Benchmarking self-supervised video representation learning

Akash Kumar[1†] **Ashlesha Kumar**[2] **Vibhav Vineet**[3] **Yogesh Singh Rawat**[1]
CRCV, University of Central Florida[1] BITS Pilani[2] Microsoft Research[3]

Abstract

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

1 Introduction

Deep learning models require a large amount of labeled data for their training. Obtaining annotations at large-scale needs a lot of effort and it becomes even more challenging as we shift from image to video domain. There are several interesting directions focusing on this issue such as domain adaptation [65], knowledge distillation [18], semi-supervised learning [68], self-supervision [28] and weakly-supervised learning [50], which attempts to rely on the knowledge learned from existing source datasets and transfer to new target datasets with minimal labels. Among these approaches, self-supervised learning use pretext task as supervisory signal and does not require any labels on source datasets which makes it more favorable. [1]

In recent years, we have seen great progress in self-supervised learning (SSL) in video domain [66, 29, 69, 62, 44, 9]. More recently, the focus is more towards context-based learning which involves modifying input data such that to derive a classification [64, 12, 66, 29], reconstruction [69, 9] or generative [60, 52, 22, 56, 41] signal which can be used as a learning objective. The main focus of these works is designing a pretext task that is computationally inexpensive and which provides a strong supervisory signal such that the model learns meaningful *spatio-temporal* features.

Despite this great progress, it is non-trivial to compare these approaches against each other due to a lack of standard protocols. These methods are evaluated under different conditions and there is no standard benchmark to evaluate the fair effectiveness of these methods. A recent study [55]

[1†] Corresponding Author: Akash.Kumar@ucf.edu

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

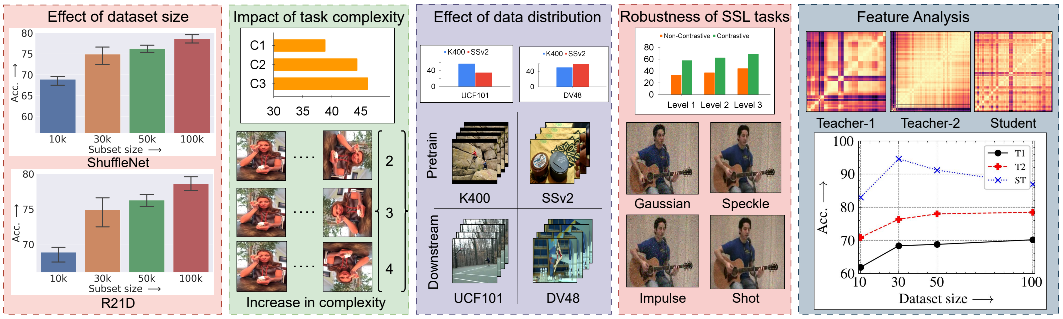


Figure 1: **Overview of proposed benchmark.** We study five different aspects in this benchmark study. Starting from left, 1) we show the analysis of *effect of dataset size vs training time*. As the dataset size increases, variation in performance decreases even with longer training time, 2) We show the effect of task complexity (C1, C2, C3 - Different complexities). The bottom figure shows one use case of how complexity increases for the RotNet task, and, the top figure shows how the performance varies for the R21D network, 3) With different data distribution shifts, the third sub-figure shows the impact of *target* data distribution on the *source* data, 4) We look into another data distribution shift due to introduction of noise. We see how *non-contrastive* tasks are more robust than *contrastive* ones even with increasing levels of severity of noise. The bottom part shows an example for each type of noise. Clips are provided in supplementary, and, 5) Finally, we further analyze whether the features learn complimentary information. In this sub-figure, we show that using different architectures as teachers can substantially improve performance even in a low-data regime.

attempts to take a step towards this direction, but it is mainly focused on downstream learning, without exploring the self-supervision aspect which is one of the main goals in our study. In this work, we present a benchmark where important self-supervised pre-training parameters are kept consistent across methods for a fair comparison. With the help of this benchmark, we study several critical aspects which are important for self-supervised learning; *1) effect of pretraining dataset size, 2) task complexity, 3) generalization under distribution shift, 4) robustness against data noise, 5) properties of learned features.*

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

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

Our contributions are threefold:

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

2 Benchmark Analysis Setup

In this section, we share the setup for analysis across the following five aspects in the next subsections.

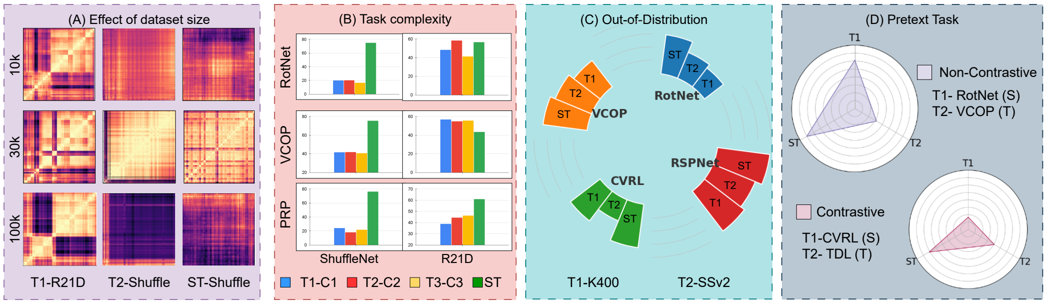


Figure 2: **Feature analysis overview.** This figure shows how knowledge distillation as a tool is beneficial across multiple scenarios. Brief details for each setup (Left to right): (A) *Effect of dataset size:* Teachers (T1, T2, T3) are different architectures for a single subset. (B) *Task Complexity:* Teachers are multiple complexities across the same task. (C1, C2, C3 - different complexities as teachers.) (C) *Out-of-Distribution:* Models from different *source* datasets are teachers. (D) *Pretext Tasks:* Spatial and temporal task networks are teachers.

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

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

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

Robustness of SSL tasks: In this aspect, we study the robustness qualities of SSL methods against data noise [24]. We analyze which factors play a key role in the robustness of these methods against such distribution shifts.

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

3 Lessons learned

With all the analysis along studied axes, we learned a few lessons in-between these axes such as: (i) Contrastive tasks are fast learners but are also most susceptible to noise. (ii) An increase in dataset size or complexity does not help smaller models in learning better spatio-temporal features but these features are more robust to noise. (iii) Temporal tasks are relatively more difficult to learn since looking at the correlation between time of training, increase in dataset size, and complexity, the performance gain is minimal in each of this axis. It means this category of tasks is actually difficult to solve. (iv) Spatio-temporal pretext tasks improve with the increase in complexity and dataset size (if the model permits), and their behavior to learn better spatio-temporal features is independent of data distribution. Using these lessons, we further do more analysis in feature space. We employ Knowledge Distiallation (KD) [13] as a tool to use the knowledge from different teachers. In Fig. 2, we show how KD as tool can help across different axes. We look into two downstream tasks: action classification and clip retrieval. In Table 1, we show that our model outperforms by a good margin on UCF101 against single and multi-modal approaches. We got competitive results on HMDB51 with a score of 51.5% wit only using 30k videos. Observations on Clip retrieval is shared in appendix.

