How Severe is Benchmark-Sensitivity in Video Self-Supervised Learning?

Fida Mohammad Thoker, Hazel Doughty, Piyush Bagad, Cees Snoek
University of Amsterdam
{f.m.thoker,hazel.doughty,piyush.bagad@student,cgmsnoek}@uva.nl

Abstract

Despite the recent success of video self-supervised learning, there is much still to be understood about their generalization capability. In this paper, we investigate how sensitive video self-supervised learning is to the currently used benchmark convention and whether methods generalize beyond the canonical evaluation setting. We do this across four different factors of sensitivity: domain, samples, actions and task. Our comprehensive set of over 500 experiments, which encompasses 7 video datasets, 9 self-supervised methods and 6 video understanding tasks, reveals that current benchmarks in video self-supervised learning are not a good indicator of generalization along these sensitivity factors. Further, we find that self-supervised methods considerably lag behind vanilla supervised pre-training, especially when domain shift is large and the amount of available downstream samples are low. From our analysis we distill the *SEVERE-benchmark*, a subset of our experiments, and discuss its implication for evaluating the generalizability of representations obtained by existing and future self-supervised video learning methods.

1 Introduction

Video self-supervised learning has progressed at a tremendous pace in recent years, *e.g.* [69, 1, 55, 54, 53, 52], as it offers a crucial starting point from which to learn. This is especially important for video understanding applications, where annotating large amounts of data is extremely expensive, error-prone and sensitive to annotator bias. Hence, learning video representations through self-supervision is crucial, especially for use cases where the downstream video data is limited, because of the domain, task or actions the video contains. However, the majority of current works in video self-supervised learning, *e.g.* [75, 47, 48, 4, 51] do not test beyond standard benchmarks. The standard protocol is to use unlabeled Kinetics-400 [35] for pre-training and then measure performance by finetuning on two action recognition datasets: UCF-101 [61] and HMDB-51 [41]. While these benchmarks have facilitated the impressive progress of video self-supervised learning in recent years, they cannot indicate the generalizability of such methods as these pre-training and downstream datasets are all similar in appearance and the type of actions they contain. Some methods have started to report finetuning performance on additional datasets like Something-Something-v2 [24] in [69, 53, 19], Diving-48 [42] in [14, 72], AVA [26] in [74, 76, 19], EPIC-Kitchens-100 [13] in [76]. However, such evaluations are insufficient to understand the generalization of video self-supervised methods by themselves since they only add a single additional dataset, often without comparison to prior methods.

In this work, we address the essential need to gauge the sensitivity of existing video self-supervised methods to the current benchmark by thoroughly evaluating their performance for generalization across diverse downstream settings. Similar benchmarking studies have been performed for self-supervised pre-training in images [12, 32, 40, 16, 23, 77, 37, 80, 5, 49, 57, 67, 17], which investigate the importance of pre-training datasets [12, 40, 23] and backbone architecture [37], transferability [32,

Preprint. Under review.

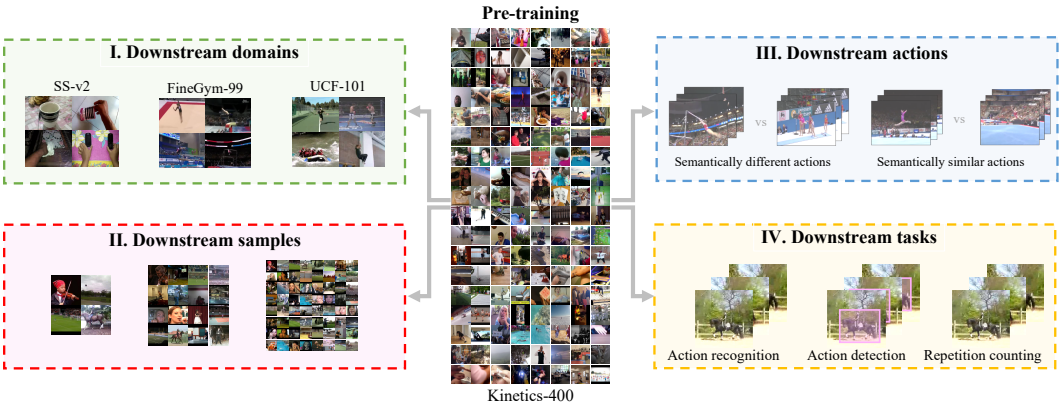

Figure 1: **Benchmark-sensitivity.** We evaluate the sensitivity of 9 video self-supervised learning methods along four downstream factors which vary from the pre-training source: the domain, the samples, the actions and the task.

16, 49, 68], amongst other aspects. Unfortunately, lessons from these works do not directly transfer to video self-supervised learning. First, video self-supervised tasks are distinct from those of images as they are designed to understand the temporal dimension of video [53, 14, 69, 76] in addition to the spatial understanding needed in images [9]. Second, video is multi-modal and several methods [52, 4, 48] are designed to exploit cross or multi-modal understanding, which is again absent in image-based methods. For videos, [19] extend four image-based self-supervised methods to videos and investigate their performance focusing on different pre-training set ups. We take inspiration from this and benchmarking works in image self-supervised learning and perform a much-needed study for understanding the generalizability of self-supervised methods for video in relation to different downstream factors.

As our first contribution, we identify the problem of benchmark-sensitivity in video self-supervised learning and examine this sensitivity along the factors of domain, samples, actions and task. As our second contribution, we perform an extensive evaluation which spans a total of over 500 experiments with 9 video self-supervised learning methods across 7 video datasets and 6 video understanding tasks. We find that standard benchmarks in video self-supervised learning do not indicate generalization along the said sensitivity factors and vanilla supervised pre-training outperforms self-supervised pre-training, particularly when domain change is large and there are only a few downstream finetuning samples available. Third, we propose a subset of our experiments as the SEVERE-benchmark for future self-supervised learning methods to benchmark generalization capability. We also discuss the implication of this benchmark for evaluating the generalizability of representations obtained by existing methods as well as the nature of video self-supervised objectives that currently generalize well.

2 Identifying Benchmark Sensitivity

The vast majority of current works in video self-supervised learning evaluate their approach by pre-training on Kinetics-400 [35] and finetuning the learned representation for action recognition on UCF-101[61] and HMDB-51[41]. Some works [52, 14, 65, 69, 53, 4, 21, 43, 30] also report performance on video retrieval for UCF-101 and HMDB-51 and several recent works [55, 76, 56] compare linear evaluation performance on Kinetics-400. However, these downstream datasets are very similar to each other and also share many similarities with the pre-training dataset of Kinetics-400. Videos in all three datasets are clips collected from YouTube that are mostly recorded with a single camera containing a well-positioned single human actor. In terms of class labels, all datasets focus on similar, coarse-grained and mutually exclusive actions with many actions common between pre-training and downstream datasets. Besides all these data similarities, the existing evaluations also ignore a major benefit of self-supervised representation learning for videos, *i.e.* finetuning the representation with only a small amount of data samples and transferring to other video understanding tasks beyond action recognition. Hence, we believe the current benchmark standard is insufficiently equipped to gain a true understanding of where video self-supervised models are successful, as it

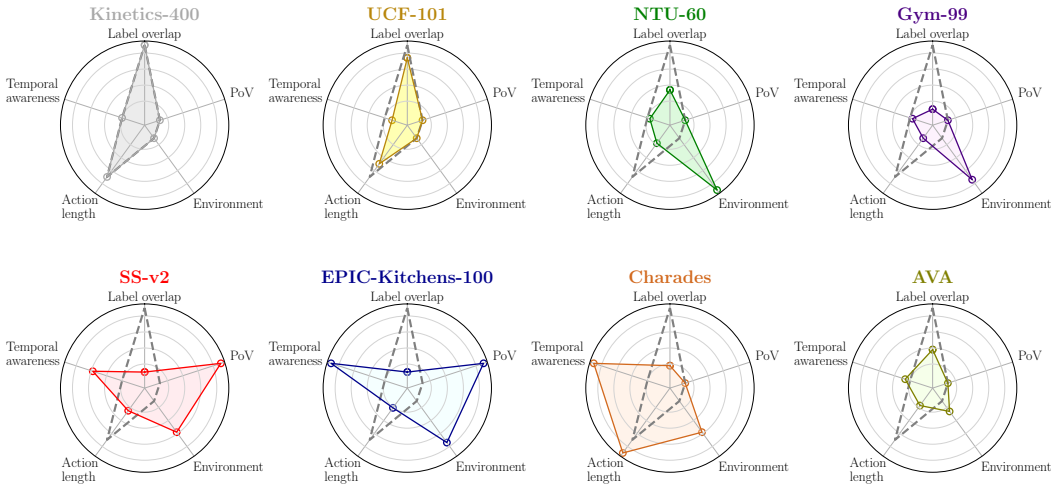

Figure 2: **Video dataset characteristics.** Characterizing domain shift in datasets via difference in label overlap, point-of-view (PoV), environment, action length and temporal awareness with Kinetics-400 (shown by dotted line). Kinetics-400 and UCF-101 are highly similar to each other, while datasets like Something-Something-v2, EPIC-Kitchens-100 and Charades have different attributes compared to Kinetics-400.

cannot show the generalizability or the sensitivity of methods to factors such as domain shift, amount of finetuning data samples, action similarity or task shift. In this study, we identify the sensitivity of existing evaluations and thoroughly benchmark self-supervised video learning methods along four sensitivity factors as depicted in Fig. 1.

I. **Downstream domain.** First, we analyse whether features learned by self-supervised models transfer to datasets that vary in domain with respect to the pre-training dataset.

II. **Downstream samples.** Second, we evaluate the sensitivity of self-supervised methods to the number of downstream samples available for finetuning.

III. **Downstream actions.** Third, we investigate whether self-supervised methods can learn fine-grained features required for recognizing semantically similar actions.

IV. **Downstream task.** Finally, we study the sensitivity of video self-supervised methods to the downstream task and question whether self-supervised features can be used beyond action recognition.

2.1 Downstream Video Datasets

We evaluate various self-supervised models along our four sensitivity factors on 7 video datasets: **UCF-101** [61], **NTU-60** [58], **FineGym** (Gym-99) [59], **SomethingSomething-v2** (SS-v2) [24], **EPIC-Kitchens-100** (EK-100) [13], **Charades** [60] and **AVA** [26]. They include a considerable variety in video domain, the actions they contain and cover a range of video understanding tasks. To get a sense of the differences between these downstream datasets and the Kinetics-400 source dataset, we summarize their similarity to Kinetics-400 by radar plots in Fig. 2 based on several attributes. *Environment* refers to the variety of settings contained in the dataset. This can be very specific, *e.g.* a kitchen, or varied when many different settings are included. *Point-of-view* is whether a video is recorded from a first-person or third-person viewpoint. *Temporal awareness* defines the extent to which temporal context is required to recognize or detect actions. We quantify this as the point at which performance saturates with increasing temporal context in the input. *Label overlap* is the fraction of actions in a target dataset that are also present in Kinetics-400. *Action length* is the temporal length of the actions in seconds. Details are provided in the appendix.

2.2 Evaluated Self-Supervised Video Learning Methods

Self-supervised learning methods in video can be grouped into two categories based on the objective they use: pretext task methods and contrastive learning methods. Pretext task methods are based on

predictive tasks such as solving spatio-temporal jigsaw puzzles [2, 31, 36], rotation prediction [34], frame and clip order [47, 20, 63, 75, 78], video speed [7, 33, 11, 79, 71], video completion [44], predicting motion statistics [70], tracking random patches in video frames [69] or audio-visual clustering [8, 29, 4, 3]. Contrastive learning methods discriminate between "positive" and "negative" pairs to learn invariances to certain data augmentations and instances either from visual-only input [51, 14, 27, 76, 55, 43, 15, 62] or multi-modal data [52, 48, 28, 64, 45, 39]. Some methods also combine the pretext and contrastive approaches [65, 53, 82, 6, 15, 30]. We consider a total of 9 video-based self-supervised methods which achieve good performance on current benchmarks and cover a range of self-supervised paradigms in the video domain, including contrastive learning, pretext-tasks, their combination and cross-modal audio-video learning.

