Benchmarking Self-supervised Learning for Spatio-temporal Representations (Supplementary)

Anonymous CVPR submission

Paper ID 3061

Here, we explain things in details aout pretext task, architecture setup, augmentations, provide some more results and include more visual analysis. We also include tables which we were not able to include in main paper due to space limitations. We also show our tables, experiments and findings at https://thecodeeagle.github.io/webb/.

- Section 1: We extend the main table and compare with previous state-of-the-art results.
- Section 2: We show additional CKA maps, results on HMDB51 dataset and more analysis on noise robustness. We added some tables for Knowledge distillation experiments that were promised in the main paper.
- Section 3: Pretext tasks explanation used in our analysis.
- Section 4: Training details about architectures, datasets, and , other hyperparameters.

1. Main Table

In this section, we firstly expand the Table 9 including more recent approaches. Initially, it was restricted to approaches with R21D backbone and pre-train dataset as K400. We include the different backbone and pre-training dataset information as well in this table. Knowledge distillation discussed in the main paper still outperforms recent as well as multi-modal approaches on UCF101 dataset (Table 1).

2. Additional Results

Here, we will talk about some additional results, which couldn't be put in main paper. We have shown results for some of the claims made in the paper. We also include more visualizations on noise robustness for different pretext tasks with different architectures with different severity levels.

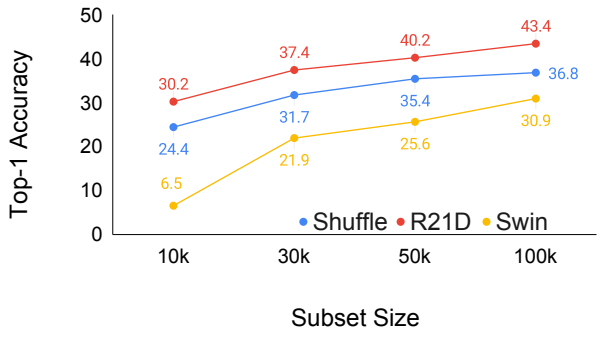

Figure 1. Left: dataset subset performance for three different architectures on RSPNet pretext task.

2.1. Pre-train dataset size

In Table 2, we extend results for different pretext tasks on HMDB51 dataset. Similar to UCF101, the scale in subset size doesn't reciprocate to gain in performance for all pretext tasks on HMDB51 dataset. From Fig. 1, we see that performance increase for R21D and Swin by a good margin, but, for ShuffleNet the gain is almost 1% only.

2.2. Training time

Inspecting the training time for Swin transformer on UCF101 dataset (Table 3), we see that performance saturates at 150 epochs in general. It suggests that VideoSwin needs more training time as compared to CNN architectures (R21D and ShuffleNet) which saturates mostly around 100 epochs. In general, similar to UCF101 performance increase with increase in training time across different pretext tasks for a fixed subset size on HMDB51 (Table 4).

2.3. Linear Probing vs Finetuning

In this section, we discuss the linear probing results for different pretext tasks and different architectures to justify the reason for choosing finetuning instead of linear probing. From Table 5, we can see that there's a performance de-

Approach	NxW/H	Venue	Backbone	Pre-training	UCF101	HMDB51
PacePred [34]	ECCV'20	16x112	R21D-18	K400	77.1	36.6
TempTrans [16]	ECCV'20	16x112	R3D-18	K400	79.3	49.8
STS [32]	TPAMI-21	16x112	R21D-18	K400	77.8	40.5
VideoMoCo [23]	CVPR'21	16x112	R21D-18	K400	78.7	49.2
RSPNet [5]	AAAI'21	16x112	R21D-18	K400	81.1	44.6
TaCo [3]	-	16x224	R21D-18	K400	81.8	46.0
TCLR [8]	CVIU'22	16x112	R21D-18	K400	88.2	60.0
CVRL† [25]	CVPR'21	32x224	R21D-18	K400	92.9	67.9
TransRank [10]	CVPR'22	16x112	R21D-18	K200	87.8	60.1
VideoMAE * [28]	NeurIPS'22	16x112	R21D-18	K400	76.2	45.4
Multi-Modal						
AVTS [20]	NeurIPS'18	25x224	I3D	K400	83.7	53.0
GDT [24]	-	32x112	R21D	IG65M	95.2	72.8
XDC [1]	NeurIPS'20	32x224	R21D	K400	84.2	47.1
Ours *	-	16x112	R21D-18	K400-50k	97.3	51.5

Table 1. Comparison with previous approaches pre-trained on K400 full set. Ours (* best performing) is RSPNet pretrained on 30k subset of K400. * reproduced results.

Epochs	VCOP	Rot	PRP	CVRL	TDL	RSPNet
10k	18.9	15.0	9.2	22.2	9.9	30.2
30k	19.3	11.7	11.5	25.0	10.1	37.3
50k	17.3	12.2	10.2	29.3	9.5	40.2

Table 2. Evaluation of different pretext tasks on different subset size on R21D network on HMDB51 dataset.

