Supplementary Material

Feature	Output	Shape
Input	-	$3 \times 16 \times 112 \times 112$
Low-level	Stage-2	$128 \times 8 \times 14 \times 14$
Mid-level	Stage-3	$256 \times 4 \times 7 \times 7$
High-level	Stage-4	$512 \times 2 \times 4 \times 4$

Table 1: Definitions of multi-level features on R3D-18.

Feature	Output	Shape
Input	-	$3 \times 16 \times 112 \times 112$
Low-level	Mixed-3c	$480 \times 8 \times 14 \times 14$
Mid-level	Mixed-4f	$832 \times 4 \times 7 \times 7$
High-level	Mixed-5c	$1024 \times 2 \times 4 \times 4$

Table 2: Definitions of multi-level features on S3D.

1. Definition of Multi-level Features

R3D-18. For R3D-18 [3], we define the features output from ResNet Stage-2 as low-level $\mathbf{f}_l$, features output from ResNet Stage-3 as mid-level $\mathbf{f}_m$, features output from ResNet Stage-4 as high-level $\mathbf{f}_h$. When input clip of size $16 \times 112 \times 112$, the channel dimension and spatiotemporal resolutions of different level features are shown in Table. 1.

S3D. For S3D [7], we define the features output from Inception Mixed-3c as low-level $\mathbf{f}_l$, features output from Inception Mixed-4f as mid-level $\mathbf{f}_m$, features output from Inception Mixed-5c as high-level $\mathbf{f}_h$. When input clip of size $16 \times 112 \times 112$, the channel dimension and spatiotemporal resolutions of different level features are shown in Table. 2.

2. More Training Details

Data Augmentations. For augmentation on input data, following [1, 2], we apply temporally-consistent random cropping, horizontal flipping, color jittering and Gaussian blurring on each input clip. For temporal augmentation on extracted multi-level features, we employ temporal reverse and random shuffling to construct negative pairs.

More Hyper-parameters. For default setting, we set the temperature parameter τ in contrastive loss to 0.07 following [4], and set the threshold η in $\mathcal{G}_{ins}$ construction to 0.05.

Self-supervised Pretraining. We use R3D-18 [3] and S3D [7] as the backbone networks and no momentum encoder is required in our experiments. For temporal augmentation on multi-level features, we use temporal shuffle and reverse as two typical transformations. For the definition of contrastive pairs, we regard clips from the same video as positive pairs, and those of different videos as negative. Specifically, we randomly sample 32 RGB frames within a video, and uniformly split them into two 16-frame clips with resolution 112×112 to form positive pairs. For the proposed timestamp retrieval, we regard 16-frame clips as short sequences and the 32-frame clips as long sequences. For multi-level feature optimization, we formulate it as a two-stage procedure. In the first few epochs, we only use $\mathcal{L}_{high}$ to optimize high-level features until they could generate reliable soft targets, i.e., $\mathcal{E}_{i,j}$. Then, we jointly use $\mathcal{L}_{high}$ and $\mathcal{L}_{mul}$ for multi-level feature learning. We note that the temporal modeling module is directly applied to multi-level features. The specific definition of low-level and mid-level features for different backbones is listed in the Supplementary Material. During the whole pretraining stage, we use batch size of 256, and set default number of prototypes to 1000 with queue length 1024. In total, we train for 100 epochs on Kinetics-400, and 300 epochs on UCF-101 using ADAM optimizer with an initial learning rate of 10^{-3} and weight decay of 10^{-5}. The learning rate is decayed by 10 at 70 epochs for Kinetics-400, and 200 epochs for UCF-101.

Action Recognition. For action recognition task, we initialize the backbone with pretrained model parameters except for the last fully-connected classification head. There are two settings for this task: 1) Only use the pretrained model for initialization and finetune the whole network in a fully supervised manner (denoted as *finetune*); 2) Use the pretrained model as feature extractor and only train the linear classifier (denoted as *linear probe*). For evaluation, following [8, 6], we uniformly sample 10 clips for each video, then center crop and resize them to 112×112. The final prediction of each video is the average softmax probabilities of each clip. Performance is measured by Top-1 accuracy.

Video Retrieval. For video retrieval task, we directly use

$\mathcal{L}_{aug}$	$\mathcal{L}_{ret}$	UCF	HMDB
✗	✗	55.9	28.1
✓	✗	59.3	30.7
✗	✓	60.4	31.9
✓	✓	**63.2**	**33.4**

Table 3: Ablation study on temporal modeling loss.

the pretrained model as a feature extractor without any finetuning. Following [8, 5], we select videos in test set of UCF-101 and HMDB-51 as query, and aim to retrieve k-nearest neighbors in the training set. We employ the cosine similarity in feature space to measure the similarity, and use Top-k recall (denoted as R@k) for evaluation.

3. Temporal Modeling Loss

Since there are two loss terms used in temporal modeling, i.e., $\mathcal{L}_{aug}$ and $\mathcal{L}_{ret}$. We explore the efficacy of each term in Table. 3. The model is pretrained on Kinetics-400 with R3D-18 as backbone, the linear probe performance on UCF-101 and HMDB-51 shows that both two terms contribute to more robust temporal modeling. And jointly exploiting these two brings further improvement.

References

[1] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. *arXiv preprint arXiv:2008.01065*, 2020.

[2] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. *arXiv preprint arXiv:2010.09709*, 2020.

[3] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 3154–3160, 2017.

[4] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[5] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video cloze procedure for self-supervised spatio-temporal learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11701–11708, 2020.

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

[7] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 305–321, 2018.

[8] 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*, pages 10334–10343, 2019.