity

Next, we study the effect of task complexity and we observe that at a certain point a task can become unsolvable or trivial. In this aspect, we analyze only non-contrastive tasks as it is non-trivial to define task complexity for contrastive based approaches. We look into three different complexities (C1, C2, C3 (Table 5)) for each task. The variation in complexity for each task is discussed as follows: **VCOP:** We increase the number of clips from 3 to 5. With these variations, the number of permutations increase from 6 to 120 which increases the memory as well as compute cost for VCOP. We reduce the batch-size by a factor of 2 as we go from 3-5 to limit the memory consumption. **RotNet:** We fluctuate the number of times clips are rotated along the spatial axis for three cases: 2, 3 and 4 rotations. The videos are rotated in multiple of 90 degrees. **PRP:** We investigate different sampling rates for this pretext task. The dilation sampling rates are 1, 2, 4 and 8. Depending upon the types of sample steps in the list, we have a class label assigned to each sample step. The sample rate of clip is classified ranging from 2 classes for $\{1, 2\}$ to 4 classes which includes all sampling rates $\{1, 2, 4, 8\}$.

Quantitative observations: From Table 5, we see increasing the number of rotations in ***RotNet*** from three to four makes the task too complex for the network and there's a sharp decrease in performance (more than 10% in case of R21D). In case of ***VCOP***, we see that 3 clips permutation leads to better solution than other settings by a margin 2-3%. For ***PRP***, small capacity networks are unable to generalize well with increase in complexity as compared to bigger capacity network.

Qualitative observations: In case of **RotNet**, both networks show staggering grids for complexity with 3 rotations. As we increase the complexity, multi-block pattern gets more prominent which indicates the saturation in performance. For **PRP**, ShuffleNet has an overall lower performance in comparison to R21D, and, hence we see a multi-block patterns for different complexities, giving the darkest pattern when increased from 3 to 4. On the other hand, R21D depicts staggering grids for both these complexities.
Inference: We derive the following two conclusions from this experiment: (i) training with a more complex problem doesn't always lead to optimal solutions.(ii) CKA maps show that increasing complexity of the problem is indicative of saturation in performance as block structures begin to emerge.

4.4. Out-of-distribution dataset

Shifting our focus to datasets which has more hidden cues in the temporal aspect, we analyze pre-training on SSv2 and finetuning on Diving48. We answer the question; how pre-training on datasets, where appearance (K400) is more important vs temporal (SSv2), affects the performance on target dataset. We show results on R21D network pre-trained on 30k subset for 200 epochs and finetune for 100 epochs.
Quantitative observations: Looking into Table 6, VCOP and RotNet, outperforms the pre-training of K400 with SSv2 by a margin of 6-9% on UCF101, 3-6% on Diving48 dataset. In case of CVRL and RSPNet, pre-training with K400 than SSv2 outperforms on both UCF101 and Diving48. The best performance on UCF101 is from RSPNet pre-trained on K400, and, on Diving48, it's RotNet pre-trained on SSv2.
Qualitative observations: We discuss the CKA maps for UCF101. R21D pretrained on K400 shows a semi-block structure for VCOP, indicating near-saturation condition of the network on this pretext task. It shows a more prominent grid-based structure on CVRL and RSPNet instead. These observations corroborate the quantitative results, where pre-training on K400 for both CVRL and RSPNet gives better performance.
Inference: Among non-contrastive tasks (VCOP and Rot), better features are learnt with SSv2 as pre-training dataset, whereas, the scenario is reversed for contrastive tasks. Looking at the best performance, pre-training on K400 performs better on UCF101 than SSv2 and vice versa for Diving48. Thus, we can deduce that pre-training on appearance vs temporal based dataset matters.

4.5. Noise robustness

Similar to OOD datasets, introducing noise also shifts the distribution of datasets. We evaluate models on different types of noises introduced in [43] with different severity levels on UCF101 test dataset. Specifically, we probe into four different types of appearance-based noises [20]: Gaussian, Shot, Impulse and Speckle. We show the average performance with severity level 1 in the main paper for each type noise. We have extended this analysis across different severity levels and another datasets in the supplementary.