Epochs	Shuffle				R21D				Swin			
	10k	30k	50k	100k	10k	30k	50k	100k	10k	30k	50k	100k
50	59.1	66.3	68.1	68.9	66.8	71.1	75.0	77.2	-	40.4	44.9	52.0
100	60.3	67.6	68.7	69.0	69.5	75.2	76.1	80.0	37.2	44.3	49.6	58.5
150	61.8	66.7	69.4	69.7	69.5	76.6	76.5	78.8	37.9	46.2	50.7	61.3
200	61.5	68.2	68.5	69.9	69.6	76.6	77.4	78.3	36.8	46.3	52.5	61.5

Table 3. RSPNet with different subset size on ShuffleNet/R21D/VideoSwin on UCF101 dataset.

Epochs	VCOP	Rot	PRP	CVRL	TDL	RSPNet
50	19.6	15.6	6.8	22.8	8.1	28.9
100	20.2	16.1	8.2	26.2	8.8	30.0
150	17.6	12.4	7.9	27.5	9.2	32.5
200	19.4	10.7	10.2	28.4	9.5	32.8

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

Network	Task	RotNet	VCOP	PRP
Shuffle	LP	4.3	12.3	2.8
	FT	16.6	40.8	21.9
R21D	LP	2.7	12.2	4.6
	FT	41.2	51.5	46.2

Table 5. Downstream accuracy classification on UCF-101 dataset. FT: Finetuning LP: Linear Probing

crease of approximately 20% and 40% for ShuffleNet and R21D respectively. Thus, we perform finetuning for all of our analysis.

2.4. Centered Kernel Alignment : Maps

We extensively discuss the use of Centered Kernal Alignment (CKA) maps to visualize layer representations of a network, and to observe how these representations differ across varying architectures. All the maps discussed in the qualitative sections of the main paper have been added here. Few maps have been moved at the end of the paper. Figure 2 depicts the variation in representations for

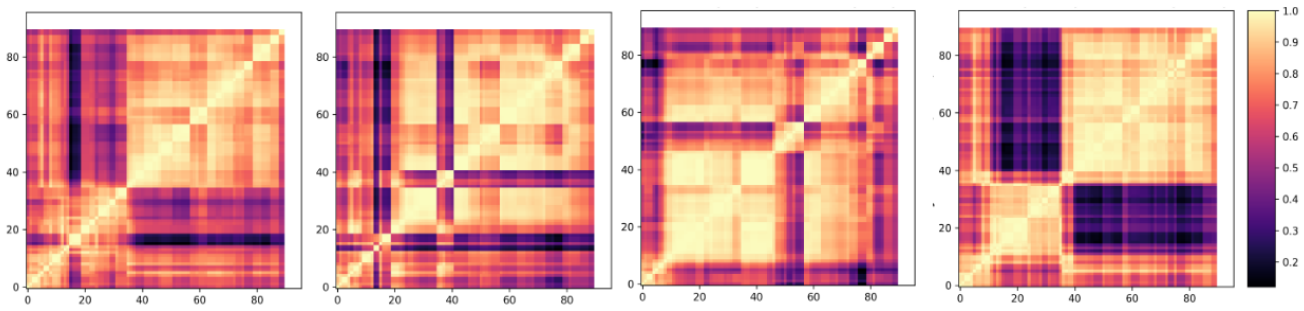

Figure 2. CKA maps for layer representations: 10k vs 10k, 30k vs 30k, 50k vs 50k, 100k vs 100k of R21D network on RSPNet pretext for all K-400 subsets (Left to right).

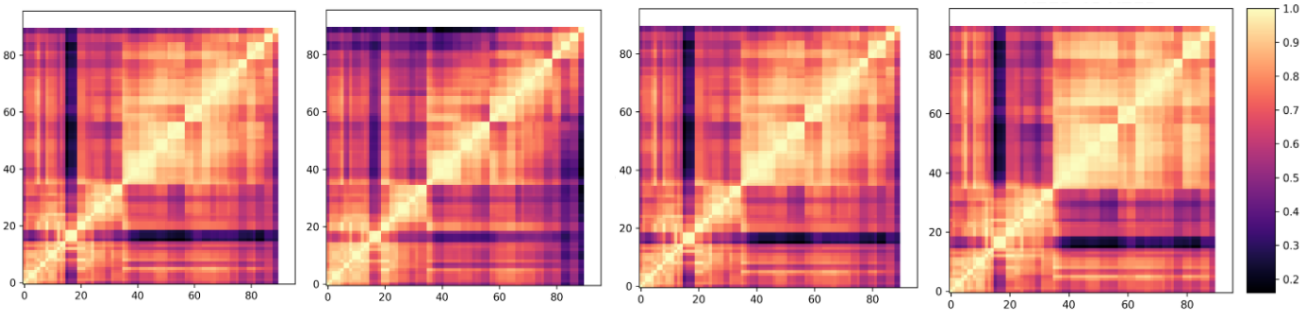

Figure 3. CKA maps for layer representations: 50 epochs vs 50 epochs, 100 epochs vs 100 epochs, 150 epochs vs 150 epochs, 200 epochs vs 200 epochs of R21D network on RSPNet pretext for K-400 10k subset (Left to right).