3.1 Surprising Findings

We have multiple inference from different axes of analysis. However, to club a few which are new and helpful for video self-supervised community, we list down those here:

Dataset size and Training time Dependency: Against the conventional belief that a lot of training data is a *must* to achieve the best performance, we demonstrate that beyond a certain amount of training

Approach	Venue	NxW/H	Backbone	Pre-training	UCF101	HMDB51
Generative						
VIMPAC [53]	-	10x256	ViT-L	HTM	92.7	65.9
VideoMAE [56]	NeurIPS’22	16x224	ViT-B	K400	91.3	62.6
VideoMAE * [56]	NeurIPS’22	16x112	R21D-18	K400	76.2	45.4
Context						
PacePred [64]	ECCV’20	16x112	R21D-18	K400	77.1	36.6
TempTrans [27]	ECCV’20	16x112	R3D-18	K400	79.3	49.8
STS [61]	TPAMI-21	16x112	R21D-18	K400	77.8	40.5
VideoMoCo [41]	CVPR’21	16x112	R21D-18	K400	78.7	49.2
RSPNet [9]	AAAI’21	16x112	R21D-18	K400	81.1	44.6
TaCo [5]	-	16x224	R21D-18	K400	81.8	46.0
TCLR[12]	CVIU’22	16x112	R21D-18	K400	88.2	60.0
CVRL† [44]	CVPR’21	32x224	R21D-18	K400	92.9	67.9
TransRank [14]	CVPR’22	16x112	R21D-18	K200	87.8	60.1
Multi-Modal						
AVTS [34]	NeurIPS’18	25x224	I3D	K400	83.7	53.0
GDT [42]	-	32x112	R21D	IG65M	95.2	72.8
XDC [3]	NeurIPS’20	32x224	R21D	K400	84.2	47.1
Ours *	-	16x112	R21D-18	K400-30k	97.3	51.5

Table 1: **Comparison with previous approaches** pre-trained on K400. Ours (* best performing) is RSPNet pretrained on a 30k subset of K400. † modified backbone.

data, additional data provides diminishing returns for SSL in terms of performance improvement. This finding has significant implications, as it allows for a substantial reduction in the training data and there is almost a 10x reduction in training time which is particularly advantageous in computationally demanding video processing tasks. Furthermore, we show how KD as a tool, outperforms the original approach (100% data) using almost 90% less data further optimizing resource utilization by roughly 80%.

Robustness to real-world noise To our surprise, contrastive tasks are more susceptible to noise than non-contrastive ones. A smaller network tends to be more robust in some scenarios than a bigger network. We believe these findings are *novel and not known* to the community as there is no existing study exploring these aspects and are helpful where robustness is necessary for real-world deployment.

Complementary knowledge Improvement in performance in the case of KD from different data distributions and categories of tasks brings out a recipe for a new SSL task. This involves utilizing a multi-teacher multi-student setup, where each teacher specializes in spatial and temporal tasks and is trained on a mixture of data sources. Our analysis indicates this would provide a powerful learning scenario.

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

Acknowledgement This research is based upon work supported in part by the Office of the Director of National Intelligence (IARPA) via 2022-21102100001. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the US Government. The US Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

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

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

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

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

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

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

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

[8] J. Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *ArXiv*, abs/1907.06987, 2019.

[9] 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.

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

[11] 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.

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

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

[14] 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.

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

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

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

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

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

[20] 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.

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

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

[23] 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.

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

[25] De-An Huang, Vignesh Ramanathan, Dhruv Mahajan, Lorenzo Torresani, Manohar Paluri, Li Fei-Fei, and Juan Carlos Niebles. What makes a video a video: Analyzing temporal information in video understanding models and datasets. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7366–7375, 2018.

[26] 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.

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

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

[29] Longlong Jing, Xiaodong Yang, Jingen Liu, and Y. Tian. Self-supervised spatiotemporal feature learning via video rotation prediction. *arXiv: Computer Vision and Pattern Recognition*, 2018.

[30] 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.

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

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

[33] 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.

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

[35] 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.

[36] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021.

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

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

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

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

[41] 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.

[42] 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.

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

[44] 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.

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

[46] 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.

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

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

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

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

[51] 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.

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

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

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

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

[56] 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.

[57] 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.

[58] 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.

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

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

[61] 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.

[62] 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.

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

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

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

[66] 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.

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

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

[69] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. Video playback rate perception for self-supervised spatio-temporal representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6547–6556, 2020.

[70] 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.

Appendix

Here, we explain things in details about pretext task, architecture setup, 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.

1 Related work

Self-supervised learning There are several works in the domain of self-supervised learning for video representation learning [28, 49]. These approaches can be grouped into two main categories on the basis of pretext task: 1) context-based [31, 63, 2, 17, 64, 54, 67, 12, 27, 62, 44, 9, 14, 21, 45], and 2) cross-modal [43, 47, 1]. Cross-modal approaches use multiple modalities such as audio, video, optical flow, and camera positions, and rely on consistencies across these modalities. Context-based learning exploits data transformations to derive supervisory signals for training the model. Context-based pretraining tasks have evolved a lot in the past few years. Our work explores the domain of how much variation in learned representations under different transformations. In contrast to other approaches, context-based approaches exploit the spatial and temporal information independently by several transformations [39, 17, 66, 6, 64, 44, 62]. Recent works have started to transform the spatial and temporal domain together [31, 38, 54, 69, 9]. Incorporating multiple modalities improves performance, but, it's not available for all datasets, especially large-scale datasets. In this work, we restrict our focus to single-modality (RGB) approaches.

Self-supervised benchmarking There are some prior efforts focusing on benchmarking self-supervised learning in the image domain. In [19], the authors provide a detailed analysis of image-based self-supervised learning approaches and study how dataset size scaling affects the learned representations. Similarly in [32], the authors analyze how different model architectures play a role in visual self-supervised learning. In both these works, the authors did not focus on the importance of various pretext tasks themselves but only showed how certain pretext tasks can be improved. Therefore, their main focus was on downstream tasks rather than pretext learning. We, on the other hand, study different pretext tasks and analyze how various aspects affect feature learning. Moreover, these works are focused on the image domain, whereas we focus on the video domain. In recent work, [16], a study was performed to better understand unsupervised learning in the video domain, it basically explored the use of several pre-text tasks from the image domain and applied them to videos. We are not merely focusing on down-stream tasks and our attention is on the self-supervised aspect which includes factors such as data subset size, task complexity, dataset distribution, and noise robustness.