Due to the high computational cost of training self-supervised methods, we focus on works with publicly available weights for a common R(2+1)D-18 network [66] pre-trained on Kinetics-400 [35]: **MoCo** [10], **SeLaVi** [4], **VideoMoCo** [51], **Pretext-Contrast** [65], **RSPNet** [53], **AVID-CMA** [48], **CtP** [69], **TCLR** [14] and **GDT** [52]. We compare these to no pre-training, *i.e.* training from scratch, and fully supervised pre-training for the task of action recognition. It is worth noting that since we use publicly available models we cannot control the exact pre-training setup. There are subtle differences in the training regime for each method, such as how long the models were trained, the data augmentations used and the batch size. Details of these differences are provided in the appendix. However, all models use the same backbone and pre-training dataset thus we can evaluate their downstream abilities in exactly the same way. To finetune for downstream tasks we simply attach a task-dependent head at the last layer of the pre-trained R(2+1)D-18 backbone to produce label predictions for the corresponding task. For a fair comparison, we use the same set of hyper-parameters, optimization and pre-processing during the downstream training of each pre-trained model.

3 Sensitivity Factor I: Downstream Domain

We first investigate to what extent self-supervised methods learn features that are applicable to action recognition in any domain. We evaluate the suite of pre-trained models on UCF-101, NTU-60, Gym-99, SS-v2 and EK-100 for the task of action recognition. It is worth noting that as well as variety in domain, these datasets include variety in the amount of training data (9.5k - 168k examples) and cardinality of classification (60 - 300 classes). We attach a single classification layer to the pre-trained backbone and evaluate the models' performance on the downstream task in two settings. First, **full finetuning** where we train the whole network from the initialization of the pre-trained weights. Second, **linear evaluation** where we train the classification layer only using the frozen features of pre-trained backbones. We follow the standard splits proposed in the original datasets and report video-level top-1 accuracy on the test sets. The details about splits, pre-processing, training for each dataset are provided in the appendix.

Full finetuning. The left part of Table 1 shows the results of full finetuning. From the results, it is clear that all self-supervised methods are very effective on UCF-101 as there is a significant gap between training from scratch and using self-supervised pre-training. This gap is reduced as the difference between Kinetics-400 and the downstream domain increases. SeLaVi, MoCo and AVID-CMA in particular are evidence of this as these methods suffer when datasets have higher temporal awareness and less label overlap with Kinetics-400. When moving from UCF-101 to NTU-60 and Gym-99 there is a change in the ordering of self-supervised methods. This demonstrates a high performance on UCF-101 does not guarantee a self-supervised model is generalizable to other domains. The change in ranking is even more prominent for SS-v2 and EK-100, which require the most temporal awareness and also shift to a first-person viewpoint. This is particularly noticeable for AVID-CMA. On these datasets, MoCo has similar results to no pre-training, which is evidence that video-specific self-supervised learning methods are needed and that image-based methods are insufficient. Overall, supervised pre-training achieves good performance across the board, outperforming self-supervised methods on the most similar domain (UCF-101) as well as the most dissimilar domains (SS-v2 and EK-100). Amidst the tested self-supervised models, CtP, RSPNet, VideoMoCo and TCLR stand out as the self-supervised pre-training methods most generalizable to different domains.

Linear classification. The right part of Table 1 shows the results for linear classification. As with finetuning, the ranking among the self-supervised methods changes as the domain difference

Table 1: **Sensitivity Factor I: Downstream Domain.** Video self-supervised methods evaluated across datasets with increasing domain shift with respect to the source dataset (see Fig. 2). Colors denote relative rankings across methods for each dataset, ranging from low high. The ranking of methods is domain-sensitive for both finetuning and linear classification and becomes less and less correlated with the current UCF-101 benchmark as the domain shift increases.

Pre-training	Finetuning					Linear Evaluation					
	UCF101	NTU60	Gym99	SSv2	EK 100	K 400	UCF101	NTU60	Gym99	SSv2	EK 100
None	75.4	92.9	89.4	56.8	25.7	-	-	-	-	-	-
MoCo	83.5	93.4	90.6	57.0	26.4	34.5	65.4	16.0	21.2	7.4	21.4
SeLaVi	84.9	92.8	88.9	56.4	33.8	24.1	51.2	15.7	20.2	4.5	22.4
VideoMoCo	85.8	94.1	90.5	58.8	43.6	31.0	66.3	51.6	41.6	19.5	25.7
Pretext-Contrast	86.6	93.9	90.3	57.0	34.3	22.4	57.2	17.6	30.0	10.9	20.0
RSPNet	88.5	93.9	91.3	59.4	42.7	46.0	76.6	33.5	32.2	12.5	24.9
AVID-CMA	89.3	94.0	90.6	53.8	29.9	43.5	78.1	53.9	45.1	16.1	22.5
CtP	89.8	94.3	92.2	60.2	42.8	7.6	37.9	22.6	30.6	12.2	20.0
TCLR	90.8	94.1	91.5	60.0	36.2	19.9	63.3	33.5	33.0	10.8	21.8
GDT	91.1	93.9	90.4	57.8	37.3	38.6	75.7	38.2	34.2	11.9	25.3
Supervised	94.1	93.9	91.8	61.0	47.7	65.9	91.7	45.5	42.7	16.6	26.6

between the pre-training and the downstream dataset increases. For example, VideoMoCo ranks lower than GDT and RSPNet for UCF-101 and Kinetics-400 but ranks higher than both for all other datasets. This again demonstrates that performance on UCF-101 does not give a complete picture of a self-supervised model's success. We also observe that linear evaluation on Kinetics-400, as some papers report [55, 56, 76], has the same issue since it is highly correlated to UCF-101 performance. For UCF-101 and Kinetics-400, self-supervised models with contrastive objectives learn highly discriminative features compared to the non-contrastive models. This can be seen by comparing contrastive models AVID-CMA, GDT and RSPNet to non-contrastive SeLaVi and CtP. From the NTU-60 and Gym-99 results we observe that as the label overlap between the pre-training and the downstream dataset decreases, the performance gap between finetuning and linear evaluation increases considerably. This is true for both supervised and self-supervised pre-training. The most generalizable methods in the linear classification setting are contrastive methods VideoMoCo and AVID-CMA as well as supervised pre-training. Interestingly, there are cases where VideoMoCo and AVID-CMA even outperform supervised pre-training, namely for NTU-60, Gym-99 and SS-v2.

> **Conclusion.** We observe from Table 1 that performance for both UCF-101 finetuning and Kinetics-400 linear evaluation is not indicative of how well a self-supervised video model generalizes to different downstream domains, with the ranking of methods changing substantially across datasets and whether full finetuning or linear classification is used.

4 Sensitivity Factor II: Downstream Samples

The previous section analyzed sensitivity to change in the downstream domain by evaluating performance on several different datasets. However, each of these datasets contains a large number of labeled examples for finetuning, which means training from scratch already obtains good performance. Not all domains and use cases have ample labeled video examples available for finetuning, thus we investigate what the impact of the number of finetuning samples is and whether self-supervised methods can be beneficial in scenarios where we have little data to finetune with. We vary the amount of finetuning data, beginning from 1000 samples, sampled uniformly from the classes, and double the amount until we reach the full finetuning training set size. We report on four of the downstream datasets from the previous section: UCF-101, NTU-60, Gym-99 and SS-v2. The results are summarized in Fig. 3.

We first observe that the trends in the low data regime are different from those for the full data regime. The gap between supervised and self-supervised pre-training is much larger in low data settings, particularly for UCF-101 and Gym-99. NTU is an exception, where, with 1000-4000 samples CtP, GDT, AVID-CMA and TCLR outperform supervised pre-training. As also observed for the downstream domain, the ranking of self-supervised models changes with the amount of downstream examples available for finetuning. For example, on UCF-101, RSPNet is much more successful

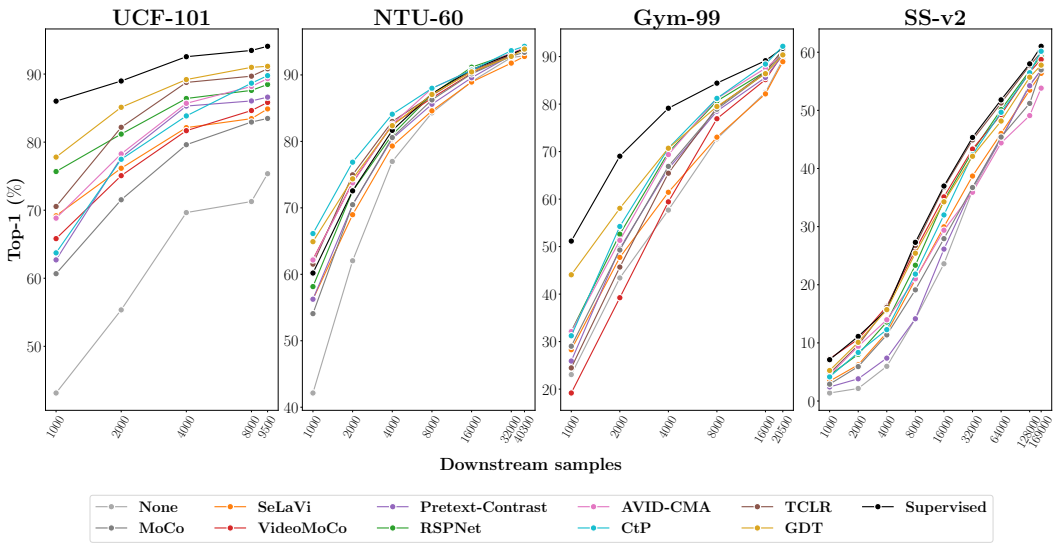

Figure 3: **Sensitivity Factor II: Downstream Samples.** Comparison of video self-supervised learning methods using varying number of finetuning samples for four downstream datasets. Both the gap and rank among pre-training methods are sensitive to the number of samples available for finetuning.

than CtP and TCLR when using only 1000 samples. This is because some self-supervised models benefit more than others from an increased amount of downstream samples. For example, CtP is one of the most generalizable pre-training strategies when finetuning with the full amount of data on UCF-101, Gym-99 and SS-v2, but this is not the case with fewer training samples. Interestingly, GDT is consistently high in the ranking with low amounts of finetuning samples. This is likely due to the large number of temporal augmentations it uses, which help the generalization ability when the training data is limited.

Conclusion. We observe from Fig. 3 that video self-supervised models are highly sensitive to the amount of samples available for finetuning, with both the gap and rank between methods changing considerably across sample sizes on each dataset.

5 Sensitivity Factor III: Downstream Actions

As indicated earlier, existing evaluations of self-supervised video learning methods have been limited to coarse-grained action recognition. In this section, we investigate whether current self-supervised tasks are only effective for these types of benchmarks or whether they are able to learn features that are useful for differentiating more challenging and semantically similar actions.

FineGym [59] provides us with an experimental setup to study sensitivity to this factor. The dataset contains different evaluations with varying levels of semantic similarity, namely action recognition *across all events*, *within an event* or *within a set*. Recognition *across all events* uses the whole of Gym-99 containing all actions from four gymnastic events. For recognition *within an event* there are two subsets: Vault (VT) and Floor Exercise (FX) containing only actions from these two events. Recognition *within a set* has two subsets namely FX-S1, containing different *leaps-jumps-hops* in Floor Exercise, and UB-S1, which consists of types of *circles* in Uneven Bars. We also experiment with the long-tailed version of FineGym, Gym-288, which adds 189 more tail classes to Gym99. Details of these subsets are in the appendix. As before, we attach a classification head to the pre-trained models and finetune the whole network with the training set of each subset. We report Top-1 accuracy (mean per-class) on the testing sets following [59]. The results are shown in Table 2.