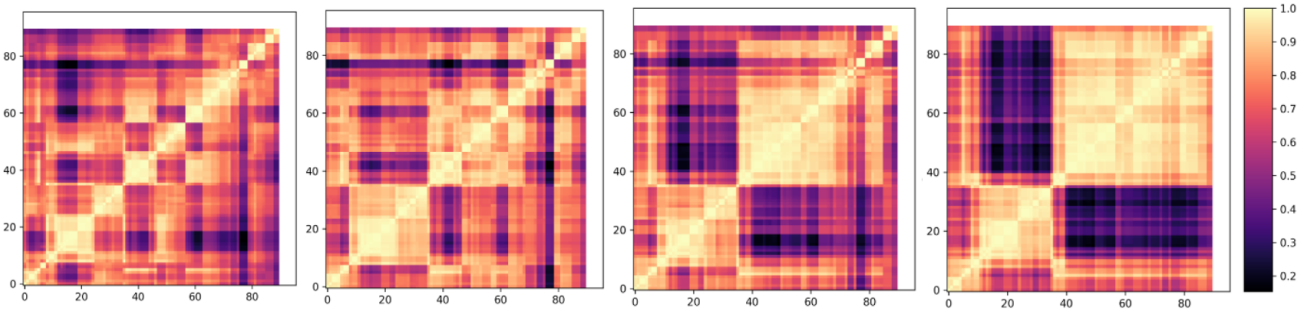

Figure 4. CKA maps for layer representations: 50 epochs vs 50 epochs, 100 epochs vs 100 epochs, 150 epochs vs 150 epochs, 200 epochs vs 200 epochs of R21D network on RSPNet pretext for K-400 100k subset (Left to right).

R21D network as the subset size of K400 for pretraining is increased from 10k to 100k. The network finetunes on UCF101 dataset, and the pretext task used for training was RSPNet. Figure 3 shows the emergence of block structures for R21D network trained on RSPNet for K400 10k subset, as the number of epochs for finetuning is increased from 50 to 200. Figure 4 depicts the same analysis as Figure 3, in this case however 100k subset of K400 was used for pretraining.

Figure 8 depicts the hidden representations of R21D network pretrained on different pretext tasks - RSPNet, PRP, VCOP, RotNet and CVRL. Here the 50k subset of K-400 was used for pretraining, while the network was finetuned on UCF-101. Figure 9 depicts the same for the ShuffleNet network.

Figures 10-13 depict the variation in features across different complexities for a network pretrained on the same pretext task. Figures 10 and 11 outline the contrast for the PRP pretext task- the former shows the same for R(2+1)D network while the latter for ShuffleNet. Similarly, Figures 12 and 13 portray the variation across 2,3,4 complexities for the RotNet pretext task, where Figure 12 shows the same

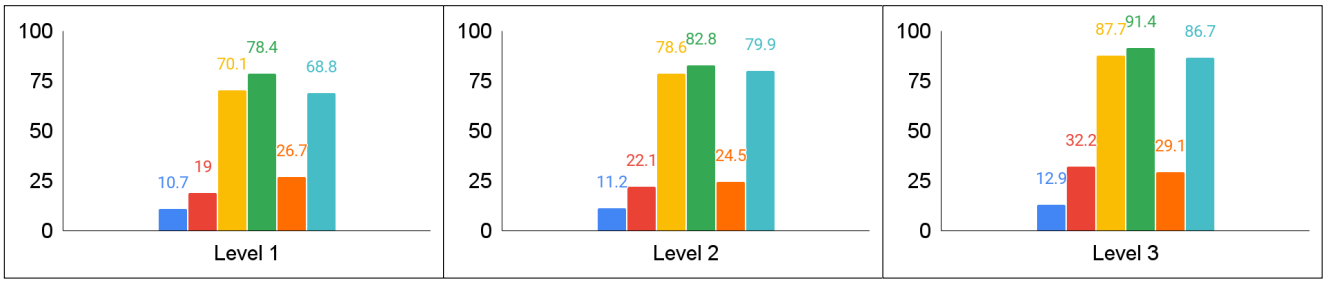

Figure 5. Relative decrease in performance at three different severity levels in increasing order from left to right. The pretext tasks is depicted by following colors - RotNet, VCOP, PRP, CVRL, TDL, RSPNet.

	RotNet	VCOP	PRP	CVRL	TDL	RSP
No Noise	41.2	51.5	46.2	61.2	31.7	78.0
Gaussian	40.9	47.0	14.6	12.7	28.0	16.7
Impulse	38.1	30.5	5.4	3.5	18.8	8.5
Shot	33.4	45.1	20.9	26.4	21.5	45.1
Speckle	34.7	43.9	14.4	13.1	24.7	27.0

Table 6. Analysis of all pretext tasks with noise severity level 1 on R21D network on UCF101 dataset.

Networks	Parameters	GFLOPs	Rot†	VCOP †	PRP†	RSPNet
ShuffleNet	4.59M	1.08	42.2	41.6	41.1	68.8
MobileNet	3.06M	1.12	38.0	40.0	37.4	63.1
SqueezeNet	1.89M	1.84	41.3	41.4	39.2	62.9
C3D	27.66M	77.22	57.7	54.5	58.1	67.6
R3D	14.36M	39.84	51.1	50.7	52.1	62.1
R(2+1)D	14.37M	42.96	46.9	56.8	58.9	78.0

Table 7. Comparison of FLOPs and trainable parameters for each network on UCF101 dataset. † - pretraining on Kinetics 700. Move to supple

behaviour for R21D network and Figure 13 for ShuffleNet.