2 Self-supervised configurations

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

2.1 Tasks categorization

We analyze two different aspects of video pretext tasks: 1) transformations applied to data, and 2) learning objectives. Data transformations include, *spatial-based (S)*, *temporal-based (T)* and *spatio-temporal (ST)*. *Spatial* transformations include reshuffling of spatial patches, temporal consistent data augmentation, or rotation of images/patches. *Temporal* tasks involve permutation classification of frames/clip, order verification, clips sampling at different paces, or, contrastive learning from temporal triplets. *Spatio-temporal* tasks include those in which we modify both of these parameters simultaneously. This includes dilated sampling and simultaneous frame reconstruction, shuffling spatial and temporal domains, or, speed prediction, and contrastive visual features. Learning objectives can be either *contrastive* [10] or *non-contrastive* such as [56].

Following this categorization, we select at least two representative pretext tasks from each *transformation* category, one *contrastive* and one *non-contrastive*. We study the following pretext tasks in this study; RotNet (Rot) [29], Video Clip Order Prediction (VCOP) [66], Playback Rate Prediction

(PRP) [69], Spatiotemporal Contrastive Video Representation Learning (CVRL) [44], Temporal Discriminative Learning (TDL) [62], Relative Speed Perception network (RSPNet) [9], and *V-MAE* [56]. In concise summary, 1) *RotNet* applies geometrical transformation on the data, 2) *VCOP* learns the representation by predicting the permutation order, 3) *PRP* has two branches, discriminative and generative that concentrate on temporal and spatial aspect respectively, 4) *CVRL* learns to cluster the video of the same class with strong temporal coherent augmentations, 5) *TDL* works on temporal triplets and minimizes the gap between anchor and positive on the basis of visual content, 6) *RSPNet* applies contrastive loss in both spatial and temporal domain, and, 7) *V-MAE* [56] mask tokens of the input video and it tries to reconstruct those missing patches using an encoder-decoder architecture. More details are provided in the supplementary.

2.2 Benchmark details

Datasets: We experiment with two different dataset types, 1) where appearance is more important, and 2) where time is more important. For appearance based, we use Kinetics-400 [30], UCF101 [51], and HMDB51 [35], where appearance is more important (recognize activity with a single frame) than temporal aspect, and for temporal aspect, we use Something Something-V2 [20], where temporal information plays a significant role (require few frames to recognize activity).

Spatio-temporal architectures We analyze three different network capacities, 1) small-capacity, 2) medium-capacity, and 3) large-capacity. For small capacity, we study the following architectures; ShuffleNet V1 2.0X [70], SqueezeNet [26], and MobileNet [46]. For medium capacity we focus on conventional 3D architectures: C3D [57], R3D [23], and, R(2+1)D [58] (R21D); . And, for big-capacity architectures, we study VideoSwin [37], which is a transformer-based model.

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

Evaluation and analysis We use top-1 accuracy for action recognition which indicates whether the class prediction is correct or not. Clip retrieval calculates the *top-k* hits for nearest neighbor search, where $k = \{1, 5, 10, 20, 50\}$. For robustness performance, we calculate the relative robustness score (R_s) using original accuracy on clean test set (A_c) and perturbed accuracy on noisy test set(A_p) as $R_s = \frac{A_c - A_p}{A_c}$. We also provide qualitative feature analysis with the help of centered kernel alignment (CKA) maps [40]. CKA maps illustrate the model's hidden representations, finding characteristic block structures in models. There are two dominant properties of CKA maps: 1) *Feature similarity:* Lighter regions in map indicate more similar features between layers than darker regions. 2) *Grid patterns:* Two main patterns stand out, a staggering grid, which indicates models are capable of learning more, and, distinctive light/dark block patterns meaning the network reached its saturation point.

3 Benchmark analysis

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

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

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

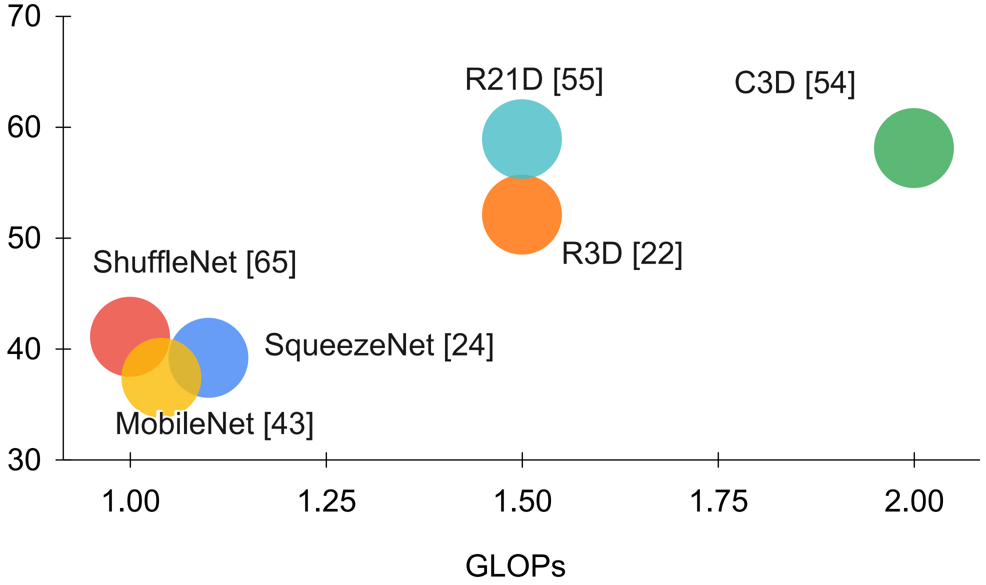


Figure 1: **Architecture Performance Analysis:** Variation in performance for different architectures. X-axis shows the relative floating point operations and Y-axis shows the Top-1 Accuracy.

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

Robustness of SSL tasks: In this aspect, we study the robustness qualities of SSL methods against data noise [24]. We analyze which factors play a key role in the robustness of these methods against such distribution shifts.

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

3.1 Preliminary Experiments

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

Backbone architectures: Looking into smaller and medium capacity networks in Figure 1, ShuffleNet outperforms among smaller networks, whereas considering the trade-off between the number of trainable parameters and performance R21D performs better in medium network category. Among big capacity networks, we look into a few recent end-to-end video-based transformer networks [4, 15, 7, 37], and Video Swin [37] outperforms other architectures by a margin of 1-3% on K400.

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

Linear probe vs finetuning: In the linear probe, we train only the linear layers attached for classification while freezing other network weights, whereas in finetuning the whole network is trained end-to-end. In our preliminary experiments we use Kinetics-400 for pretraining and UCF-101 as the target dataset. On several pretext tasks, we observe an average drop of 25% (ShuffleNet) and 40% (R21D) in performance when comparing linear probe with finetuning. However, we do not

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

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

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

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

usually observe this significant drop when both the pretraining and target datasets are the same [49]. It indicates that *finetuning is important for the model to adapt to downstream dataset* in case it is different. Therefore, some of the existing works [55] rely on finetuning when the source and target datasets are different. Since we are interested in cross-dataset learning, we perform finetuning on all our downstream datasets.