Performance of self-supervised methods also varies considerably across downstream actions. The methods that perform best on Gym-99 do not necessarily generalize well to the subsets with higher semantic similarity among actions. This is particularly noticeable for RSPNet and TCLR which drop in the ranking for the within-set subsets. All self-supervised methods, except GDT, struggle on Vault,

Table 2: **Sensitivity Factor III: Downstream Actions.** Video self-supervised models evaluated on different semantic similarities of action in FineGym: across events, within an event and within a set. Colors denote relative rankings across methods for each dataset, ranging from low high. Many methods struggle on the within a set benchmark where actions are most semantically similar.

Pre-training	Gym99					Gym288
	Across Events	Within Event		Within Set		Across Events
	All	Vault	Floor	FX-S1	UB-S1	All
None	84.4	24.7	75.9	45.0	84.0	50.0
SeLaVi	84.8	25.4	76.0	50.2	81.5	52.8
Pretext-contrast	85.7	28.5	81.4	65.8	86.2	52.7
AVID-CMA	85.8	30.4	82.7	67.2	88.4	52.5
MoCo	86.2	33.2	83.3	65.1	85.0	55.1
VideoMoCo	86.4	28.4	79.5	60.4	82.1	54.1
GDT	86.5	36.9	83.6	65.7	81.6	55.4
RSPNet	87.6	33.4	82.7	63.5	85.1	55.2
TCLR	88.0	29.8	84.3	61.0	85.3	55.4
CtP	88.3	26.8	86.2	79.7	88.4	56.5
Supervised	88.0	37.7	86.1	81.0	86.9	58.4

likely due to the very intense motions of this action. Surprisingly, MoCo performs reasonably well when actions are more semantically similar, and is even comparable to GDT and RSPNet. The best self-supervised method for the subsets with high semantic similarity is CtP. This is especially evident from FX-S1 where it outperforms the second-best self-supervised method, AVID-CMA, with a 12% absolute margin. As with downstream domain and downstream samples, supervised pre-training generalizes better than self-supervised methods across downstream actions with only CtP achieving comparable performance.

Table 2 also compares balanced Gym-99 with long-tailed Gym-288. We observe that self-supervised methods are not robust to this change in distribution, with the gap in performance with respect to supervised pre-training increasing. However, the ranking remains consistent, meaning the performance on the balanced set is generally indicative of the performance on the long-tailed set.

> **Conclusion.** Most self-supervised methods in Table 2 are sensitive to the actions present in the downstream dataset and do not generalize well to more semantically similar actions. This further emphasizes the need for proper evaluation of self-supervised methods beyond current coarse-grained action classification.

6 Sensitivity Factor IV: Downstream Tasks

The fourth factor we investigate is whether self-supervised video models are sensitive to the downstream task or whether features learned by self-supervised models are useful to video understanding tasks beyond action recognition. We evaluate this in two ways. First, we keep the domain fixed and evaluate different tasks in a domain similar to the pre-training dataset. We also explore further tasks by changing the domain and seeing how these two factors interplay.

6.1 Task-shift within domain.

We consider three different tasks which are all defined for UCF-101: spatio-temporal action detection [38], repetition counting [81] and arrow-of-time prediction [22]. Using UCF-101 allows us to keep the domain fixed across tasks and eliminates the impact of domain shift. Note that each task uses a different subset of the full UCF-101 dataset, however, the domain remains consistent. For each task, we use the R(2+1)D-18 networks as the pre-trained backbones as before and attach task-dependent heads. We report mean Average Precision for spatio-temporal localization [46], mean absolute counting error for repetition counting [81] and classification accuracy for arrow-of-time prediction [22, 73]. Further details are in the appendix.

Table 3: **Sensitivity Factor IV: Downstream Tasks.** Transferability of self-supervised video learning methods across video understanding tasks. Colors denote relative rankings across methods for each dataset, ranging from low high. Note that for repetition counting lower (error) is better. Self-supervised features are transferable to different downstream tasks when the domain shift is low, but struggle when there is also a domain shift. Action recognition on UCF-101 is not a good proxy for self-supervised video learning use cases where a downstream domain- and task-shift can be expected.

Pre-training	**Task-shift within domain**				**Task-shift out of domain**	
	Action Recognition	Action Detection	Repetition Counting	Arrow of Time	Multi-label Recognition	Action Detection
None	75.4	0.327	0.232	56.1	7.9	7.4
MoCo	83.5	0.416	0.220	80.3	8.1	11.7
SeLaVi	84.9	0.419	0.171	77.4	8.2	10.2
VideoMoCo	85.8	0.440	0.171	72.9	10.5	13.1
Pretext-contrast	86.6	0.462	0.168	77.2	8.9	12.7
RSPNet	88.5	0.467	0.151	87.0	9.1	14.1
AVID-CMA	89.3	0.435	0.162	83.3	8.4	10.0
CtP	89.8	0.465	0.178	77.1	9.6	10.0
TCLR	90.1	0.476	0.149	85.6	11.1	10.8
GDT	91.1	0.463	0.137	76.4	8.5	12.6
Supervised	94.1	0.482	0.137	77.0	23.6	17.9

From the results in Table 3, we observe that self-supervised learning is beneficial to tasks beyond action recognition, with almost all methods outperforming training from scratch on spatio-temporal action detection, repetition counting and arrow-of-time prediction. Action detection results are well correlated with action recognition. Repetition counting and arrow-of-time have less correlation with action recognition, suggesting that the current benchmark on UCF-101 action recognition by itself is not a good indication of how well self-supervised methods generalize to other tasks. For repetition counting and arrow-of-time prediction, some methods perform comparably to or outperform supervised pre-training. Notably, RSPNet and TCLR generalize the best across these tasks, with GDT also performing well on repetition counting. CtP ranks high on action recognition and detection but performs modestly for repetition counting. This shows that different methods have different task sensitivity, so a thorough evaluation along downstream tasks is needed.

6.2 Task-shift out of domain.

We also evaluate how well the self-supervised models generalize when both the domain and the task change. We do so with two popular video understanding benchmarks: long-term multi-label classification on Charades [60] and short-term spatio-temporal action detection on AVA [26]. For both, we follow the setup and training procedure from [18] with R(2+1)D-18 models as the pre-trained backbone and we measure performance in mean Average Precision. Details are in the appendix. From the results in Table 3, we observe that supervised pre-training is far more generalizable than all self supervised methods, which all struggle considerably when both the domain and task change. For long-term action classification on Charades, TCLR is slightly better than other methods. On AVA, RSPNet is the best performing self-supervised method with VideoMoCo second. In Section 3, we earlier observed that these were two of the methods more robust to domain shift suggesting that this factor is key to success on AVA.

> **Conclusion.** The results in Table 3 reveal that action classification performance on UCF-101 is mildly indicative for transferability of self-supervised features to other tasks on UCF-101. However, when methods pre-trained on Kinetics-400 are confronted with a domain change in addition to the task change, UCF-101 results are no longer a good proxy and the gap between supervised and self-supervised pre-training is large.

7 SEVERE-benchmark

As evident from the results in previous sections, current video self-supervised methods are benchmark-sensitive to the four factors we have studied. Based on our findings, we propose the SEVERE-

Table 4: **Proposed SEVERE-benchmark** for evaluating video self-supervised methods for generalization along downstream domains, samples, actions and tasks.

Pre-training	Existing	SEVERE-benchmark							
		Domains		Samples		Actions		Tasks	
	UCF101	SS-v2	Gym-99	UCF (10^3)	Gym-99 (10^3)	FX-S1	UB-S1	UCF-RC	Charades-MLC
None	75.4	56.8	89.4	43.1	23.1	45.0	84.0	0.232	7.9
MoCo	83.5	57.0	90.6	60.7	29.0	65.1	85.0	0.220	8.1
SeLaVi	84.9	56.4	88.9	69.2	28.3	50.2	81.5	0.171	8.2
VideoMoCo	85.8	58.8	90.5	65.8	19.2	60.4	82.1	0.171	10.5
Pretext-Contrast	86.6	57.0	90.3	62.7	25.9	65.8	86.2	0.168	8.9
RSPNet	88.5	59.4	91.3	75.7	32.2	63.5	85.1	0.151	9.1
AVID-CMA	89.3	53.8	90.6	68.8	32.1	67.2	88.4	0.162	8.4
CtP	89.8	60.2	92.2	63.7	31.2	79.7	88.4	0.178	9.6
TCLR	90.8	60.0	91.5	70.6	24.5	61.0	85.3	0.149	11.1
GDT	91.1	57.8	90.4	77.8	44.1	65.7	81.6	0.137	8.5
Supervised	94.1	61.0	91.8	86.0	51.2	81.0	86.9	0.137	23.6

benchmark (SEnsitivity of VidEo REpresentations) for use in future works to more thoroughly evaluate new video self-supervised methods for generalization along the four sensitivity factors we have examined. Since we do not expect future works to run all the experiments from our study, we create a subset of experiments that are indicative benchmarks for each sensitivity factor and realistic to run. We summarize the benchmark composition in Table 4 and detail its motivation per factor.

Downstream domain. To measure a self-supervised model's domain sensitivity we recommend using Something-Something-v2 and FineGym-99. These two datasets come from domains distinct to Kinetics-400 and UCF-101 and also each other. FineGym-99 evaluates a model's ability to generalize to datasets with less distinctive backgrounds where there are few actions in common with Kinetics-400. SS-v2 evaluates the generalizability to actions that require high temporal awareness as well as the shift to a first-person viewpoint. It is evident from Table 4 that there are significant rank changes between UCF-101, Gym-99 and SS-v2 thus these three datasets provide a challenging subset for future methods.

Downstream samples. For the sample sensitivity, we recommend using 1000 samples on UCF-101 and Gym-99. Using 1000 samples showed the most dramatic difference from the full dataset size particularly for these datasets where there is a considerable gap between self-supervised and supervised pre-training as well as considerable rank change among the methods.

Downstream actions. To test generalizability to recognizing semantically similar actions, we recommend evaluating the two within-set granularities of Gym-99 *i.e.* FX-S1 and UB-S1. Both of these subsets have high semantic similarity between actions with methods currently struggling to generalize to both of these subsets as can be seen in Table 4. There is also a significant gap between supervised and self-supervised pre-training for FX-S1, highlighting the potential for future works in this area.

Downstream task. To evaluate the task sensitivity, we recommend future works use repetition counting on UCF-101 and multi-label action classification on Charades. Repetition counting on UCF-101 highlights different strengths to action recognition as it allows investigation of a model's ability to generalize to a task that requires more temporal understanding without measuring the impact of the domain. We recommend multi-label classification on Charades as it is currently a very challenging task for self-supervised models and allows the combination of domain change and task shift to be investigated. Code to compare on the SEVERE-benchmark will be made available.

8 Observations, Limitations and Recommendations

Observations. We hope that our study and resulting benchmark provides a helpful insight for future research to design novel self-supervised methods for generalizable video representation learning. From the benchmark results in Table 4, we observe that:

(i) There is no clear winner as different methods stand out in different downstream settings.

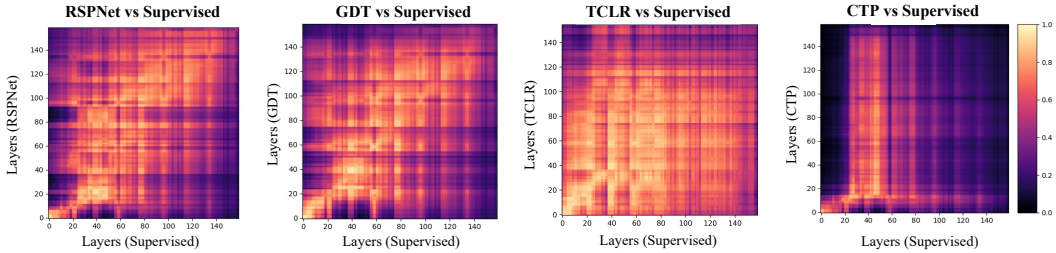

Figure 4: **Representation similarity** between features of top self-supervised methods and supervised pre-training on Kinetics-400 validation set (using centered kernel alignment [50]). Contrastive methods have a high correlation with supervised pretraining, while CtP's features are far away. Thus, showing potential for both imitating supervised learning as well as learning features distinct to it.