Figure 14 illustrates CKA maps for networks pretrained on OOD dataset – for R21D pretrained on K400 for pretext tasks VCOP and CVRL respectively. The stark difference in semi-block structure of VCOP vs grid-like structure of CVRL can be observed.

2.5. Noise robustness

Table 6 shows performance of each pretext on each type of noise for severity level 1. Fig. 5 shows a relative decrease in performance for three different severity level on UCF101 dataset. RotNet is most robust across different severity levels and CVRL is the least. A clip sample for each noise is attached in the zip folder.

2.6. Network Parameters

We have shown the performance across different architectures in Table 6. ShuffleNet and R21D performs the best across small and medium capacity networks.

Network	Top@1	Top@5
Squeeze	15.9/38.5	37.6/56.5
Mobile	16.2/37.4	36.5/55.6
Shuffle	19.3/43.1	42.0/62.1
C3D	19.9/43.2	43.4/61.6
R3D	19.3/40.4	42.5/60.2
R21D	18.2/42.7	40.1/62.8

Table 8. Top K Clip Retrieval on HMDB51/UCF101 across different architectures for RSPNet.

2.7. Clip retrieval

In Table 8, we show clip retrieval across different architectures on HMDB51 and UCF101 dataset.

2.8. Knowledge Distillation

We employ knowledge distillation to evaluate how complementary information from different datasets can be used to train a student model that could take advantage of this in terms of performance gain and training time reduction.

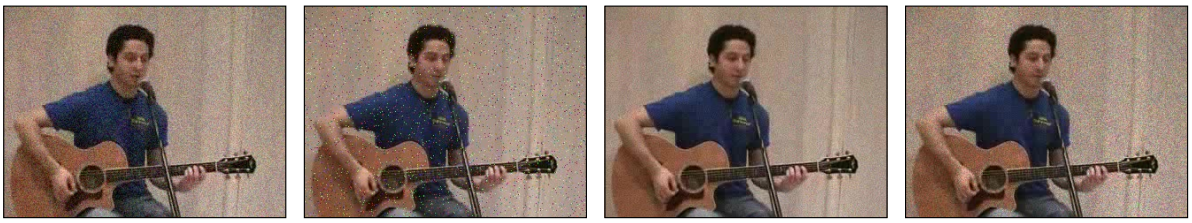

Figure 6. An example frame sample for each noise Gaussian, Impulse, Shot and Speckle respectively. Clips are provided in supplementary.

	S (T1)	T(T2)	Student
Non-Contrastive	RotNet	VCOP	61.1
Contrastive	CVRL	TDL	70.3

Table 9. KD across different Pretext Tasks. Teachers: ShuffleNet; Student: ShuffleNet. ST refers to student without pretraining

	K400 (T1)	SSV2(T2)	Student
RotNet	36.2	42.5	59.8
VCOP	50.4	59.7	67.6
CVRL	56.9	34.7	66.6
RSPNet	76.4	69.5	80.2

Table 10. KD OOD experiments on UCF101 dataset using R21D network.

TC↓	RotNet	VCOP	PRP
T1	20.1/48.3	41.6/**56.8**	24.2/38.9
T2	20.2/**58.3**	41.8/54.8	18.1/44.4
T3	16.6/41.2	40.6/55.6	21.9/46.2
S	**75.0**/56.6	**75.4**/43.5	**76.1/61.0**

Table 11. KD Complexity variation with different complexities as teachers (T1, T2, T3) for all three pretext tasks. TC: Task complexity. Results are shown on UCF101 with ShuffleNet/R21D as backbones.

We discuss Tables 9, 10 and 11in the main paper, and have added the results here for the same. We also mention the performance gain for out-of-distribution knowledge distillation for two pretext tasks- VCOP and RotNet in the paper, and we extend the same analysis here to CVRL and RSPNet. We notice the performance gain for all pretext tasks as compared to the individual teacher networks.

3. Pretext Tasks

In this section, we go through each pretext task in more detail that are used in our main work for analysis.

3.1. Spatial Transformation

Rotation Net [17] (RotNet) applies geometrical transformation on the clips. The videos are rotated by various angles and the network predicts the class which it belongs to. Since the clips are rotated, it helps the network to not converge to a trivial solution.

Contrastive Video Representation Learning [25] (CVRL) technique applies temporally coherent strong spatial augmentations to the input video. The contrastive framework brings closer the clips from same video and repels the clip from another video. With no labels attached, the network learns to cluster the videos of same class but with different visual content.

3.2. Temporal Transformation

Video Clip Order Prediction [35] (VCOP) learns the representation by predicting the permutation order. The network is fed N clips from a video and then it predicts the order from N! possible permutations.

Temporal Discriminative Learning [33] (TDL) In contrast to CVRL, TDL works on temporal triplets. It looks into the temporal dimension of a video and targets them as unique instances. The anchor and positive belongs to same temporal interval and has a high degree of resemblance in visual content compared to the negative.