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

3.2 Effect of dataset-size

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

Observations: From Table 2, we observe that apart from TDL each pretext task performance improves with an increase in subset size. If we look into specific pretext task transformation category (Table 2), the most gain with an increase in data is for *spatio-temporal* tasks (13%), whereas the least gain is for *temporal* pretext tasks (3%). Looking across different architectures in Figure 2, there's a minimal gain for R21D and ShuffleNet beyond increasing dataset size from 30k subset against VideoSwin which improves with an increase in dataset size which relates to similar behavior like image models discussed in [19]. Analyzing the effect of the duration of training across different pretext tasks, in Table 3, the performance gain is minimal (<1.5%) after training for more than 100 epochs. Comparing contrastive and non-contrastive approaches, the gain in contrastive-based approaches is on average 1% compared to 5% for non-contrastive tasks beyond *100 epochs* of training.

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

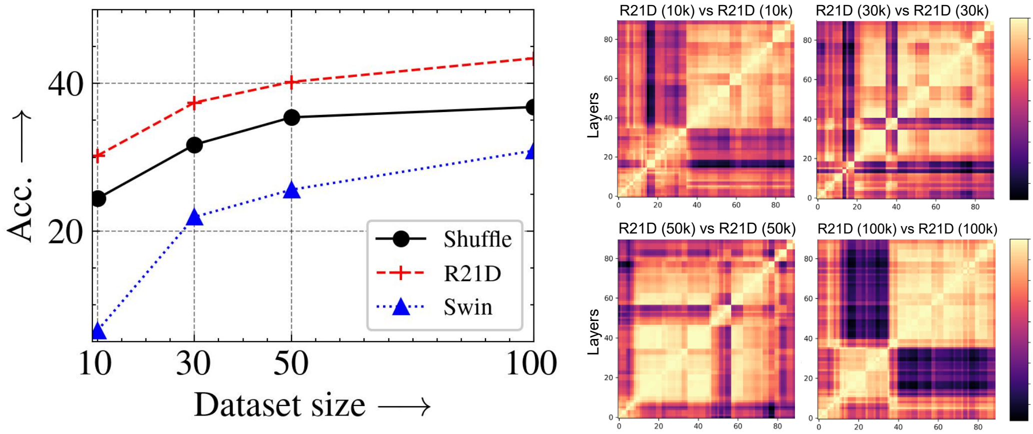


Figure 2: Left: **Dataset subset** performance for three different architectures on RSPNet pretext task (x-axis: subset size, y-axis: Top-1 Accuracy). Here, 10 means 10k dataset subset, 30 means 30k, and so on. Right: **CKA maps** for RSPNet on different subsets with R21D backbone.

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

Table 3: **Performance at different stages** of training for all pretext tasks on R21D with 50k pre-training subset size.

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

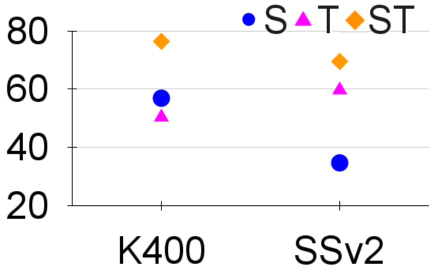
Table 4: **Complexity Variation.** TC: Task complexity. Results are shown on UCF101 with ShuffleNet/R21D backbone.

3.3 Impact of change in task complexity

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

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

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



(a) UCF101

Figure 3: **Effect of different dataset distributions:** Pretraining on K400 and SSv2 with 30k subset size, finetuning on UCF101 using R21D network. Here, S, T, and ST mean spatial(CVRL), temporal(VCOP), and, spatio-temporal(RSPNet) respectively. X-axis shows *source* dataset and Y-axis shows Top-1 accuracy.

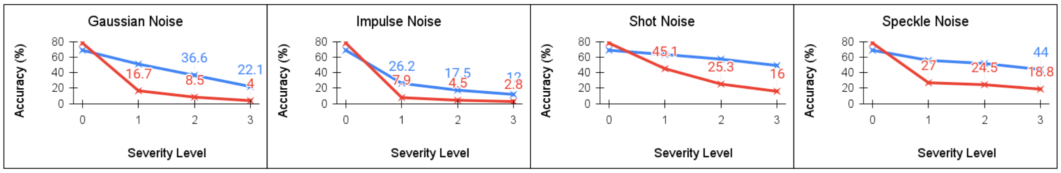


Figure 4: Performance with different types of noises. ShuffleNet and R21D scores are shown by blue and red lines respectively.

3.4 Effect of dataset distribution

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

Observations: Looking into Figure 3, we observe that *spatio-temporal* pretext task outperforms other pretext tasks on both *target* (downstream) datasets UCF101 and DV48 by a margin of 15-40% and 10-13% respectively whether the *source* datasets is K400 or SSv2. Comparing, spatial and temporal-based pretext tasks, we see that they are *majorly* dependent on *source* datasets. Looking at Figure 3, performance is better on both *target* datasets if *source* dataset has the same underlying properties as the pre-text task is trying to learn. Furthermore, the spatial task is more dependent on the *source* dataset, since the relative drop on both UCF101 and DV48 for CVRL is significant (40% and 30% respectively) when the source dataset is SSv2 against K400. However, in the case of the temporal task, the drop is 15% and 10% respectively when the source dataset is K400 against SSv2.

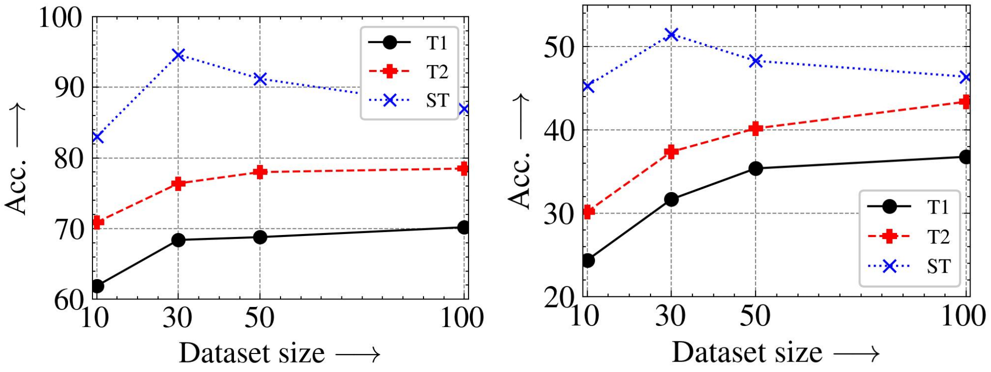
Inference: (i) *Spatio-temporal pretext task learns better features independent of source and target data distribution.* (ii) *Spatial and temporal pre-text tasks are better learners when source data distribution belongs to spatial and temporal respectively.* (iii) *Temporal pretext task prevails when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