(ii) Supervised pre-training is dominant across all sensitivity factors, especially when the number of available downstream samples are limited and when there is a change in both the downstream domain and the downstream task.

(iii) Self-supervised contrastive methods that explicitly encourage features to be distinct across the temporal dimension transfer well. This is visible from the consistent performance of GDT, TCLR and RSPNet across different sensitivity factors.

(iv) Learning certain temporal invariances may prevent generalizability to temporal or fine-grained benchmarks. This is evident from GDT's performance on SS-v2 and UB-S1. These benchmarks require distinction between actions such as *moving something left* vs. *moving something right* in SS-v2 and *giant circle forwards* vs. *giant circle backwards* in UB-S1. The invariance to temporal reversal learned by GDT impacts its ability to recognize such actions. Similarly, MoCo outperforming VideoMoCo on the FX-S1 and UB-S1 Gym-99 subsets suggests that invariance to frame dropout in VideMoCo can harm the performance on highly similar actions.

(v) Pretext-tasks specific to videos can be effective to learn more fine-grained features. CtP generalizes well both to different domains where the background is less indicative of the action and to more semantically similar actions. The pretext task is to track and estimate the position and size of image patches moving in a sequence of video frames. Such a formulation requires the network to learn to follow moving targets and ignore the static background information. CtP's generalization success demonstrates that contrastive learning is not the only way forward for self-supervised video representation learning.

(vi) Fig. 4 shows the feature similarity on Kinetics using centered kernel alignment [50] between supervised pre-training and the best self-supervised methods *i.e*. GDT, RSPNet, TCLR, CtP. This figure illustrates that contrastive methods seem to imitate supervised pre-training as the correlation between supervised pre-training and the three contrastive methods (RSPNet, GDT and TCLR) is high. This explains the good performance of these methods on UCF-101 with 1000 examples. By contrast, CtP's features are far away from supervised pre-training. This is interesting because CtP generalizes well to new domains and actions, it shows that good generalization capability can be obtained without imitating supervised pre-training.

Limitations. While our study has highlighted the benchmark sensitivity of video self-supervised learning across four factors, there are many more factors that we do not consider in this work. Due to computational limits, we keep the source dataset fixed as Kinetics-400 and use publicly available pre-trained models. This means there is variability in the exact pre-training setup used by each model. We hope that future works will explore this factor as well as the impact of other large-scale pre-training datasets such as Ego4D [25] for the generalization of video self-supervised models. Another limitation of our study is that we only consider a fixed R(2+1)D-18 backbone, which is currently one of the most commonly used in video self-supervised learning. This allows our comparison between methods to be fair, however, it does limit the ability of methods to perform well on datasets such as EPIC-Kitchens-100. Another factor that could be explored further is the task. We have considered a selection of various video understanding tasks centered around actions. However, there are many more tasks that could be explored both action-centric and beyond, for example action anticipation, action segmentation, tracking and motion estimation.

Recommendations. Based on the results and our observations, we have several recommendations for future works in video self-supervised learning. (i) Our study has highlighted the need for more focus on generalizability of self-supervised learning methods, particularly along the domain and dataset size factors. (ii) Distinguishing across the temporal dimension is effective and is a useful direction to pursue further for generalizability. (iii) Pretext-tasks like the one used in CtP are good for the generalizability to domain and action, thus designing new video specific pretext tasks is a promising direction. This could also be combined with contrastive learning tasks to gain the benefits of both types of learning.

Acknowledgments and Disclosure of Funding

This work is part of the research programme Perspectief EDL with project number P16-25 project 3, which is financed by the Dutch Research Council (NWO) domain Applied and Engineering/ Sciences (TTW).

References

[1] Triantafyllos Afouras, Andrew Owens, Joon Son Chung, and Andrew Zisserman. Self-supervised learning of audio-visual objects from video. In *European Conference on Computer Vision*, 2020. 1

[2] Unaiza Ahsan, Rishi Madhok, and Irfan Essa. Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 179–189. IEEE, 2019. 4

[3] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *Advances in Neural Information Processing Systems*, volume 33, pages 9758–9770, 2020. 4

[4] Yuki M. Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 2, 4, 16

[5] Yuki M Asano, Christian Rupprecht, and Andrea Vedaldi. A critical analysis of self-supervision, or what we can learn from a single image. In *International Conference on Learning Representations (ICLR)*, 2020. 1

[6] Yutong Bai, Haoqi Fan, Ishan Misra, Ganesh Venkatesh, Yongyi Lu, Yuyin Zhou, Qihang Yu, Vikas Chandra, and Alan Yuille. Can temporal information help with contrastive self-supervised learning? *arXiv preprint arXiv:2011.13046*, 2020. 4

[7] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9922–9931, 2020. 4

[8] Brian Chen, Andrew Rouditchenko, Kevin Duarte, Hilde Kuehne, Samuel Thomas, Angie Boggust, Rameswar Panda, Brian Kingsbury, Rogerio Feris, David Harwath, et al. Multimodal clustering networks for self-supervised learning from unlabeled videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 8012–8021, 2021. 4

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning (PMLR)*, 2020. 2

[10] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 4, 16

[11] Hyeon Cho, Taehoon Kim, Hyung Jin Chang, and Wonjun Hwang. Self-supervised spatio-temporal representation learning using variable playback speed prediction. *IEEE Access*, 9:79562–79571, 2021. 4

[12] Elijah Cole, Xuan Yang, Kimberly Wilber, Oisin Mac Aodha, and Serge Belongie. When does contrastive visual representation learning work? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[13] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision (IJCV)*, 2021. 1, 3, 17, 27

[14] Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. Tclr: Temporal contrastive learning for video representation. *arXiv preprint arXiv:2101.07974*, 2021. 1, 2, 4, 16

[15] Ali Diba, Vivek Sharma, Reza Safdari, Dariush Lotfi, Saquib Sarfraz, Rainer Stiefelhagen, and Luc Van Gool. Vi2clr: Video and image for visual contrastive learning of representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1502–1512, 2021. 4

[16] Linus Ericsson, Henry Gouk, and Timothy M Hospedales. How well do self-supervised models transfer? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5414–5423, 2021. 1, 2

[17] Linus Ericsson, Henry Gouk, and Timothy M Hospedales. Why do self-supervised models transfer? investigating the impact of invariance on downstream tasks. *arXiv preprint arXiv:2111.11398*, 2021. 1

[18] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6201–6210, 2019. 8

[19] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3299–3309, 2021. 1, 2, 19

[20] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3636–3645, 2017. 4

[21] Kirill Gavrilyuk, Mihir Jain, Ilia Karmanov, and Cees G M Snoek. Motion-augmented self-training for video recognition at smaller scale. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10429–10438, 2021. 2

[22] A. Ghodrati, E. Gavves, and C. G. M. Snoek. Video time: Properties, encoders and evaluation. In *British Machine Vision Conference (BMVC)*, 2018. 7, 19

[23] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6391–6400, 2019. 1

[24] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5842–5850, 2017. 1, 3, 17

[25] Kristen Grauman et al. Ego4d: Around the World in 3,000 Hours of Egocentric Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 10

[26] Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 3, 8, 19

[27] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019. 4

[28] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 4

[29] Di Hu, Feiping Nie, and Xuelong Li. Deep multimodal clustering for unsupervised audiovisual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9248–9257, 2019. 4

[30] Deng Huang, Wenhao Wu, Weiwen Hu, Xu Liu, Dongliang He, Zhihua Wu, Xiangmiao Wu, Mingkui Tan, and Errui Ding. Ascnet: Self-supervised video representation learning with appearance-speed consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8096–8105, 2021. 2, 4

[31] Yuqi Huo, Mingyu Ding, Haoyu Lu, Zhiwu Lu, Tao Xiang, Ji-Rong Wen, Ziyuan Huang, Jianwen Jiang, Shiwei Zhang, Mingqian Tang, Songfang Huang, and Ping Luo. Self-supervised video representation learning with constrained spatiotemporal jigsaw. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021. 4

[32] Ashraful Islam, Chun-Fu Richard Chen, Rameswar Panda, Leonid Karlinsky, Richard Radke, and Rogerio Feris. A broad study on the transferability of visual representations with contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8845–8855, 2021. 1, 2

[33] Simon Jenni, Givi Meishvili, and Paolo Favaro. Video representation learning by recognizing temporal transformations. In *European Conference on Computer Vision (ECCV)*, pages 425–442, 2020. 4

[34] Longlong Jing, Xiaodong Yang, Jingen Liu, and Yingli Tian. Self-supervised spatiotemporal feature learning via video rotation prediction. *arXiv preprint arXiv:1811.11387*, 2018. 4

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

[36] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8545–8552, 2019. 4

[37] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1920–1929, 2019. 1

[38] Okan Köpüklü, Xiangyu Wei, and Gerhard Rigoll. You only watch once: A unified CNN architecture for real-time spatiotemporal action localization. *arXiv preprint arXiv:1911.06644*, 2019. 7, 19

[39] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *Advances in Neural Information Processing Systems*, volume 31, 2018. 4

[40] Klemen Kotar, Gabriel Ilharco, Ludwig Schmidt, Kiana Ehsani, and Roozbeh Mottaghi. Contrasting contrastive self-supervised representation learning pipelines. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9949–9959, 2021. 1

[41] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011. 1, 2

[42] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 513–528, 2018. 1

[43] Yuanze Lin, Xun Guo, and Yan Lu. Self-supervised video representation learning with meta-contrastive network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8239–8249, 2021. 2, 4

[44] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video cloze procedure for self-supervised spatio-temporal learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11701–11708, 2020. 4, 16

[45] Shuang Ma, Zhaoyang Zeng, Daniel McDuff, and Yale Song. Active contrastive learning of audio-visual video representations. In *International Conference on Learning Representations*, 2021. 4

[46] Pascal Mettes, Jan C van Gemert, and Cees G M Snoek. Spot on: Action localization from pointly-supervised proposals. In *European conference on computer vision*, pages 437–453. Springer, 2016. 7

[47] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision (ECCV)*, pages 527–544, 2016. 1, 4

[48] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 4, 16

[49] Alejandro Newell and Jia Deng. How useful is self-supervised pretraining for visual tasks? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[50] 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. In *International Conference on Learning Representations*, 2021. 10, 23

[51] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11205–11214, 2021. 1, 4, 16

[52] Mandela Patrick, Yuki M. Asano, Polina Kuznetsova, Ruth Fong, João F. Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal self-supervision from generalized data transformations. In *International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 4, 16

[53] Chen Peihao, Huang Deng, He Dongliang, Long Xiang, Zeng Runhao, Wen Shilei, Tan Mingkui, and Gan Chuang. Rspnet: Relative speed perception for unsupervised video representation learning. In *The AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 1, 2, 4, 16

[54] AJ Piergiovanni, Anelia Angelova, and Michael S Ryoo. Evolving losses for unsupervised video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 133–142, 2020. 1

[55] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6964–6974, 2021. 1, 2, 4, 5

[56] Adria Recasens, Pauline Luc, Jean-Baptiste Alayrac, Luyu Wang, Florian Strub, Corentin Tallec, Mateusz Malinowski, Viorica Pătrăucean, Florent Altché, Michal Valko, et al. Broaden your views for self-supervised video learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1255–1265, 2021. 2, 5