3.3. Spatio-Temporal Transformation

Playback Rate Prediction [6] (PRP) has two branch, generative and discriminative. Discriminative focuses on the classifying the clip's sampling rate, whereas, generative reconstructs the missing frame due to dilated sampling. Thus, the first one concentrates on temporal aspect and second one on spatial aspect.

Relative Speed Perception Network [5] (RSPNet) applies contrastive loss in both spatial and temporal domain. Clips are samples from a same video to analyze the relative speed between them. A triplet loss pulls the clips with same speed together and pushes clips with different speed apart in the embedding space. To learn spatial features, InfoNCE

loss [31] is applied. Clip from same video are positives whereas clips from different videos are negatives.

4. Implementation Details

4.1. Architecture Details

Preliminary research has shown that 3D networks [13, 30] have outperformed 2D CNN variants on video recognition tasks. We looked into three types of capacity - small, medium and big on the basis of number of trainable parameters. The architecture details of all networks are mentioned in supplementary.

Small capacity networks: are resource efficient, implying they can be trained in larger batches within short span of time. The network selection is done on the basis of supervised training scores on Kinetics [18] and UCF101 [19]. ShuffleNet V1 2.0X [36] utilizes point-wise group convolutions and channel shuffling. SqueezeNet [15] reduces the filter size and input channels to reduce the number of parameters. MobileNet [26] has ResNet like architecture. With its depthwise convolution, there's a reduction in model size and the network can go more deep.

Medium capacity networks: Following the conventional 3D architectures for self-supervised learning approaches C3D, R21D and R3D are used in this study.

Big Capacity networks: We are the first to study the performance of self-supervised video representation learning on transformer based architectures. Comparing across three transformer architectures, ViViT [2] Timesformer [4] and MViT [11], we selected ViViT, because, firstly, it's a direct extension of ViT [9] from images to videos incorporating spatio-temporal attention, and, secondly, all these architectures have comparable performance.

Based on [19], we probed into the performance comparison of several versions of these architectures. We choose 3D-ShuffleNet V1 2.0X, 3D-SqueezeNet, and 3D-MobileNet V2 1.0X networks based on their performance on Kinetics and UCF-101 dataset

3D-ShuffleNet V1 2.0X [36]: It utilize point-wise group convolutions and channel shuffling and has 3 different stages. Within a stage, the number of output channel remains same. As we proceed to successive stage, the spatiotemporal dimension is reduced by a factor of 2 and the number of channels are increased by a factor of 2. V1 denotes version 1 of ShuffleNet and 2.0X denotes the 2 times number of channels compared to original configuration.

3D-SqueezeNet [15]: It uses different alteration to reduce the number of parameters as compared to the 2D version which employs depthwise convolution. Those three modifications are: 1) Change the shape of filters from 3x3 to 1x1, 2) Input channels to 3x3 filters is reduced, and, 3) to maintain large activation maps high resolution is maintained till deep layers.

3D-MobileNet V2 1.0X [26]: This network employs skip connections like ResNet architecture in contrast to version 1. It helps the model in faster training and to build deeper networks. There are also linear bottlenecks present in the middle of layers. It helps in two ways as we reduce the number of input channels: 1) With depthwise convolution, the model size is reduced, and 2) at inference time, memory usage is low. V2 denotes version 2 of mobilenet and 1.0X uses the original parameter settings.

The architectures of medium capacity networks are described as follows:

C3D [29]: This follows a simple architecture where two dimensional kernels have been extended to three dimensions. This was outlined to capture spatiotemporal features from videos. It has 8 convolutional layers, 5 pooling layers and 2 fully connected layers.

R3D [13]: The 2D CNN version of ResNet architecture is recasted into 3D CNNs. It has skip connections that helps make the gradient flow better as we build more deeper networks.

R(2+1)D [30]: In this architecture, 3D convolution is broken down into 2D and 1D convolution. 2D convolution is in spatial dimension and 1D convolution is along the temporal dimension. There are two benefits of this decomposition: 1) Increase in non-linearity as the number of layers have increased, and, 2) Due to factorization of 3D kernels, the optimization becomes easier.

4.2. Original and Noise Datasets

We have shown the examples of each dataset used in the paper in Fig. 7.

The test datasets have different number of videos for different levels and types of noises. For Gaussian noise, we manipulated all 3783 samples. For noise level 1, apart from Gaussian, we had roughly 400 samples and all other levels of severity, we have approximately 550 samples.

4.3. Pretext Tasks Configurations

Here, we briefly describe the configurations used in our training. For VCOP, RotNet and PRP, we just manipulated the type of augmentation from the original work. We applied Random Rotation, Resizing, Random Crop, Color Jittering and Random Horizontal Flipping to the input clip. CVRL has some extra data augmentation compare to the previous ones we mentioned. It includes grayscale and gamma adjustment as well. RSPNet also uses some temporal augmentation. For finetuning the augmentations are Resize and Center Crop for all the approaches.

The k-value for Momentum contrastive network is 16384 for RSPNet, it's 500 for TDL.

Figure 7. An example sample from each dataset.