3.5 Robustness of SSL tasks

Similar to OOD datasets, introducing noise also shifts the distribution of datasets. We evaluate models on different types of noises introduced in [48] with different severity levels on the UCF101 test dataset. Specifically, we probe into four different types of appearance-based noises: Gaussian, Shot, Impulse, and Speckle [24]. Here we look into the following aspects: a) how robust different categorizations of pretext tasks are? b) is the network's architecture dependent on the noise in the

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

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



(a) UCF101

(b) HMDB51

Figure 5: **Knowledge distillation** using teachers trained on multiple subset sizes on RSPNet. Student: ShuffleNet UCF101/HMDB51. Here T1 is Teacher-1 (shufflenet) and T2 is teacher-2 (R21D).

dataset? In the main paper, we only discuss one severity level and have provided a detailed analysis of multiple severity levels in the supplementary.

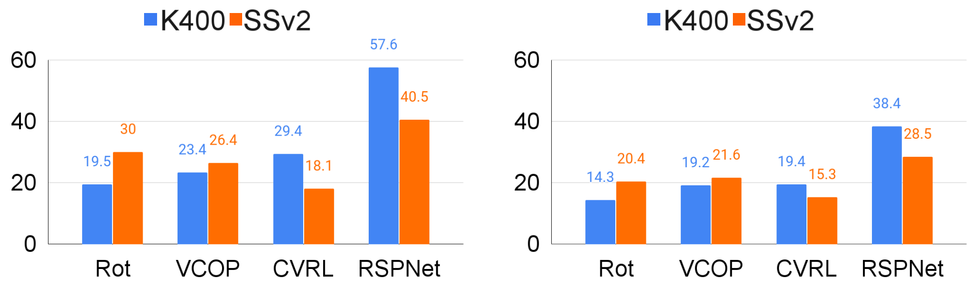
Observations: From Table 5, we observe that the relative drop in performance for contrastive tasks is more than non-contrastive tasks for both R21D and ShuffleNet backbone. The most and least robust models are RotNet-R21D and PRP-R21D with 10.7% and 70.1% relative decrease. From Figure 4, we can observe looking across different *severity levels* for each type of noise ShuffleNet is more robust than R21D.

Inference: (i) *Contrastive approaches are less robust to noise when compared with non-contrastive approaches.* (ii) *ShuffleNet outperforms R21D in robustness few scenarios despite being smaller in terms of a number of parameters.*

3.6 Feature analysis

We further analyze the learned features by these pretext tasks under different configurations. We specifically focus on understanding the complementary nature of these features. We employ knowledge distillation [13] as a tool to study this aspect. It is based on the idea that distilled knowledge from the ensemble of teacher networks makes the student model stronger. We use our benchmark models as teachers in different combinations to analyze whether a student learns orthogonal information on four different axes: 1) different architectures as the teacher within a *dataset size*, 2) teachers with different complexities in a pretext task, 3) models from multiple *source* datasets, and, 4) same architecture as teachers from multiple pretext tasks. Figure 2 summarizes the *observations* for each aspect.

Observations: Although teacher network performance improves with subset, gain in complementary information reduces beyond 30k (Fig. 5). However, distillation does help in the reduction of training time with a significant improvement in performance which is evident from Fig. 2(a). Independent of the pretext tasks category smaller architecture learns complimentary information and outperforms the teacher whereas bigger architecture it's task-dependent. Irrespective of task category whether transformation-based or contrastive, each task learns corresponding features from both source datasets and outperforms the teacher. Student network outperforms standalone spatio-temporal network performance in both contrastive and non-contrastive domains.



(a) UCF101

(b) HMDB51

Figure 6: **Top@5 Clip Retrieval** - R21D on a) UCF101 and b) HMDB51, pre-trained on K400 and SSv2 - 30k subset.

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

4 Lessons learned

With all the analysis along studied axes, we learned a few lessons in-between these axes such as: (i) Contrastive tasks are fast learners but are also most susceptible to noise. (ii) An increase in dataset size or complexity does not help smaller models in learning better spatio-temporal features but these features are more robust to noise. (iii) Temporal tasks are relatively more difficult to learn since looking at the correlation between time of training, increase in dataset size, and complexity, the performance gain is minimal in each of this axis. It means this category of tasks is actually difficult to solve. (iv) Spatio-temporal pretext tasks improve with the increase in complexity and dataset size (if the model permits), and their behavior to learn better spatio-temporal features is independent of data distribution.

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

Action Classification For this task, the model is finetuned end-to-end on downstream datasets, on UCF101 and HMDB51. In Table 1, we compare our best-performing model with other previous state-of-the-art approaches. ***Observations:*** With only 30k videos compared to 200k+ videos used by other pretext tasks, we show that our model outperforms by a good margin on UCF101 against single and multi-modal approaches. We got competitive results on HMDB51 with a score of 51.5%. Looking in depth regarding HMDB performance, approaches that are ahead of ours are [34], [42] in cross-modal and [44] and [12] in single modality (ignoring ViT backbone and IG65M dataset). Compared to ours, these approaches use bigger spatial resolution (CVRL and AVTS), and multiple modalities (AVTS and GDT) and all of them use more effective frames.

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

4.1 Surprising Findings

We have multiple inference from different axes of analysis. However, to club a few which are new and helpful for video self-supervised community, we list down those here:

Dataset size and Training time Dependency: Against the conventional belief that a lot of training data is a *must* to achieve the best performance, we demonstrate that beyond a certain amount of training data, additional data provides diminishing returns for SSL in terms of performance improvement. This finding has significant implications, as it allows for a substantial reduction in the training data and there is almost a 10x reduction in training time which is particularly advantageous in computationally demanding video processing tasks. Furthermore, we show how KD as a tool, outperforms the original approach (100% data) using almost 90% less data further optimizing resource utilization by roughly 80%.

Robustness to real-world noise To our surprise, contrastive tasks are more susceptible to noise than non-contrastive ones. A smaller network tends to be more robust in some scenarios than a bigger network. We believe these findings are *novel and not known* to the community as there is no existing study exploring these aspects and are helpful where robustness is necessary for real-world deployment.

Complementary knowledge Improvement in performance in the case of KD from different data distributions and categories of tasks brings out a recipe for a new SSL task. This involves utilizing a multi-teacher multi-student setup, where each teacher specializes in spatial and temporal tasks and is trained on a mixture of data sources. Our analysis indicates this would provide a powerful learning scenario.