[57] Mert Bulent Sariyildiz, Yannis Kalantidis, Diane Larlus, and Karteek Alahari. Concept generalization in visual representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9629–9639, 2021. 1

[58] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1010–1019, 2016. 3, 17

[59] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3, 6, 17, 18, 19, 23

[60] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In *European Conference on Computer Vision (ECCV)*, pages 510 – 526, 2016. 3, 8, 19

[61] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 1, 2, 3, 17

[62] Chen Sun, Arsha Nagrani, Yonglong Tian, and Cordelia Schmid. Composable augmentation encoding for video representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8834–8844, 2021. 4

[63] Tomoyuki Suzuki, Takahiro Itazuri, Kensho Hara, and Hirokatsu Kataoka. Learning spatiotemporal 3d convolution with video order self-supervision. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 4

[64] Li Tao, Xueting Wang, and Toshihiko Yamasaki. Self-supervised video representation learning using inter-intra contrastive framework. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2193–2201, 2020. 4

[65] Li Tao, Xueting Wang, and Toshihiko Yamasaki. Pretext-contrastive learning: Toward good practices in self-supervised video representation leaning. *arXiv preprint arXiv:2010.15464*, 2021. 2, 4, 16

[66] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450–6459, 2018. 4

[67] Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisin Mac Aodha. Benchmarking representation learning for natural world image collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12884–12893, 2021. 1

[68] Bram Wallace and Bharath Hariharan. Extending and analyzing self-supervised learning across domains. In *European Conference on Computer Vision*, pages 717–734. Springer, 2020. 2

[69] Guangting Wang, Yizhou Zhou, Chong Luo, Wenxuan Xie, Wenjun Zeng, and Zhiwei Xiong. Unsupervised visual representation learning by tracking patches in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2, 4, 16

[70] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4006–4015, 2019. 4

[71] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In *European Conference on Computer Vision (ECCV)*, pages 504–521, 2020. 4

[72] Jinpeng Wang, Yuting Gao, Ke Li, Yiqi Lin, Andy J Ma, Hao Cheng, Pai Peng, Rongrong Ji, and Xing Sun. Removing the background by adding the background: Towards background robust self-supervised video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[73] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8052–8060, 2018. 7

[74] Fanyi Xiao, Joseph Tighe, and Davide Modolo. Modist: Motion distillation for self-supervised video representation learning. *arXiv preprint arXiv:2106.09703*, 2021. 1

[75] 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)*, pages 10334–10343, 2019. 1, 4

[76] Ceyuan Yang, Yinghao Xu, Bo Dai, and Bolei Zhou. Video representation learning with visual tempo consistency. *arXiv preprint arXiv:2006.15489*, 2020. 1, 2, 4, 5

[77] Xingyi Yang, Xuehai He, Yuxiao Liang, Yue Yang, Shanghang Zhang, and Pengtao Xie. Transfer learning or self-supervised learning? a tale of two pretraining paradigms. *arXiv preprint arXiv:2007.04234*, 2020. 1

[78] Ting Yao, Yiheng Zhang, Zhaofan Qiu, Yingwei Pan, and Tao Mei. Seco: Exploring sequence supervision for unsupervised representation learning. In *AAAI*, volume 2, page 7, 2021. 4

[79] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. Video playback rate perception for self-supervised spatio-temporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6548–6557, 2020. 4

[80] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019. 1

[81] Huaidong Zhang, Xuemiao Xu, Guoqiang Han, and Shengfeng He. Context-aware and scale-insensitive temporal repetition counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 7, 19

[82] Yujia Zhang, Lai-Man Po, Xuyuan Xu, Mengyang Liu, Yexin Wang, Weifeng Ou, Yuzhi Zhao, and Wing-Yin Yu. Contrastive spatio-temporal pretext learning for self-supervised video representation. In *Proceedings of the AAAI Conference on Artificial Intelligenc*, 2022. 4

Appendix

In Appendix A, we provide details of the video self-supervised models we use in our evaluation study. Appendix B provides details on the experimental setup for each of our downstream sensitivity factors. We also show correlation plots between current benchmarks and the experimental results for each sensitivity factor in Appendix C. Feature similarities between supervised pre-training and each self-supervised pre-training method are shown in Appendix D. In Appendix E, we describe domain difference between the downstream video datasets we use and the attributes we use to characterize this difference. Finally, we also report results of some additional experiments in Appendix F and Appendix G that we did not have room for in the main paper.

A Details of the Evaluated Self-Supervised Models

We use a variety of different self-supervised methods in our paper, here we describe each method:

MoCo [10] is a contrastive learning method proposed for representation learning in images. Positives are created by performing different spatial augmentations on a video. Negatives are other videos. To obtain negatives beyond the current batch, MoCo proposes a momentum encoder which maintains a queue of momentum-updated data samples from previous batches.

SeLaVi [4] views the audio and visual modalities as different augmentations of a video and learns with a cross-modal clustering pretext task.

VideoMoCo [51] extends MoCo to the temporal domain. It does this with an adversarial dropout augmentation which removes the frames the model considers most important. With the contrastive learning loss, the model learns invariance to this adversarial frame dropout alongside the spatial augmentations used in MoCo.

Pretext-Contrast [65] combines the pretext task approach with contrastive learning. As its pretext task it uses video cloze procedure [44] where the goal is to predict which augmentations have been applied to a video clip. For the contrastive learning objective different temporal shifts, *i.e.* distinct clips from the same video, are considered.

RSPNet [53] also combines pretext and contrastive tasks, with a focus on video speed. The pretext task is to predict the relative difference in speed between two versions of the same video, while the contrastive task creates extra positives and negatives by augmenting videos with different speeds along with the spatial augmentations.

AVID-CMA [48] is a multi-modal contrastive learning method which uses audio in addition to the visual modality. It first uses cross-modal contrastive learning where the one modality is used as the positives and the other as the negatives. Then it uses within modality contrastive learning where additional positives which have high audio and visual similarity are sampled.

CtP [69] performs self-supervised learning through a "catch the patch" pretext task. The goal in this task is to predict the trajectory of an image patch which is resized and moved through a sequence of video frames.

TCLR [14] is a contrastive method which encourages features to be distinct across the temporal dimension. It does this by using clips from the same video as negatives. Therefore, instead of encouraging invariance to temporal shift as other methods to, it encourages the model to be able to distinguish between different shifts. It also uses an extensive set of spatial augmentations.

GDT [52] is a multi-modal contrastive method which composes a series of different augmentations and encourages model to learn invariance to some and learns to distinguish between others. We use the best performing version of GDT which encourages invariance to spatial augmentations, the audio

Table 5: **Pre-training differences of our evaluated self-supervised methods.** While all models are pre-trained with the same backbone and dataset, there are differences in how many epoches they were trained for, the batch size and number of frames they use and the spatial and temporal augmentations they are encouraged to be invariant to.

Method	Extra Modality	Epochs	Batch Size	Num Frames	Spatial Augmentations						Temporal Augmentations		
					Random Crop	Horiz. Flip	Grayscale	Color Jitter	Gaussian Blur	Scaling	Shift	Reversal	Speed
MoCo		200	128	16	✓	✓	✓	✓			✓		
SeLaVi	Audio	200	1024	30	✓	✓							
VideoMoCo		200	128	32	✓	✓	✓	✓					
Pretext-Contrast		200	16	16	✓	✓	✓	✓	✓		✓		
RSPNet		200	64	16	✓			✓	✓		✓		✓
AVID-CMA	Audio	400	256	16	✓	✓		✓		✓			
CtP		90	32	16									
TCLR		100	40	16	✓	✓	✓	✓		✓			
GDT	Audio	100	512	30	✓	✓		✓				✓	
Supervised		45	32	16	✓	✓					✓		

and visual modalities and temporal reversal, while encouraging the model to distinguish between different temporal shifts.

While all models are pre-trained on Kinetics-400 and use an R(2+1)D-18 backbone with 112x112 spatial input size, there are some smaller differences in how the models are trained. Due to the computational cost of training these models we download publicly available models or obtain them from the authors, therefore we cannot control for these smaller differences in the pre-training set up. These differences include number of pre-training epochs, batch size, number of video frames used and spatial and temporal augmentations. We list these differences in Table 5.

B Downstream Experimental Details

B.1 Downstream Domain

In Section 3 we investigate to what extent self-supervised methods learn features applicable to action recognition in any domain. Here we explain the datasets, splits and training details we use to do this.

Datasets We report our experiments on the following datasets:
UCF-101 [61] is currently one of the most widely used datasets for evaluating video self-supervised learning models. It consists of YouTube videos from a set of 101 coarse-grained classes with a high overlap with actions in Kinetics-400. We use the first standard split proposed in the original paper [61] containing 9,537 training and 3,783 testing samples for the 101 action classes.
NTU-60: [58] consists of daily human actions captured in a controlled lab setting with a fixed number actors. Although it has some overlap with Kinetics-400 actions, it is quite different visually due to the setting. We use the cross-subject protocol proposed in [58] to split the data into 40,320 training and 16,560 testing samples for 60 action classes.
Gym-99. We use FineGym version $v1.0$ [59] which is a dataset of fine-grained actions constructed from recorded gymnastic competitions. We use the Gym 99 subset which contains 99 action classes with 20,484 and 8,521 samples in the train and test sets respectively.
SS-v2: [24] is a crowdsourced collection of first-person videos aimed to instill common-sense understanding. It differs significantly with respect to Kinetics-400 in terms of visual appearance and point-of-view. We use the original dataset splits from [24] containing 168,913 training and 24,777 testing samples for 174 action classes.
EPIC-Kitchens-100: [13] is a large-scale egocentric dataset consisting of daily actions performed in a kitchen. It has annotations for verbs (97) and nouns (300) and the action is defined a tuple of these. Like SS-v2, EK-100 also differs significantly from Kinetics-400 in terms of visual appearance and point-of-view. We use standard splits from [13] containing 67,217 samples in training set and 9,668 in the validation set. In the main paper we only aim to recognize the 97 verb classes, we provide results for the noun and action recognition tasks in Appendix G.

Training Details During training, we sample a random clip from each video of 32 frames with standard augmentations *i.e.* a random multi-scale crop of size 112x112 and color jittering. We train with the Adam optimizer. The learning rates, scheduling and total number of epochs vary across datasets and are shown in Table 6. However, each model is trained with the same hyper-parameters

Table 6: **Training details** of finetuning and linear evaluation on various downstream datasets. Learning rate is scheduled using a multip-step scheduler with $\gamma = 0.1$ at corresponding steps for each dataset. We train all the models with same hyperparameters for the corresponding dataset.

Dataset	Finetuning				Linear Evaluation			
	Batch Size	Learning rate	Epochs	Steps	Batch Size	Learning rate	Epochs	Steps
UCF-101	32	0.0001	160	[60,100,140]	64	0.01	100	[40,80]
NTU-60	32	0.0001	180	[90, 140, 160]	64	0.01	120	[40,80,100]
Gym-99	32	0.0001	160	[60,100,140]	64	0.01	120	[40,80,100]
SS-v2	32	0.0001	45	[25, 35, 40]	64	0.01	40	[20,30]
EK-100	32	0.0025	30	[20, 25]	32	0.0025	30	[20, 25]
K-400	-	-	-	-	64	0.01	40	[10,20,30]

for the corresponding dataset. For inference, we use 10 linearly spaced clips of 32 frames each. For each frame we take a center crop which is resized to 112x112 pixels. To calculate the action class prediction of a video, we take the mean of the predictions from each clip and report top-1 accuracy.

B.2 Downstream Samples

In Section 4 we measure how sensitive current video self-supervised models are to the amount of downstream samples. We do this by varying the size of the training data starting from 1000 examples and doubling it until we reach the full train set. We use the same data splits as in the downstream domain experiments, explained in Appendix B.1, and sample a subset of video clips from the respective train sets. We use the same random subset across the different models to make the comparison fair. For each dataset, we use same training and testing procedure as the downstream domain experiments, explained in Appendix B.1 and Table 6.