4.4. Datasets

Here we discuss datasets in detail. We use Kinetics-400 (K400) [18] and Something-Something V2 [12] for our pre-training. For the downstream task evaluation, we perform our experiments on UCF-101 [27], HMDB-51 [21], and Diving48 [22]. Since, the pretraining and finetuning datasets are different, the performance variation will provide us a better picture about how much meaningful spatiotemporal features are learned by these networks. K400 has approximately 240k videos distributed evenly across 400 classes respectively. The approximate number of videos in finetuning datasets are: 1) UCF101-10k, 2) HMDB51-7k, and, 3)Diving48-16k. The datasets can be categorized into two ways:

Appearance-based: Kinetics, UCF101 and HMDB51 comes under this category [7, 14]. Kinetics videos length are generally 10s centered on human actions. It mainly constitutes singular person action, person-to-person actions and person-object action. For pre-training, we select a random subset of videos and maintain equal distribution from each class. Unless otherwise stated, pre-training is done on K400-50k subset for all experiments.

Temporal-based: In Kinetics, we can estimate the action by looking at a single frame [7, 14]. From Fig. 7, top two rows, we can see the person with a javelin and basketball. This information helps in class prediction. Looking at bottom two rows (SSv2 and Diving48 respectively), we can't describe the activity class until we look into few continuous frames. It shows that temporal aspect plays an important role for these datasets, that's why we categorize them into temporal-based datasets.

UCF-101 [27] : It's an action recognition dataset that spans over 101 classes. There are around 13,300 videos, with 100+ videos per class. The length of videos in this dataset varies from 4 to 10 seconds. It covers five types of categories: human-object interaction, human-human interaction, playing musical instruments, body motion and sports.

HMDB-51 [21] : The number of videos in this dataset is 7000 comprising 51 classes. For each action, at least 70 videos are for training and 30 videos are for testing. The actions are clubbed into five categories: 1) General facial actions, 2) Facial actions with body movements, 3) General body movements, 4) Body movements with object interaction, and, 5) Body movements for human interaction.

References

[1] 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. 2

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. Vivit: A video vision transformer. *ArXiv*, abs/2103.15691, 2021. 6

[3] 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. 2

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *ArXiv*, abs/2102.05095, 2021. 6

[5] 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. 2, 5

[6] 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. 5

[7] Jinwoo Choi, Chen Gao, Joseph C.E. Messou, and Jia-Bin Huang. Why can't i dance in the mall? learning to mitigate scene bias in action recognition. In *NeurIPS*, 2019. 7

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

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner,

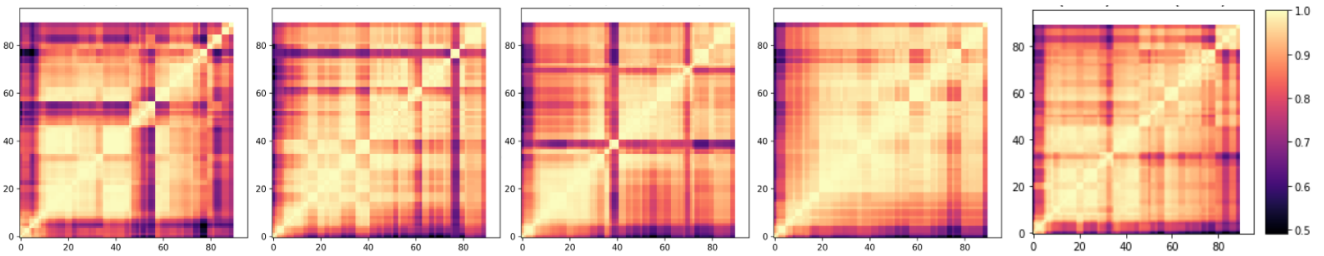

Figure 8. CKA maps for layer representations: RSPNet, PRP, RotNet, VCOP, CVRL of R21D network for K-400 50k subset (Left to right).

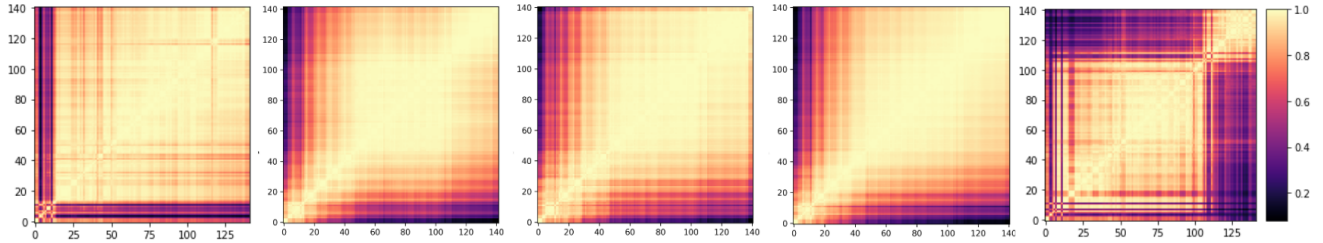

Figure 9. CKA maps for layer representations: RSPNet, PRP, RotNet, VCOP, CVRL of Shuffle network for K-400 50k subset (Left to right).