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

5 Pretext Tasks Details

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

5.1 Spatial Transformation

Rotation Net [29] (RotNet) applies geometrical transformation on the clips. The videos are rotated by various angles and the network predicts the class to which it belongs to. Since the clips are rotated, it helps the network to not converge to a trivial solution.

Contrastive Video Representation Learning [44] (CVRL) technique applies temporally coherent strong spatial augmentations to the input video. The contrastive framework brings closer the clips from the same video and repels the clip from another video. With no labels attached, the network learns to cluster the videos of the same class but with different visual content.

5.2 Temporal Transformation

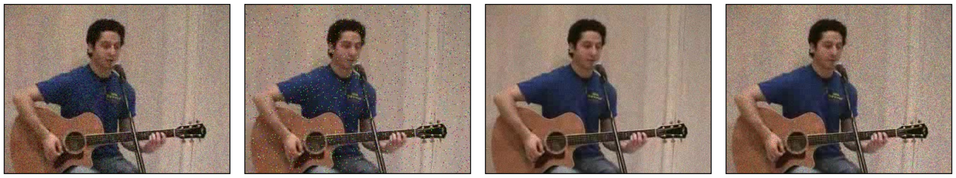


Figure 7: **Example frame sample for each noise** Gaussian, Impulse, Shot, and Speckle from left to right. Sample clips are provided in the supplementary.

Video Clip Order Prediction [66] (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 [62] (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 belong to the same temporal interval and have a high degree of resemblance in visual content compared to the negative.

5.3 Spatio-Temporal Transformation

Playback Rate Prediction [69] (PRP) has two branch, generative and discriminative. Discriminative focuses on classifying the clip's sampling rate, whereas, generative reconstructs the missing frame due to dilated sampling. Thus, the first one concentrates on the temporal aspect, and the second one on the spatial aspect.

Relative Speed Perception Network [9] (RSPNet) applies contrastive loss in both spatial and temporal domain. Clips are samples from the same video to analyze the relative speed between them. A triplet loss pulls the clips with the same speed together and pushes clips with different speeds apart in the embedding space. To learn spatial features, InfoNCE loss [59] is applied. Clips from the same video are positives whereas clips from different videos are negatives.

Video MAE [56] (V-MAE) applies a spatio-temporal tube masking to the input video. The pretext task is to reconstruct those missing tubes. Mean-squared error loss is applied between the masked tokens and the reconstructed tokens.

6 Implementation Details

6.1 Architecture Details

Preliminary research has shown that 3D networks [58, 23] have outperformed 2D CNN variants on video recognition tasks. We looked into three types of capacity - small, medium, and big on the basis of the number of trainable parameters. The architecture details of all networks are mentioned in the supplementary.

Small capacity networks: are resource efficient, implying they can be trained in larger batches within a short span of time. The network selection is done on the basis of supervised training scores on Kinetics[30] and UCF101[33]. ShuffleNet V1 2.0X [70] utilizes point-wise group convolutions and channel shuffling. SqueezeNet [26] reduces the filter size and input channels to reduce the number of parameters. MobileNet [46] 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: Comparing across four transformer architectures, ViViT [4] Timesformer [7], VideoSwin [37] and MViT [15], we selected VideoSwin, because it outperforms others on Kinetics 400 dataset.

Based on [33], 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 [70]: It utilizes point-wise group convolutions and channel shuffling and has 3 different stages. Within a stage, the number of output channels remains the same. As we proceed to the successive stage, the spatiotemporal dimension is reduced by a factor of 2, and the number of channels is increased by a factor of 2. V1 denotes version 1 of ShuffleNet and 2.0X denotes the 2 times number of channels compared to the original configuration.

3D-SqueezeNet [26]: It uses different alterations 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 [46]: 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 [57]: 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 [23]: The 2D CNN version of ResNet architecture is recasted into 3D CNNs. It has skip connections that help make the gradient flow better as we build deeper networks.

R(2+1)D [58]: In this architecture, 3D convolution is broken down into 2D and 1D convolution. 2D convolution is in the spatial dimension and 1D convolution is along the temporal dimension. There are two benefits of this decomposition: 1) An increase in non-linearity as the number of layers has increased, and, 2) Due to factorization of 3D kernels, the optimization becomes easier.

VideoSwin [37] It is an inflated version of original Swin [36] transformer architecture. The attention is now spatio-temporal compared to the previous which is only spatial. 3D tokens are constructed from the input using a patch partition and sent to the network. The architecture includes four stages of transformer block and patch merging layers.

6.2 Original and Noise Datasets

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. An example of each type of noise is shown in Fig. 7.

6.3 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 compared 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 cropped for all the approaches.

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

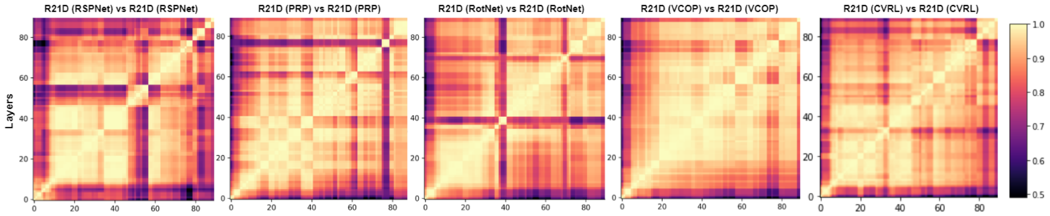


Figure 8: **Pretext tasks CKA maps** for RSPNet, PRP, RotNet, VCOP, CVRL on K-400 50k subset using R21D network (Left to right). R21D pretrained on K400 shows a semi-block structure for VCOP, indicating the near-saturation condition of the network on this pretext task. It shows a more prominent grid-based structure on CVRL and RSPNet instead. These observations corroborate the quantitative results, where pretraining on K400 for both CVRL and RSPNet gives better performance.

6.4 Datasets

Here we discuss datasets in detail. We use Kinetics-400 (K400) [30] and Something-Something V2 [20] for our pre-training. For the downstream task evaluation, we perform our experiments on UCF-101 [51], and HMDB-51 [35]. Since the pretraining and finetuning datasets are different, the performance variation will provide us with a better picture of 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, and, 2) HMDB51-7k. The datasets can be categorized into two ways:

Appearance-based: Kinetics, UCF101 and HMDB51 comes under this category [11, 25]. 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 the K400-50k subset for all experiments.

Temporal-based: In Kinetics, we can estimate the action by looking at a single frame [11, 25]. However in SSv2, we can't describe the activity class until we look into a few continuous frames. It shows that the temporal aspect plays an important role for these datasets, that's why we categorize them into temporal-based datasets.

UCF-101 [51] : 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 [35] : 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.

7 Additional Results

Here, we will talk about some additional results, to further strengthen the claims made in the main paper.

7.1 Preliminary Experiments

Pretext tasks evaluation Figure 8 depicts the hidden representations of R21D network pretrained on different pretext tasks. Here the 50k subset of K-400 was used for pretraining and finetuned on UCF-101.