		VCOP	Rot	CVRL	RSPNet
UCF101	K400	50.4	36.2	56.9	76.4
	SSv2	59.7	42.5	34.7	69.5
Diving48	K400	9.3	14.8	8.2	19.1
	SSv2	10.4	21.3	5.6	15.9

Table 6. Pretraining on K400 and SSv2 with 30k subset size, finetuning on UCF101/Diving48 using R21D network.

		Rot	VCOP	PRP	CVRL	TDL	RSP
R21D	Abs.	36.8	41.7	13.8	13.2	23.3	24.3
	Rel.	10.7	19.0	70.1	78.4	26.7	68.8
Shuffle	Abs.	11.9	29.2	16.9	30.0	7.0	49.1
	Rel.	28.3	28.4	22.8	51.9	43.5	28.6

Table 7. Analysis on noise across different pretext tasks on UCF101 dataset. The performance is averaged over 4 noises. Second row for each network shows relative percentage decrease in performance. Abs. means absolute decrease and Rel. mean relative decrease in %.

Quantitative observations: From Table 7, looking at the relative decrease in performance for R21D backbone, *spatio-temporal* based pretext task is least robust (↓ 69.5), whereas, the scenario is opposite for ShuffleNet backbone (↓ 25.7). Most robust model is RotNet with R21D backbone with 10.7% relative decrease, and, the least robust model is PRP with R21D backbone with 70.1% relative decrease in performance. We have shown TSNE plots in supplementary for *qualitative* analysis.
Inference: (i) Between contrastive and non-contrastive approaches, relative drop in performance is less for non-contrastive approaches, and, (ii) R21D is more robust than ShuffleNet in both contrastive and non-contrastive domain.

5. Knowledge distillation for SSL

Knowledge distillation has been extensively studied with respect to image networks, employing the idea of learning under supervision from a larger, better-trained teacher network. Instead of leveraging a single teacher network, distilling knowledge from an ensemble of teacher networks is supposed to achieve more promising performance. In this regard, we utilize [11] extending it to the videos from image domain. The training details are mentioned in the supplementary. We use our benchmark models as pre-trained teachers in logical combinations, with the motive to investigate on four types of analysis: 1) performance with different models as teachers for various subset sizes, 2) whether teacher with different complexities within a pretext task provide orthogonal information, 3) knowledge distillation

	10k	30k	50k	100k
T1	61.9/24.4	68.4/31.7	68.8/35.4	70.2/36.8
T2	70.9/30.2	76.4/37.4	78.0/40.2	78.5/43.4
ST	83.0/45.3	94.6/51.5	91.2/48.3	87.0/46.4

Table 8. 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).

from different pre-training datasets, and, 4) effect of teachers from multiple pretext tasks. To save space, we have put most of the tables for KD experiments in supplementary, since teacher numbers are repeated from above tables.

Pre-training subset size Our first stream of experiments involved using KD teachers trained on a specific subset of the pretraining dataset K-400. We use teachers pretrained on the RSPNet pretext task, since the finetuning accuracies for these were the best among all others. The motivation behind the experiment was to observe if distillation using teachers trained on a smaller subset could yield a gain in performance, since this would reduce the training time and compute required. Using ShuffleNet as the student network, and ShuffleNet and R(2+1)D pretrained on 10k, 30k, 50k and 100k subsets as the teachers respectively, we evaluate the performance of the student networks for classification on UCF-101. ***Observations:*** From Table 8, we can see that student outperforms the teacher in all cases for both the datasets. The best performance is obtain on 30k subset.

Task complexities For the pretext tasks VCOP, PRP and RSPNet (Table 5), we use benchmark models for multiple complexities. It is imperative to investigate how networks train on increasing complexity of the same task learn and disseminate additional information, which a student network could take advantage of. We ensemble three models corresponding to each of the pretext tasks, for both ShuffleNet and R21D. Each ensemble consist of networks trained on C1, C2, and C3 for the same task, keeping the teacher and student architecture same. ***Observations:*** In case of PRP, R21D as a student outperforms teachers. CKA maps for VCOP and RotNet for R21D student depict block structures, indicative of its low performance. On the other hand, ShuffleNet outperforms teacher for all pretext tasks. Table is present in supplementary.