B.3 Downstream Actions

In Section 5 we measure how benchmark-sensitive current video self-supervised models are to downstream actions. We do so by measuring performance on different subsets, defined in the FineGym dataset [59], which have increasing semantic similarity. We provide the details of Gym-99, Gym-288 and the four different subsets we use of Gym-99 below:

Gym-99 consists of 29k video clips of 99 different actions across the four different gymnastic events in FineGym: Vault, Floor Exercise, Balance Beam and Uneven Bars. This is a relatively balanced subset of the full FineGym dataset with all actions having more than 80 occurrences. There are a total 20.5k training videos and 8.5k testing videos.

Vault is a subset of Gym 99 containing 1.5k videos of the 6 actions from the Vault event. The training split contains 1.0k examples and the testing split contains 0.5k examples.

Floor contains actions in the Floor Exercise event from Gym-99. It consists of 7.5k instances of over 35 actions with a split of 5.3k for training and 2.2k for testing.

FX-S1 is a subset of actions of leaps, jumps and hops from the Floor event in Gym-99. This subset of 11 actions contains a total of 2.6k video clips with 1.9k for training and 0.7k for testing.

UB-S1 contains 5k videos of 15 actions from the Uneven Bars event with a split of 3.5k for training and 1.5k for testing. The actions consist of different types of circles around the bars.

Gym-288 is a long-tailed version of Gym 99 containing 32k videos with 22.6K training and 9.6K testing samples. It adds 189 infrequent classes to the 99 classes in Gym 99, where actions can have as little as 1 or 2 instances in training. This results in a total of 288 action classes from the four different gymnastic events.

We follow the same training and evaluation procedure as that for finetuning Gym-99 in downstream domain training. In particular, for training we sample a random clip from each video of 32 frames with standard augmentations *i.e.* a random multi-scale crop of size 112x112 and color jitter. Each model is trained with the Adam optimizer using a learning rate of 0.0001 and multi-step scheduler with γ=0.1 at epochs [60, 100, 140] for 160 epochs. For inference, we use 10 linearly spaced clips of 32 frames each. For each frame we take a center crop which is resized to 112x112 pixels. To

calculate the action class prediction of a video, we take the mean of the predictions from each clip. For each subset, we compute accuracy per action class and report the mean over all action classes as in the original dataset [59].

B.4 Downstream Tasks

In Section 6 we investigate how sensitive self-supervised methods are to the downstream task and whether they generalize beyond action recognition. We provide details of the experimental setup used for each task below.

Spatio-temporal action detection. The goal of this task is to predict the bounding box of an actor in a given video clip, both spatially and temporally, along with the action class. We use the UCF101-24 benchmark which is a subset of UCF-101 with bounding box annotations for 3,207 videos from 24 action classes. We follow the implementation of Köpüklü *et al*. [38] using only a 3D-CNN branch for spatio-temporal action detection. We initialize the 3D backbone with the pre-trained, self-supervised R(2+1D)-18 models. A clip size of 16 frames is sampled from the video as the input with standard data augmentations *i.e*. horizontal flipping, random scaling and random spatial cropping. Each model is trained using the Adam optimizer with an initial learning rate of 1e-4, weight decay of 5e-4 and batch size 64, for a total of 12 epochs. The learning rate is decayed using a multi-step scheduler with γ=0.5 at epochs [4,6,8,10]. For testing we also follow [38] and report video-mAP over all the action classes.

Repetition counting. The goal of the this task is to estimate the number of times an action repeats in a video clip. We use the UCFRep benchmark proposed by Zhang *et al*. [81], which is a subset of UCF-101. The dataset consists of 526 videos with 3,506 repetition number annotations. From the annotated videos, 2M sequences of 32 frames and spatial size 112x112 are constructed which are used as the input. We use the implementation from the original benchmark [81] with pre-trained R(2+1)D-18 models as the backbone networks. Each model is trained for 100 epochs with a batch size of 32 using the Adam optimizer with a fixed learning rate of 0.00005. For testing, we follow the protocol from [81] and report mean counting error.

Arrow-of-time. The goal of this task is to predict the direction (forward of backward) of the video. We closely follow the setup used by Ghodrati *et al*. [22]. The full UCF-101 dataset is used with two versions of each video, one normal and one reversed. During training, for each video, we sample 8 frames linearly with a random offset, with batch size of 12 and 112x112 center crops, number of epochs 10, learning rate of $1e^{-5}$. We do not use any augmentations or learning rate schedulers. During testing, we sample 8 frames linearly. We report top-1 binary classification accuracy.

Multi-label classification on Charades. Charades [60] is made up of videos of people recording casual everyday activities at their homes. Videos in Charades are longer than the other datasets we use and the goal is to recognize multiple different actions in each video. A per-class sigmoid output is used for multi-class prediction. We use the implementation of Feichtenhofer *et al*. [19][1] with the R(2+1)D-18 backbone. During training, we use 32 frames with a sampling rate of 8. Since this task requires longer temporal context, we observe that using more frames with higher sampling rate is beneficial. We use a spatial crop of 112x112 and augmentations such as random short-side scaling, random spatial crop and horizontal flip. We train for 57 epochs in total with a batch size of 16 and a learning rate of 0.0375 with multi-step scheduler with $\gamma = 0.1$ at epochs [41, 49]. During testing, following [19], we spatio-temporally max-pool predictions over 10 clips for a single video. We report mean average precision (mAP) across classes.

Action detection on AVA. AVA [26] consists of clips extracted from films. We use version v2.2 with bounding box annotations for spatio-temporal action detection of temporally fine-grained action classes. The goal of this task is to detect and predict action classes from proposals generated by off-the-shelf person detectors. We again use the implementation of [19] with the R(2+1)D-18 backbone. During training, we use 32 frames with a sampling rate of 2. We use spatial crop of 112x112 and augmentations such as random short-side scaling, random spatial crop, horizontal flip. We train for 20 epochs with learning rate of 0.1 with multi-step scheduler with $\gamma = 0.1$ at epochs [10, 15] and a batch size of 32. During testing, following [19], we use a single clip at the center of the video with 8 frames and sampling rate of 8. We report mean average precision (mAP) across the classes.

[1] https://github.com/facebookresearch/SlowFast

C Correlations of Downstream Performance

As observed from the results of Section 3, the performance for both UCF-101 finetuning and Kinetics-400 linear evaluation is not indicative of how well a self-supervised video model generalizes to different downstream domains, samples, actions and tasks. Here, we plot the performance of each pre-trained model for each downstream settings and show the correlation with UCF-101 finetuning and Kinetics-400 linear evaluation performances. The results are shown in Figs. 5 to 12. These plots further demonstrate that the correlations are overall low for each downstream factor *i.e.* domain, samples, actions and tasks, indicating that more thorough testing of video self-supervised methods is needed.

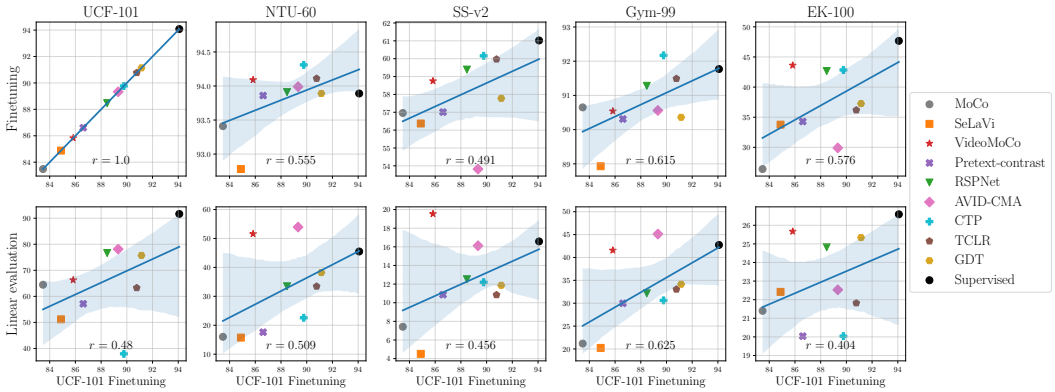

Figure 5: **Downstream domain against UCF-101 finetuning.** We plot the corelations between finetuning performance of video pre-training methods on UCF-101 and performances on finetuning and linear-evaluation on all downstream datasets.

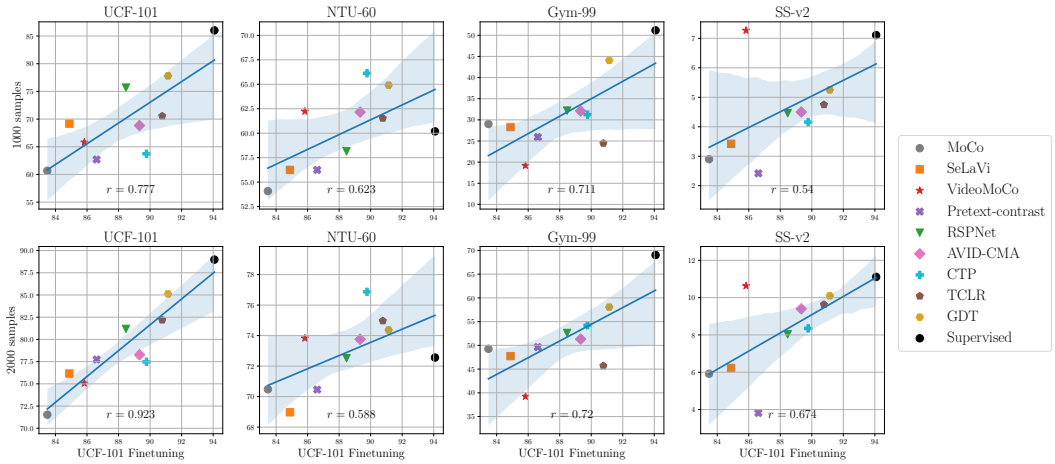

Figure 6: **Downstream samples against UCF-101 finetuning.** For the low data setting (1000-2000 samples), we plot the correlations of performance of video pre-training methods against that for UCF-101 finetuning.

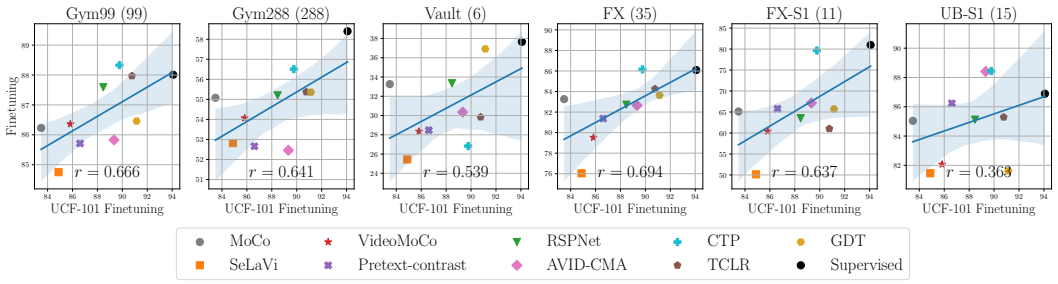

Figure 7: **Downstream actions against UCF-101 finetuning.** We plot the corelations of performances of video pre-training methods between UCF-101 finetuning and FineGym subsets.