Linear Probing vs Finetuning Firstly, we discuss linear probing (LP) vs finetuning (FT) results for different pretext tasks and different architectures. From Table 7, we can see that FT outperforms

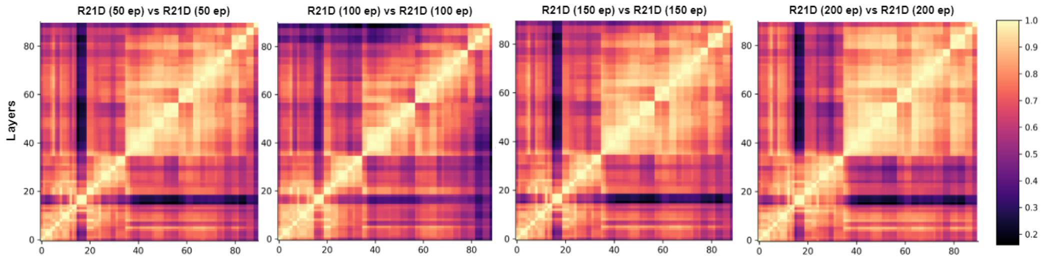


Figure 9: **Training time CKA maps** on 50, 100, 150, 200 epochs of R21D network on RSPNet pretext for K-400 10k subset (Left to right). The block structure is visible from 50 epochs itself, which then darkens and becomes prominent by 200 epochs. With 10k subset, the saturation starts hitting at 100 epochs.

	Non-contrastive			Contrastive		
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 6: **Evaluation of different pretext tasks** on different subset size on R21D network on HMDB51 dataset.

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

Table 7: **Downstream accuracy** classification on UCF-101 dataset. FT: Finetuning LP: Linear Probing

Networks	Parameters	GFLOPs	Rot†	VCOP †	PRP†	RSPNet
ShuffleNet	4.6M	1.1	**42.2**	**41.6**	**41.1**	**68.8**
MobileNet	3.1M	1.1	38.0	40.0	37.4	63.1
SqueezeNet	1.9M	1.8	41.3	41.4	39.2	62.9
C3D	27.7M	77.2	**57.7**	54.5	58.1	67.6
R3D	14.4M	39.8	51.1	50.7	52.1	62.1
R(2+1)D	14.4M	42.9	46.9	**56.8**	**58.9**	**78.0**

Table 8: **Comparison of FLOPs** and trainable parameters for each network on UCF101 dataset. † - pretraining on Kinetics 700 [8].

LP by a margin of approximately 20% and 40% on ShuffleNet and R21D respectively. Thus, we perform finetuning for all of our analyses.

Network Parameters We have shown the performance across different architectures in Table 8. ShuffleNet and R21D perform the best across small and medium capacity networks in most of the pretext tasks. Thus, we choose ShuffleNet and R21D for our benchmark analysis.

7.2 Effect of dataset size

In Table 2, we extend results for different pretext tasks on the HMDB51 dataset. Similar to UCF101, *the scale in subset size doesn't reciprocate to gain in performance* for all pretext tasks on the HMDB51

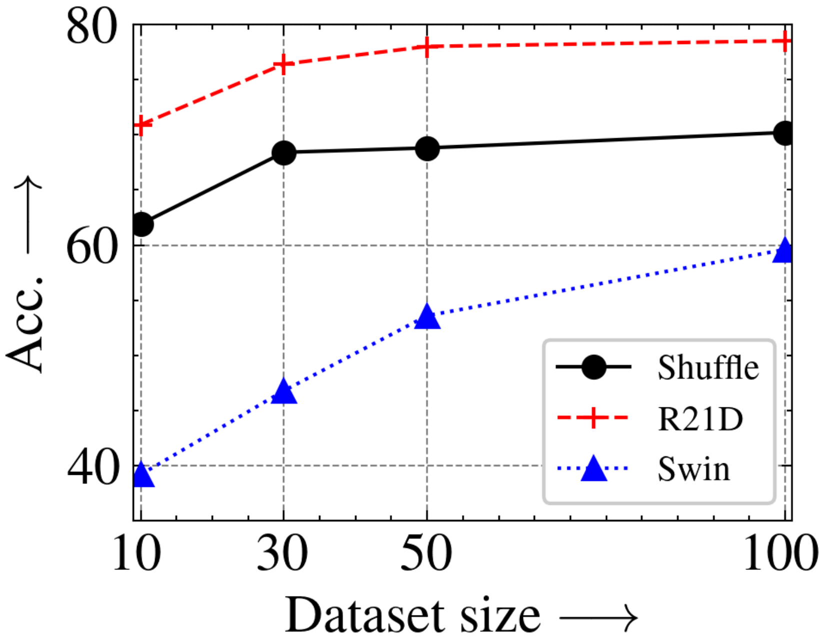


Figure 10: **Multiple architectures and data subsets on UCF101.** Pretext task is RSPNet. (x-axis: subset size, y-axis: Top-1 Accuracy) Here, 10 means 10k dataset subset, 30 means 30k and so on.

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 9: RSPNet with different subset size on ShuffleNet/R21D/VideoSwin on UCF101 dataset.

dataset. From Figures 10 and 11, we see that performance increase for Swin by a good margin, whereas in the case of ShuffleNet and R21D, it's relatively less beyond 50k subset.

Training time Table 9 shows VideoSwin saturates at 150 epochs on UCF101 whereas CNN architectures saturate earlier (100 epochs) which reflects the limitation of model capacity. Figure 9 shows the emergence of block structures for the R21D network trained on RSPNet for K400 10k. The saturation point has reached earlier around 100 epochs which supports the hypothesis in the main work that CNN architectures mostly saturates around 100 epochs. We see a similar pattern even after increasing the dataset size.

7.3 Impact of task complexity

Figures 12 show for ShuffleNet dark patterns with an increase in complexity. R21D shows staggering grids. It supports our hypothesis that *model capacity* plays an important role to learn meaningful features and always increasing the complexity doesn't reciprocate to *better spatio-temporal features*.

7.4 Effect of data distrbituion

Figure 14 illustrates CKA maps for networks pretrained on *different source datasets* - for R21D pretrained on K400-50k on VCOP and CVRL respectively. The stark difference in the semi-block structure of *spatial (VCOP)* vs grid-like structure of *spatio-temporal (CVRL)* shows spatio-temporal outperforms spatial pretext task.