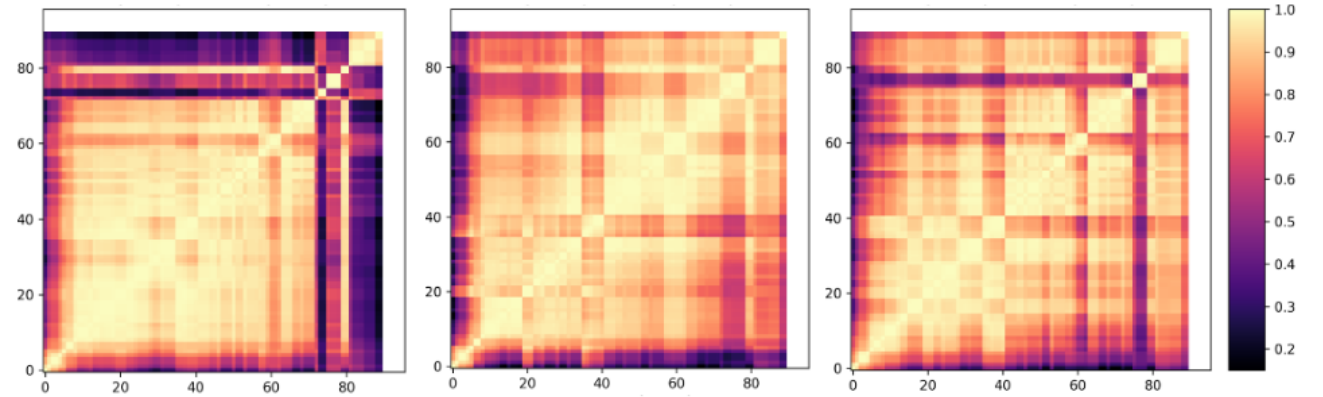

Figure 10. CKA maps for layer representations: Complexity 2,3,4 for PRP pretext, Network: R21D (Left to right).

Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2021. 6

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

[11] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *ArXiv*, abs/2104.11227, 2021. 6

[12] 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. 7

[13] K. Hara, H. Kataoka, and Y. Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 3154–3160, 2017. 6

[14] De-An Huang, Vignesh Ramanathan, Dhruv Mahajan, Lorenzo Torresani, Manohar Paluri, Li Fei-Fei, and Juan Carlos Niebles. What makes a video a video: Ana-

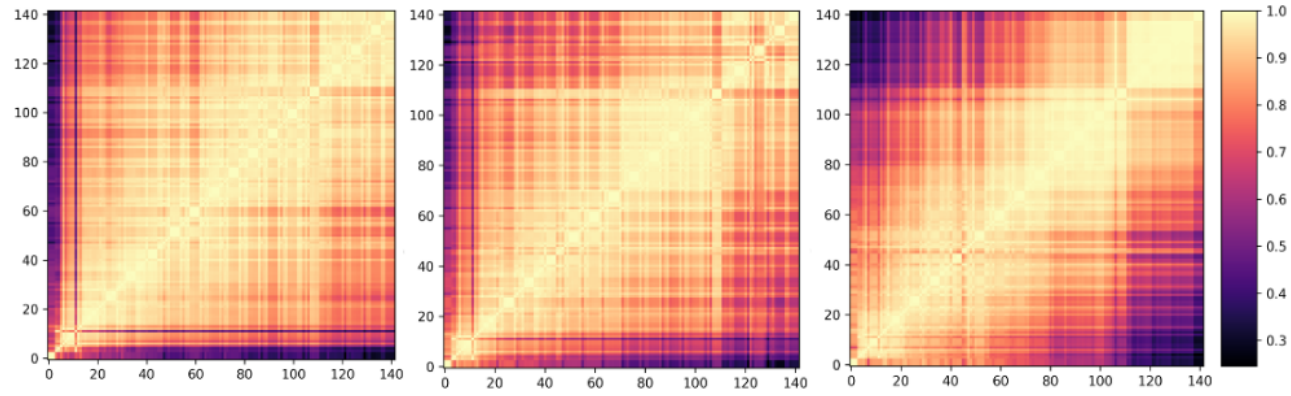

Figure 11. CKA maps for layer representations: Complexity 2,3,4 for PRP pretext, Network: ShuffleNet (Left to right).

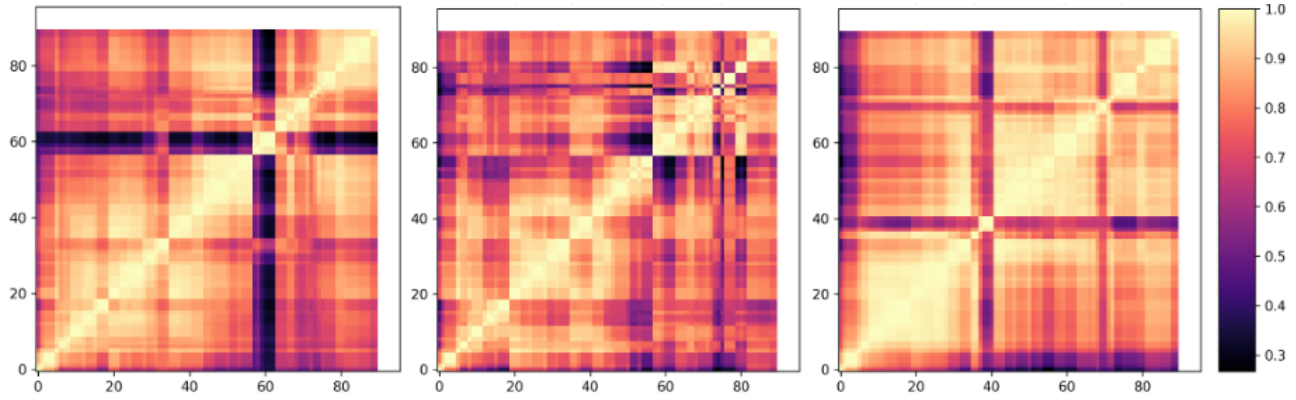

Figure 12. CKA maps for layer representations: Complexity 2,3,4 for RotNet pretext, Network: R21D (Left to right).