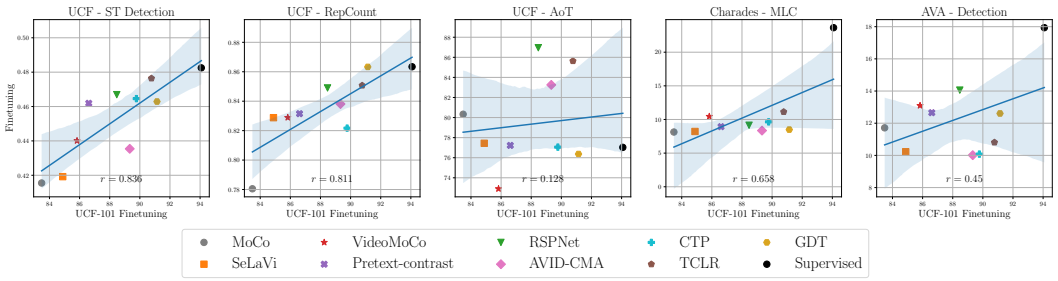

Figure 8: **Downstream tasks against UCF-101 finetuning.** We plot the corelations between performance on UCF-101 finetuning and other downstream tasks for the video pre-training methods.

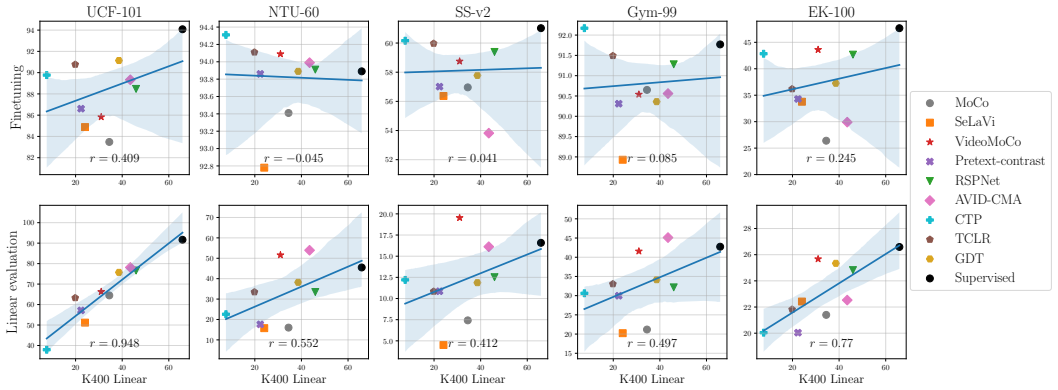

Figure 9: **Downstream domain against Kinetics-400 linear evaluation.** We plot the corelations between finetuning performance of video pre-training methods on Kinetics-400 linear-evaluation and performances on finetuning and linear-evaluation on all downstream datasets.

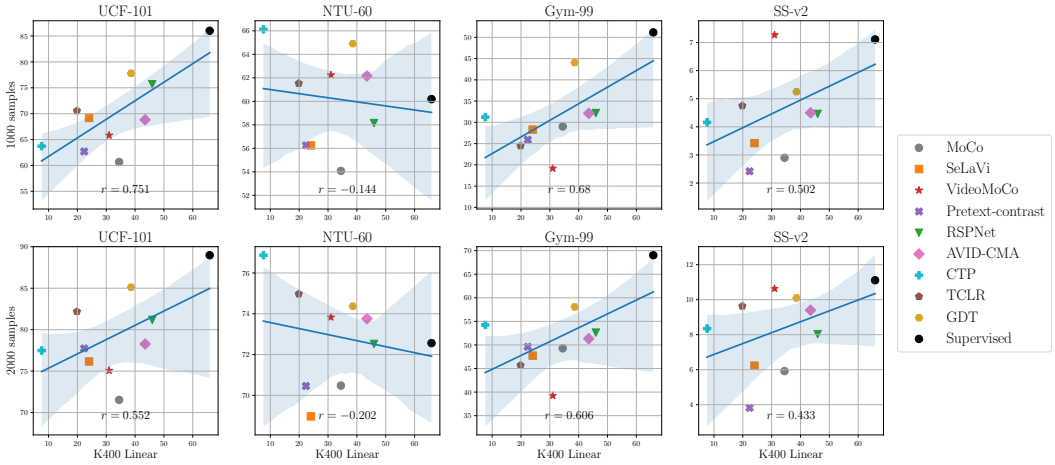

Figure 10: **Downstream samples against Kinetics-400 linear evaluation.** For the low data setting (1000-2000 samples), we plot the correlations of performance of video pre-training methods against that for Kinetics-400 linear-evaluation.

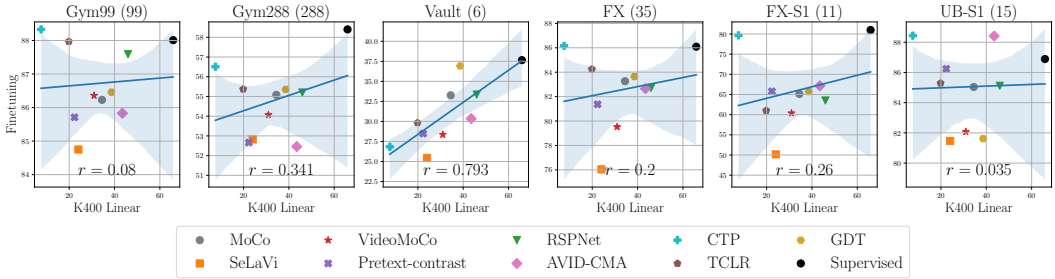

Figure 11: **Downstream actions against Kinetics-400 linear evaluation.** We plot the corelations of performances of video pre-training methods between Kinetics-400 linear-evaluation and FineGym subsets.

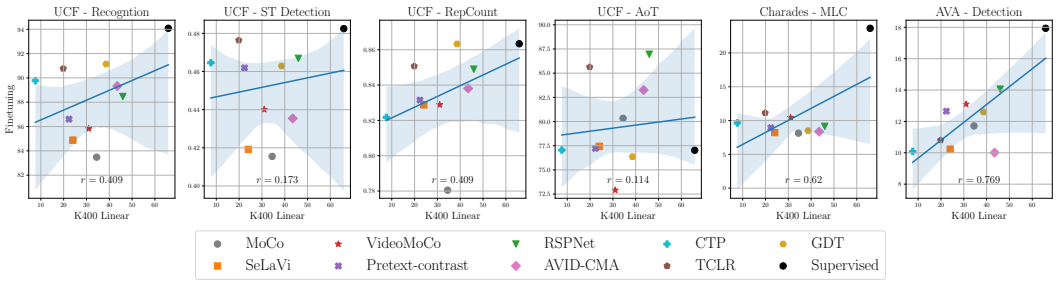

Figure 12: **Downstream tasks against Kinetics-400 linear evaluation.** We plot the corelations between performance on Kinetics-400 linear-evaluation and other downstream tasks for the video pre-training methods.

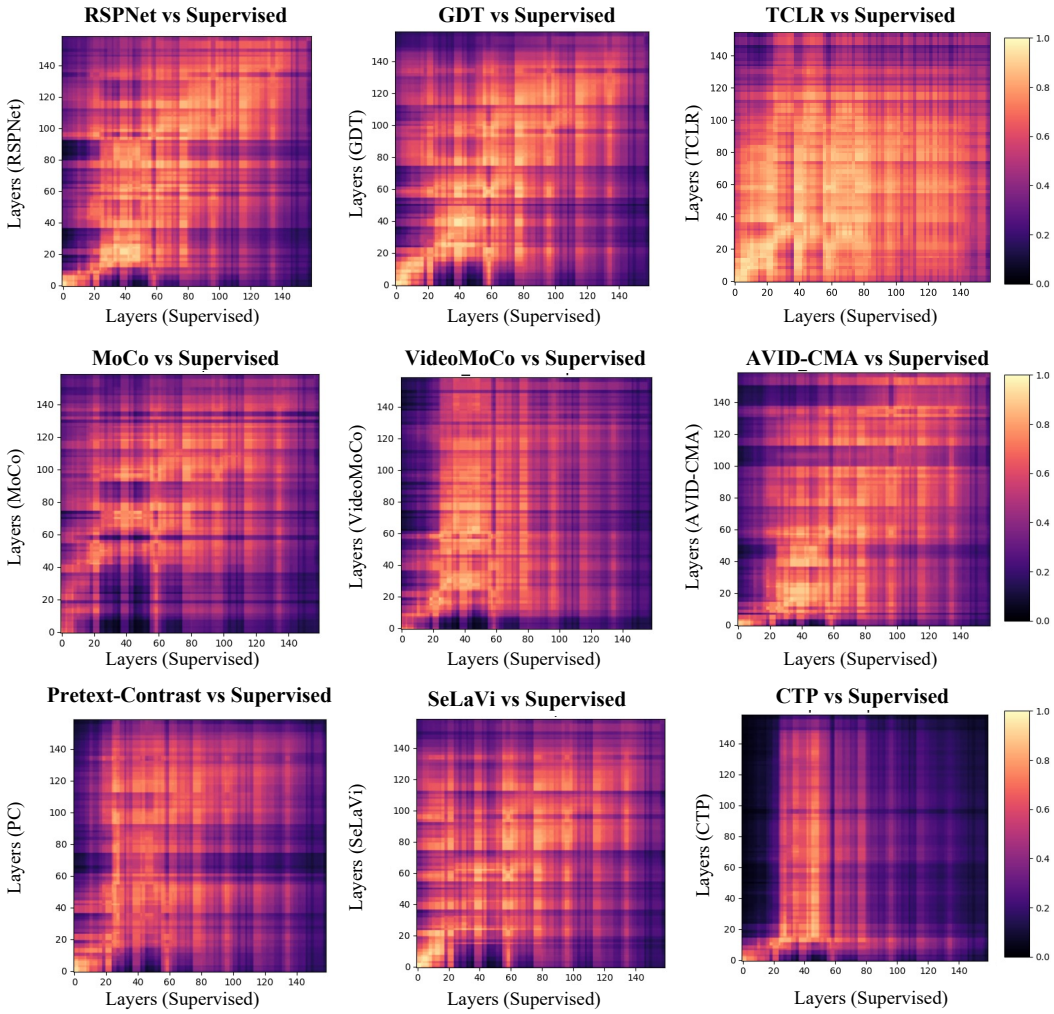

Figure 13: **Representation similarity** between features of self-supervised methods and supervised pre-training on Kinetics-400 validation set using centered kernel alignment. Features of contrastive methods are more closer to the features of supervised pretraining.

D Representation Similarity Matrices

We plot the the feature similarity on Kinetics validation set using centered kernel alignment [50] between supervised pre-training and our evaluated self-supervised pre-training methods in Fig. 13. We showed a subset of these plots in Fig. 4, here we show the feature similarity for all the self-supervised models we used in our experiments.

