Supplementary Material

Rui Qian[1], Shuangrui Ding[2], Xian Liu[1], and Dahua Lin[1,3⋆]

[1] The Chinese University of Hong Kong, Hong Kong, China
[2] Shanghai Jiao Tong University, Shanghai, China
[3] Shanghai Shanghai Artificial Intelligence Laboratory, Shanghai, China
{qr021,lx021,dhlin}@ie.cuhk.edu.hk dsr1212@sjtu.edu.cn

1 More Training Details

In default settings, we sample 16-frame clip sequence, then apply temporally consistent random resized crop and random horizontal flip to obtain the frame sequence with spatio-temporal resolution $16 \times 112 \times 112$. We randomly select a frame, and repeat 16 times on temporal axis to form the static frame sequence, which does not contain temporal dynamics. We respectively apply color jitter and Gaussian blur to form the RGB input v, static frame input s. We calculate the difference between adjacent frames in v to form frame difference input d. In this way, v, s and d are all of spatio-temporal resolution $16 \times 112 \times 112$ and input to the same encoder.

The feature maps extracted by the encoder are of size $C \times T \times H \times W = 512 \times 2 \times 7 \times 7$. We set the feature transformation σ to identity mapping in default, and employ two-layer MLP 100-512-512 as the light-weight decoder to reconstruct the feature vector. In the first 5 epochs, we do not include $\mathcal{L}_{loc}$ in the loss function to stabilize training.

We respectively use the training set of UCF-101, Kinetics-400 and Diving-48 for self-supervised pretraining. Following [1,4,3], we use split 1 of UCF-101 and HMDB-51, and V2 test set of Diving-48 for downstream evaluation and analysis.
Action Recognition. We use the pretrained parameters to initialize the network except the last fully-connected layer. We employ two popular protocols: (1) *Finetune* the whole network; (2) Only train the last linear classifier denoted as *linear probe*. We follow the prevalent evaluation protocols [6] to uniformly sample ten 16-frame clips from each video, then center crop and resize to 112×112. We average the softmax probability of each clip and report Top-1 accuracy.
Video Retrieval. We use the pretrained model to extract video features without training. We use videos in the test set as query and retrieve nearest neighbors in the training set, and report Top-k recall R@k.

2 More Ablation Study

We show more ablation studies regarding to some modules in the framework, including the feature transformation head σ, the decoder g, and the number of valid concepts top-K.

⋆ Corresponding author. Email: dhlin@ie.cuhk.edu.hk

σ	Shape	UCF-101	HMDB-51
Identity	512-512	72.1	45.9
Linear	512-512	73.3	46.5
MLP	512-512-512	73.4	46.3

Table 1. Ablation study on the feature transformation head.

Feature Transformation Head. We use the same architecture but do not share parameters for all three transformation heads, i.e., σ_v, σ_s and σ_d, and we use the same symbol σ for concise presentation in Table. 1. We report the linear probe accuracy on UCF-101 and HMDB-51. We observe that using extra transformation improves the performance over the identity mapping, which is partially consistent with the analysis in SimCLR [2]. But the difference is that nonlinear transformation head is comparable with linear transformation, this is probably because we calculate cosine similarity to generate latent concept code for contrast, which has contained nonlinear operations ($\mathcal{L}_2$ normalization). Thus, a linear transformation head is enough to further improve performance.

g	Shape g_v	Shape g_s/g_d	UCF-101	HMDB-51
Linear	100-512	50-512	70.2	44.5
MLP	100-128-512	50-128-512	71.8	45.6
MLP	100-512-512	50-512-512	72.1	45.9

Table 2. Ablation study on the concept latent code decoder.

Concept Code Decoder. We show ablation study on the concept latent code decoder in Table. 2, and report linear probe accuracy on two datasets. We set $K_s = K_d = 50$, and compare three variants of the decoder. We observe that the linear decoder leads to slight performance drop due to limited reconstruction ability. And the performance under different MLP designs maintains stable.

K	Avg	UCF-101	HMDB-51
2	1.6	70.4	44.1
5	3.8	72.1	45.9
10	7.1	71.8	45.3

Table 3. Ablation study on the number of valid concepts.

Number of Valid Concepts. We explore using different number of valid concepts, i.e., the hyper-parameter K in Eq. 8. The total number of concepts are $K_s = K_d = 50$, and in default settings we select top-10%, i.e., $K = 5$. Recall that the final number of valid concepts is the intersection of top-K indexes from

two concept codes, the real number of valid concepts is no greater than K. Thus, besides linear probe accuracy, we also report average number of valid concepts after taking intersection. From Table 3, we can see that the ratio of final number of valid concepts over K is around 0.7-0.8, and we reach best performance with $K = 5$. It indicates that when K is small, we just neglect some useful concepts, thus failing to make full use of the detailed information of valid concepts. While when K is large, there exists redundancy and also corrupts the performance.

3 More Experimental Results

We also validate the potential of our method to scale to deeper backbone or larger resolution. Due to limited computation resource, we do not directly compare with CVRL [5] under the same settings. But in Table 4, the improvements brought by using deeper backbone (R3D-34 vs R(2+1)D-18) or larger resolution (224 vs 112) indicate that our method has potential to reach higher performance.

Method	Backbone	Resolution	UCF-101	HMDB-51
Ours	R(2+1)D-18	112	86.1	54.8
Ours	R(2+1)D-18	224	89.2	60.1
Ours	R3D-34	112	89.4	58.3
CVRL [5]	R3D-50	224	92.9	67.9

Table 4. Experiments on deeper backbone and larger resolution.

Besides, we also compare the results of using separate backbones for v, s, d, or share the same backbone. We pretrain for 100 epochs and show the linear probe accuracy on UCF-101 every 20 epochs as well as the per epoch training time in Table 5. We observe that these two settings reach comparable performance, but using the same backbone (default setting) leads to faster convergence. Also, using three different backbones costs about 1.2× training time. This is because after the normalization data pre-processing, the distribution of the original clip, static frame and frame difference is not that different, thus it is practical to use the same backbone to extract features. And the shared gradient back-propagation improves learning efficiency.

Setting	20	40	60	80	100	Time
Share	52.3	61.5	66.8	71.1	72.1	1.0×
Separate	45.5	57.2	63.3	68.9	71.8	1.2×

Table 5. Experiments with using the same backbone or three different backbones as feature extractors.

References

1. Benaim, S., Ephrat, A., Lang, O., Mosseri, I., Freeman, W.T., Rubinstein, M., Irani, M., Dekel, T.: Speednet: Learning the speediness in videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9922–9931 (2020)
2. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
3. Dave, I., Gupta, R., Rizve, M.N., Shah, M.: Tclr: Temporal contrastive learning for video representation. arXiv preprint arXiv:2101.07974 (2021)
4. Han, T., Xie, W., Zisserman, A.: Memory-augmented dense predictive coding for video representation learning. In: Proceedings of the European conference on computer vision. pp. 312–329. Springer (2020)
5. Qian, R., Meng, T., Gong, B., Yang, M.H., Wang, H., Belongie, S., Cui, Y.: Spatiotemporal contrastive video representation learning. arXiv preprint arXiv:2008.03800 (2020)
6. Xu, D., Xiao, J., Zhao, Z., Shao, J., Xie, D., Zhuang, Y.: Self-supervised spatiotemporal learning via video clip order prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10334–10343 (2019)