OOD analysis We examine whether knowledge distillation from two different datasets helps in improving performance or not. We use finetuned weights on UCF101, pretrained on K400 and SSV2 respectively as the two teacher networks for pretext tasks RotNet and VCOP. ***Observations:*** For both RotNet and VCOP, we observe that the student network outperforms the teacher accuracies by ana average of 20.5% and 12.6% respectively. This demonstrates that knowledge learned from both datasets is in fact, complementary in nature. Tabl is provided in supplementary.

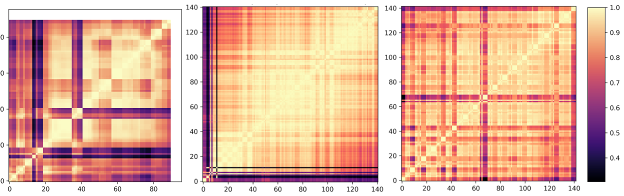


Figure 3. CKA maps for layer representations: R21D-teacher, ShuffleNet-teacher, and, Shufflenet-student for RSPNet 30k subset (Left to right).

Pretext task categories Finally, we look into knowledge distillation of teachers from multiple pretext task with the same architecture. Here, the motivation is to analyze whether the combination of spatial and temporal pretext tasks as teachers learn complementary information and outperform the standalone spatio-temporal pretext task training. From non-contrastive tasks, we employ VCOP and RotNet as teachers, and, similarly from contrastive, CVRL and TDL. ***Observations:*** We see that student network outperforms the standalone spatio-temporal pre-training for both contrastive and non-contrastive by a margin of $+39.2\%$ and $+1.5\%$ respectively on R21D backbone. Table is provided in the supplementary.

Inference: We derive the following conclusions from KD experiments: 1) KD helps in reduction of training subset size, 2) Different complexities can help models learn complementary features, 3) Knowledge from different datasets brings in complementary information, and, 4) Orthogonal features are learnt across different categories of pretext tasks, and different architectures. Form qualitative point of view, we observe that student's CKA maps is perfectly symmetrical grid like plot (Fig. 3) with no block formations which indicates no redundancy, and, thus, improve in performance over teachers.

5.1. Downstream tasks

For evaluation, we look into two downstream tasks: action classification and clip retrieval.

Action Classification For this task, the model is finetuned end-to-end on downstream datasets, which are UCF101 and HMDB51 in our case. In Table 9, we compare our best performing model with other previous state-of-the-art approaches. ***Observations:*** With only 30k videos compared to 240k videos used by other pretext tasks, we show that our model outperforms by good margin on UCF101 against single and multi-modal approaches. On HMDB51, we have a comparable performance. TCLR effectively takes in a larger clip duration at pre-training stage, whereas, other approaches use a bigger frame size or more number of frames.

Approach	NxW/H	UCF101	HMDB51
PacePred [54]	16x112	77.1	36.6
STS [51]	16x112	77.8	40.5
VideoMoCo [37]	16x112	78.7	49.2
RSPNet [8]	16x112	81.1	44.6
TaCo [5]	16x224	81.8	46.0
TCLR [10]	16x112	88.2	60.0
CVRL† [40]	32x224	92.9	67.9
VideoMAE * [49]	16x112	76.2	45.4
Multi-Modal			
AVTS† [29]	25x224	83.7	53.0
GDT [38]	32x112	89.3	60.0
XDC [3]	32x224	84.2	47.1
Ours *	16x112	97.3	51.5

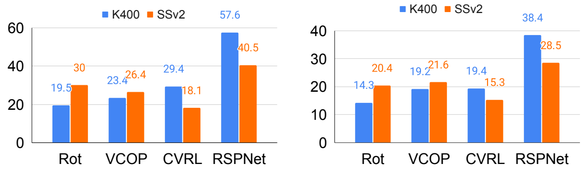
Table 9. Comparison with previous approaches pre-trained on K400 full set. Ours (* best performing) is RSPNet pretrained on 30k subset of K400. † represents model with different backbone than R21D. * reproduced results.

Clip retrieval For this downstream task, we generate the feature vectors using pretraining weights. Nearest neighbour is found out by measuring the cosine distance between the test feature vector and training feature vectors. Here, we compare the Top@5 on UCF101 and HMDB51 for both architectures, pre-trained on K400 and SSv2. ***Observations:*** Amongst non-contrastive tasks, SSv2 learns better features and outperforms K400 (Fig. 4) on both UCF101 and HMDB51, whereas, it's reversed for contrastive tasks. Overall the best performance is with pretraining on K400 with RSPNet on both the datasets.