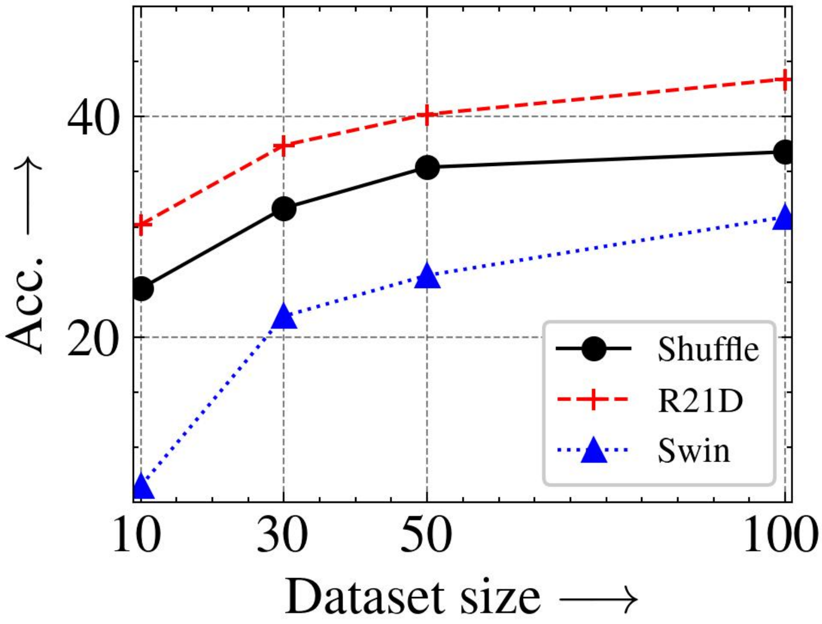


Figure 11: **Multiple architectures and data subsets on HMDB51.** Pretext task is RSPNet. (x-axis: subset size, y-axis: Top-1 Accuracy) Here, 10 means 10k dataset subset, 30 means 30k and so on.

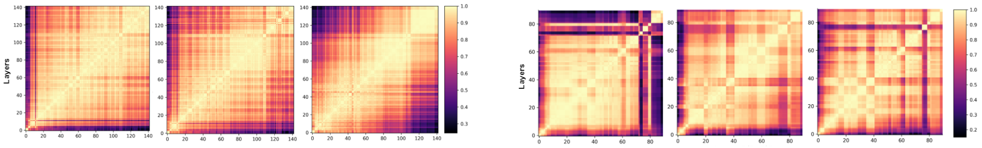


Figure 12: **Complexity CKA maps** PRP ShuffleNet (Left) and R21D (Right) network increasing complexity from 2 to 4 (Left to right). ShuffleNet has lower performance than R21D, and it shows darkest patterns when complexity is increased from 3 to 4. For both of these complexities, R21D shows staggering grids.

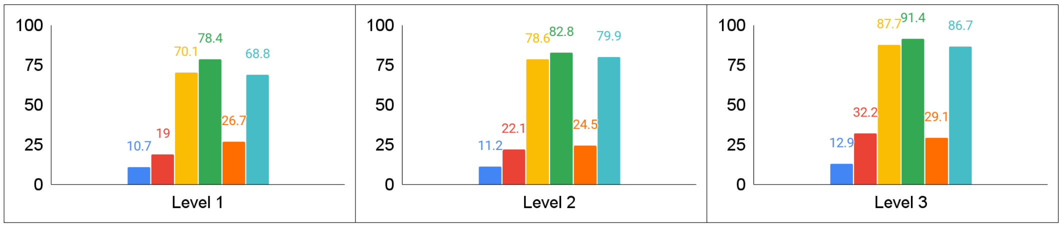


Figure 13: **Relative decrease in performance** at three different severity levels in increasing order from left to right. The pretext tasks is depicted by following colors - RotNet, VCOP, PRP, CVRL, TDL, RSPNet.

7.5 Robustness of SSL tasks

Table 10 shows the performance of each pretext on each type of noise for severity level 1. Fig. 13 shows a relative decrease in performance for three different severity levels on the UCF101 dataset. *Non-contrastive* tasks are more robust than *contrastive* on average even at different severity levels.

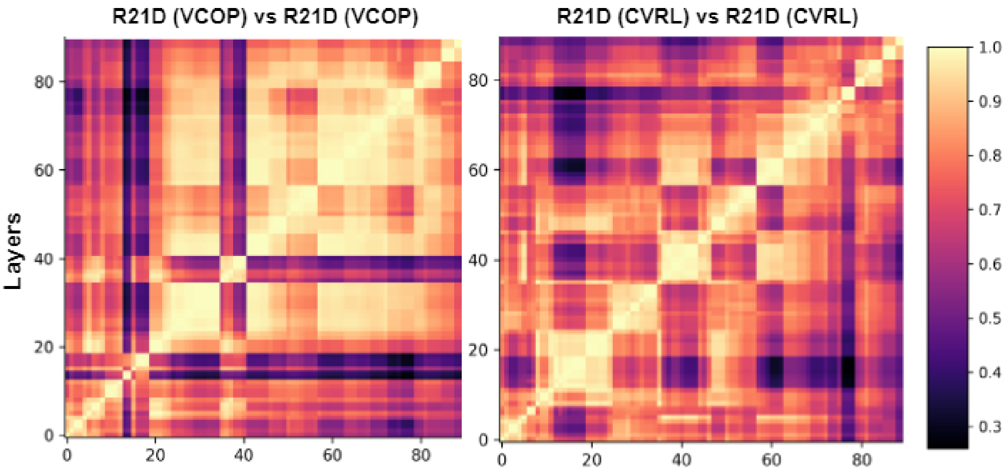


Figure 14: **Out-of-distribution CKA maps:** on VCOP and CVRL for R21D Network (Left to right). The semi-block structure of VCOP contrasts sharply with the grid-like structure of CVRL.

	Non-contrastive			Contrastive		
	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 10: Analysis of all pretext tasks with noise severity level 1 on R21D network on UCF101 dataset.

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: **Complexity variation** with at three levels as teachers (T1, T2, T3) for all three pretext tasks. TC: Task complexity. Results are shown on UCF101 with ShuffleNet/R21D as backbones.

7.6 Feature Analysis

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. Here we show the numbers quantitatively. Table 11 shows smaller architecture leans complementary information whereas bigger architecture depends on pretext task. Table 12 shows that for each pretext task, we learn *complimentary information* from two *different source* datasets. Thus, the student always outperforms the teachers. Table 13 shows that distilling knowledge from a *spatial* and a *temporal* task outperforms the standalone *spatio-temporal* task in both *contrastive* and *non-contrastive* case.

	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 12: **Out-of-Distribution** settings on UCF101 dataset using R21D network with teachers as different *source* datasets.

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

Table 13: **Knowledge distillation across different pretext tasks.** Teachers: ShuffleNet; Student: ShuffleNet.

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 14: Top K Clip Retrieval on HMDB51/UCF101 across different architectures for RSPNet.

7.7 Clip retrieval

In Table 14, we show clip retrieval across different architectures on HMDB51 and UCF101 dataset. Amongst small capacity networks, ShuffleNet outperforms others and in medium-capacity R21D outperforms.