E Downstream Dataset Attributes

We define several attributes in Section 2.1 in order to characterize differences in domain between the downstream datasets and the Kinetics-400 pre-training dataset in Fig. 2. We provide detailed radar plots in Fig. 14 with axes labeled with relevant values for each attribute. The attributes *Point-of-view* and *Environment* are defined qualitatively based on the contents of the target dataset. Examples of videos from each of the datasets are shown in Fig. 15. We can see that FineGym [59] consists of videos of Olympic gymnastic events. Thus, we label it as *stadium* for environment and *third-person* for point-of-view. On the radar plots, we order environment in descending order of variability contained in a given dataset. Kinetics-400 is placed near the origin as it has much higher variability

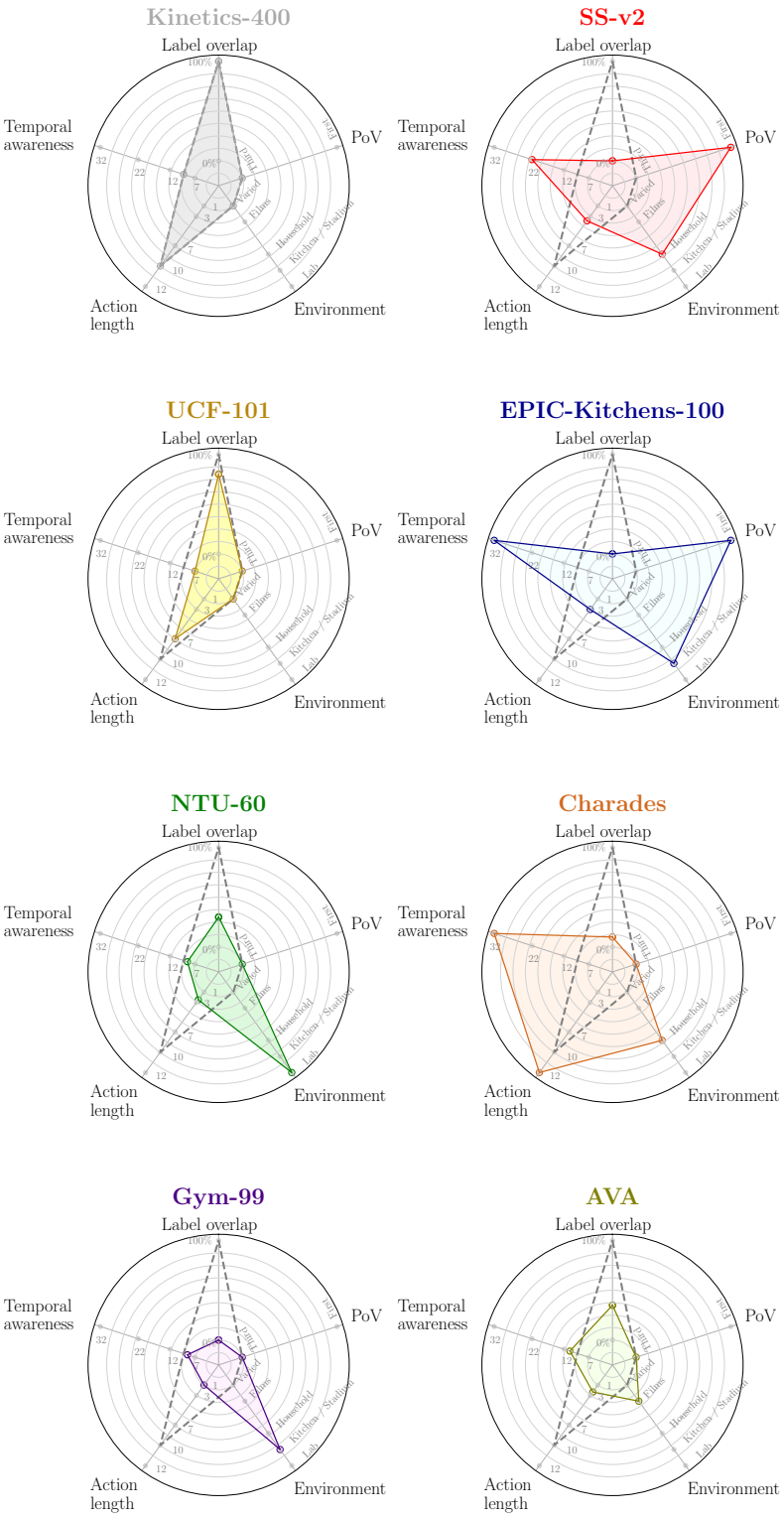

Figure 14: **Radar plots with details**. The radar plots contain details of the values along the axis for every attribute for the datasets we use in this study.

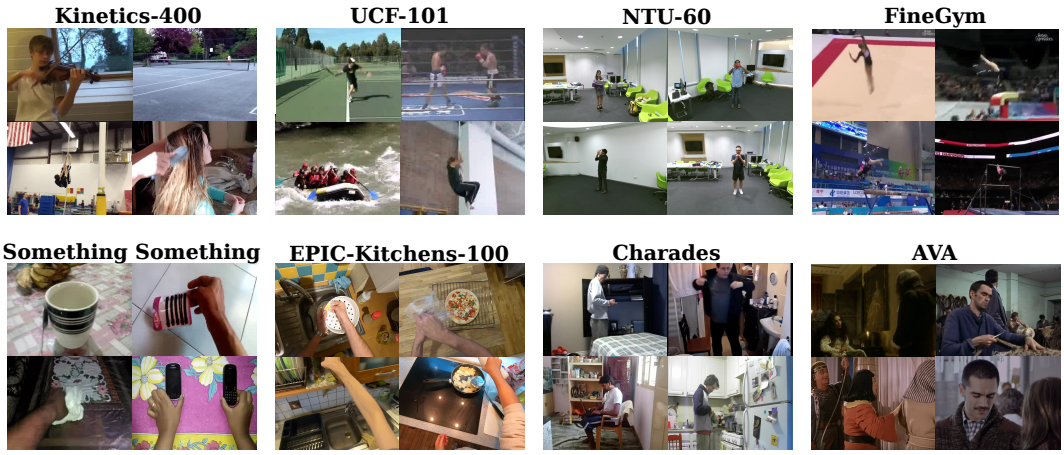

Figure 15: Example video frames from the Kinetics-400 pre-training dataset and the 7 different downstream datasets we consider. Note the differences in the capture setting and point-of-view across these datasets.

than NTU-60 for example, which is captured in a controlled lab setting. *Action length* is the average duration of the actions in each of the datasets.

We quantify *temporal awareness* as the minimum number of frames (temporal context) required to best recognize the action. We do this by finetuning R(2+1)D with weights initialized from supervised pre-training on Kinetics-400 and we denote temporal awareness (τ) as:

$$\tau = \arg \min_{t \in \{1,2,\ldots,N\}} \left[\left(100 \times \frac{f_{t+1} - f_t}{f_t} \right) < \alpha \right] \tag{1}$$

where α is chosen to be 1. This means τ indicates the number of frames after which relative improvement in performance is lesser than α, *i.e.* when the performance has plateaued. Fig. 16 shows the top-1 action recognition performance against increasing number of frames for each of our downstream datasets. We use bilinear interpolation to estimate performance at given number of frames beyond those that we experiments with. For example, using our method to compute temporal awareness, the performance for UCF-101 plateaus at 7 frames while that for EK-100 plateaus at 32 frames indicating that EK-100 needs much larger temporal context for recognition while UCF-101 may suffice with a shorter temporal context.

Label overlap is the amount of actions which are present in both the downstream dataset and the pretraining dataset (Kinetics-400). We quantify this by matching identical actions as well as manually checking for reworded versions of the same action class. For example, "head massage" in UCF-101 has a corresponding action "massaging person's head" in Kinetics-400. In NTU-60 action class "brushing teeth" has a matching action "brushing teeth" in Kinetics-400.

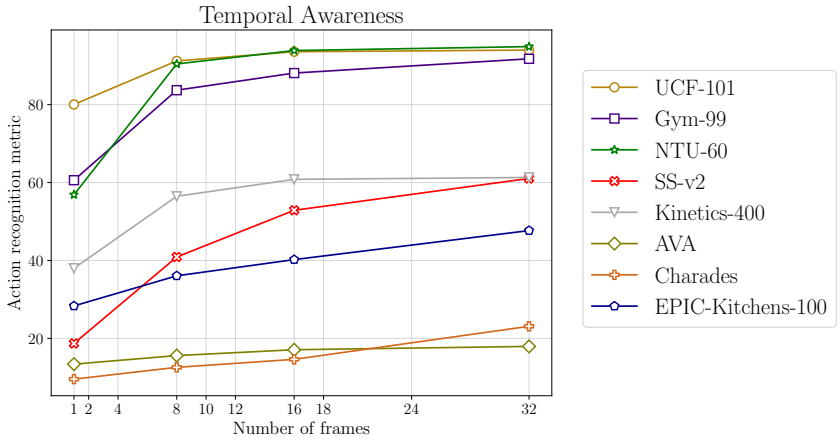

Figure 16: **Temporal awareness**. Illustrating the effect of temporal awareness (increasing temporal-context) on the action recognition performance using a standard 3D-CNN for different action datasets.

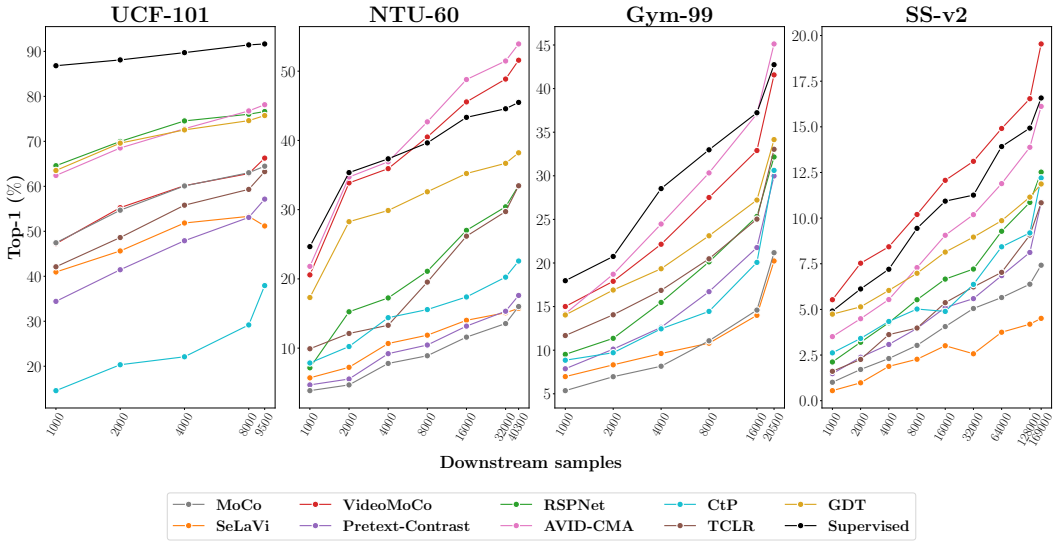

Figure 17: **Linear evaluation for Downstream Samples.** Comparison of video self-supervised learning methods using varying number of samples on linear evaluation for four downstream datasets. Rank changes are less significant with increasing sample size.

Table 7: **Ablation on Verb and Noun Recognition.** On EPIC-Kitchens-100, we show results for noun, verb and action recognition. Colors denote relative rankings across methods for each dataset, ranging from low high. Most pre-training methods struggle on noun and action recognition and have little correlation with verb recognition.

	EPIC-Kitchens-100		
Pre-training	Verb	Noun	Action
None	25.7	6.9	1.8
MoCo	26.4	13.9	6.9
SeLaVi	33.8	12.1	5.9
VideoMoCo	43.6	15.1	9.4
Pretext-contrast	34.3	11.4	5.6
RSPNet	42.7	18.7	11.7
AVID-CMA	29.9	8.7	3.6
CtP	42.8	12.0	7.8
TCLR	36.2	11.7	5.8
GDT	37.3	15.5	8.4
Supervised	47.7	24.5	16.0

F Linear Evaluation for Downstream Samples

In Section 4 we evaluated our pre-trained models with varying amounts of downstream samples for finetuning. In this section we provide the results for the same experiment but using linear-evaluation instead of finetuning. The results are shown in Fig. 17. We observe that rank changes are not significant between different sample sizes as they are for full finetuning., However similar to finetuning, supervised pretraining is dominant for low data setting as shown by the performance on NTU-60 and GYM-99 with 1000-4000 examples.

G Verb vs. Noun in Downstream Action Recognition

EPIC-Kitchens-100 [13] consists of noun and verb annotations for each video. An action is defined as a tuple of these. In the main paper, we report verb recognition performance across all experiments. In Table 7 we compare the performance on verb recognition to the performance on noun and action recognition. In general, performance is lower for noun and action recognition in comparison to verb recognition. This is likely due to the R(2+1)D-18 backbone being insufficient to model the complex actions found in EPIC-Kitchens-100. Interestingly, good performance on verb recognition is not a good indication that the model will perform well at noun or action recognition. Notably, some methods such as VideoMoCo and CtP perform well at verb recognition but struggle on noun recognition. RSPNet seems to perform reasonably well for both verb and noun recognition.