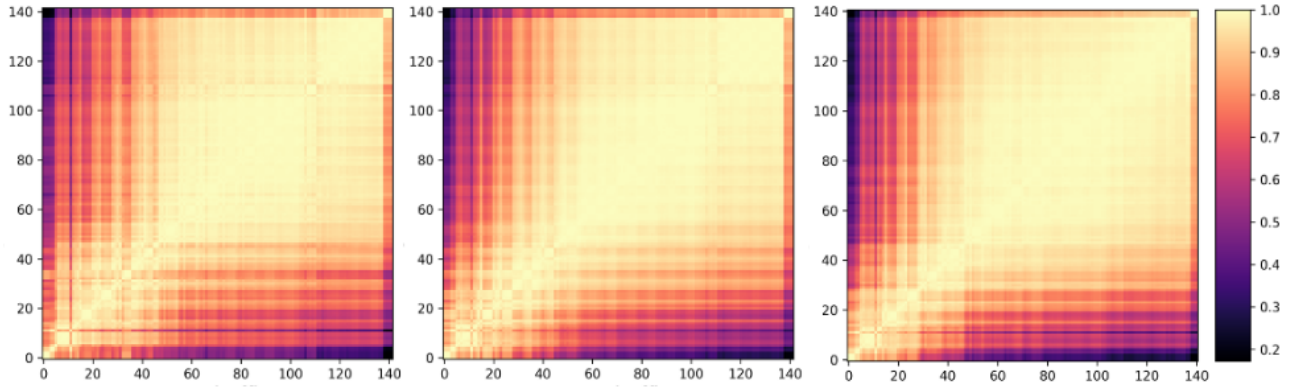

Figure 13. CKA maps for layer representations: Complexity 2,3,4 for RotNet pretext, Network: ShuffleNet (Left to right).

lyzing temporal information in video understanding models and datasets. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7366–7375, 2018. 7

[15] 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. 6

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

[17] Longlong Jing, Xiaodong Yang, Jingen Liu, and Y. Tian.

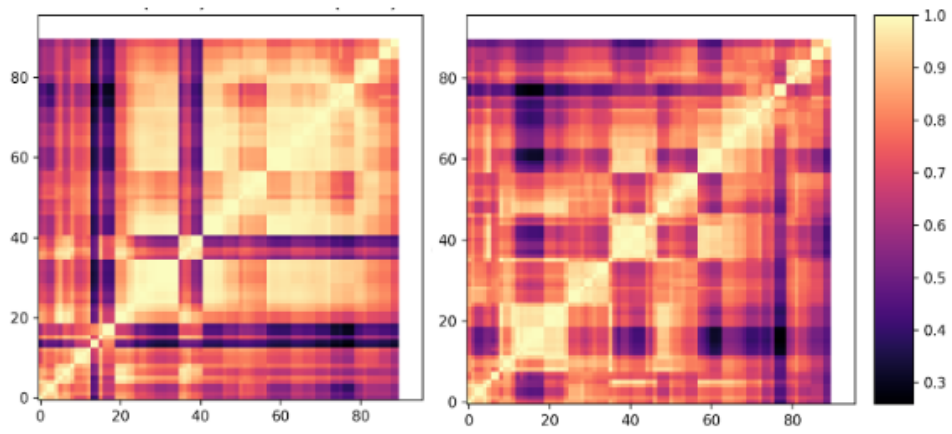

Figure 14. CKA maps for layer representations: Out of Distribution on VCOP and CVRL for R21D Network (Left to right).

Self-supervised spatiotemporal feature learning via video rotation prediction. *arXiv: Computer Vision and Pattern Recognition*, 2018. 5

[18] 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. 6, 7

[19] Okan Köpüklü, Neslihan Kose, Ahmet Gunduz, and Gerhard Rigoll. Resource efficient 3d convolutional neural networks. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 1910–1919. IEEE, 2019. 6

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

[21] 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. 7

[22] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *ECCV*, 2018. 7

[23] 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. 2

[24] Mandela Patrick, Yuki M. Asano, Ruth Fong, João F. Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal self-supervision from generalized data transformations. *ArXiv*, abs/2003.04298, 2020. 2

[25] 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. 2, 5

[26] 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. 6

[27] 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. 7

[28] 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. 2

[29] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 4489–4497, USA, 2015. IEEE Computer Society. 6

[30] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 6

[31] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018. 6

[32] 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. 2

[33] 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. 5

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

[35] 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. 5

[36] 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. 6