6. Discussions and analysis

Discussions Apart from the factors we analyzed, we look into different backbone architectures as well. We investigate three small and three big capacity networks on network parameters, floating point operations (FLOPs), and, performance. The experimentation is done on RSPNet and VCOP with same constraints. ***Network parameters:*** Diving deeper into various network architectures, we can attribute that backbone architecture matters and it's not always necessary that a model with highest number of parameters is able to learn the better features. We observe that all the smaller capacity networks outperform R3D network. In bigger capacity networks, R21D has less parameters than C3D, but it outperforms the later in case of RSPNet and VCOP. With the residual connections in R3D as compared to C3D architecture, one would think that R3D will have higher accuracy. However, C3D outperforms R3D by 5.5% on RSPNet. Table is provided in the supplementary.

Recommendations Looking into several factors, few recommendations to set up the recipe for self-supervised training could based on our findings: 1) Pretrain dataset size: It depends upon learning capability of pretext task and model architecture. For the studied CNN architectures around 30K subset size was found effective. 2) In general, CNN requires less, whereas, transformer requires more data, 3) Train time - 100-150 epochs is suitable amongst all pretext tasks, and, 4) In case of out-of-distribution datasets, type of pre-training data matters.



(a) UCF101 (b) HMDB51

Figure 4. Top@5 Clip Retrieval - R21D on a) UCF101 and b) HMDB51, pre-trained on K400 and SSv2 - 30k subset.

Limitations This study explores different aspects of self-supervised learning in videos. We explore certain set of experiments based on our conclusion from the preliminary experiments. We agree that exhaustive set of experiments will substantiate our claims further, but it will also require large amount of computation resources. Also, we observe that different pre-training dataset affects the performance on both downstream tasks, finetuning accuracy and clip retrieval, differently. We mainly focus on Kinetics and Something-Something V2 as our pretraining dataset, but pre-training on different set of datasets with varying distribution will provide interesting insights in this domain. This can be an interesting area to explore in future.

7. Conclusion

In this study, we explore different parameters for self-supervised learning in video domain. We set a benchmark in self-supervised learning for videos which provides an intuitive task categorization and enables better comparison of different pretext tasks. Such an analysis has never been explored for video understanding to best of our knowledge. With this, we came up with the following conclusions: 1) different transformation-based pretext tasks prevail in contrastive vs non-contrastive; 2) complexity variation of pretext task may leave the network unoptimizable on the task; 3) pre-training dataset matters in case of OOD datasets; 4) Contrastive based pretext tasks are more robust to noise in comparison to non-contrastive ones; 5) Knowledge distillation helps in learning of complimentary information that provides benefits across multiple axes for self-supervised domain; and 6) we concluded that architectural design does matters, and, with the advancement of embedded-based technology it's important to look into networks with smaller capacity as they have low memory requirement and can be easily deployed with real-time performance.

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] 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

[10] 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

[11] 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. 6

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

[13] 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. 2

[14] 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

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

[16] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. *arXiv preprint arXiv:1905.01235*, 2019. 1, 2

[17] 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

[18] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *European Conference on Computer Vision*, 2020. 1, 2

[19] Srinidhi Hegde, Ranjitha Prasad, Ramya Hebbalaguppe, and Vishwajith Kumar. Variational student: Learning compact and sparser networks in knowledge distillation framework, 2019. 3

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

[21] 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. 2, 3

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

[23] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Regularizing very deep neural networks on corrupted labels. *CoRR*, abs/1712.05055, 2017. 3

[24] 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

[25] 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

[26] 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

[27] 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

[28] 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. 1, 2

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

[30] 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

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

[32] 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. 2, 3, 4

[33] 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

[34] Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. *CoRR*, abs/1902.03393, 2019. 3

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

[36] 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. 4

[37] 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

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

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

[40] 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

[41] 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. 2, 3

[42] 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

[43] 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

[44] 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

[45] 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

[46] 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, 2

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

[48] 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

[49] 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, 8

[50] 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, 2

[51] 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

[52] 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

[53] 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

[54] 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

[55] 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

[56] 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

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

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

[59] 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. 2, 3