Supplementary for ASCNet: Self-supervised Video Representation Learning with Appearance-Speed Consistency

In this supplementary, we shall provide more implementation details and experimental results of our proposed ASCNet. We organize the supplementary materials as follows.

- In Section A, we discuss the architectures of the projection and predictor in ASCNet.
- In Section B, we provide the configurations for downstream action recognition and video retrieval tasks.
- In Section C, we show the implementation details of the memory bank for appearance-based feature retrieval.
- In Section D, we provide more experimental results of our ASCNet under linear evaluation protocol.
- In Section E, we show the detailed structure of different video encoder backbones applied in ASCNet.

A. Architectures of Projection and Predictor

We apply 3D convolutional residual network [5] with 18 layers (3D R18) as the default video encoder $f(\cdot, \theta)$. We also consider R(2+1)D [13] and S3D-G [14]. Specifically, we use the output of the final average pooling layer as the representation of the corresponding video clip. The output dimension is 512 for 3D R18 and R(2+1)D, and 1024 for S3D-G. Similar to SimCLR [1], we conduct projections $g_a(\cdot, \theta_a)$ and $g_m(\cdot, \theta_m)$ to the task corresponding representations, respectively. These projections are both *multi-layer perceptrons* (MLP) and have the same architecture, as shown in Figure 1. The MLP contains a linear layer to project the representation to a dimension of 4096. After a batch normalization [7] and a rectified linear unit (ReLU) [9] layer, we apply another linear layer to project the representation to a dimension of 256 as the final projection output. The architectures of predictors $h_a(\cdot, \theta'_a)$ and $h_m(\cdot, \theta'_m)$ are the same as $g_a(\cdot, \theta_a)$.

B. Configurations for Downstream Tasks

To evaluate the pre-trained representation for the downstream action recognition task, we remove the projection and add a linear layer with the dimensions of input and output fitting specific tasks. There are 101 classes in UCF-101 dataset [11] and 51 in HMDB-51 dataset [8]. For the video retrieval task, we use the raw outputs. The details are shown in Table 1.

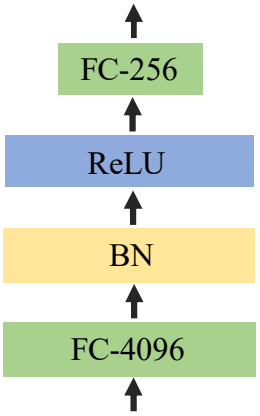

Figure 1: **Structure of the projection and predictor.** FC denotes fully-connected layer and the number beside it means its output dimension. BN denotes batch normalization. ReLU denotes rectified linear units.

Downstream task	Backbone	Input	Output
Action recognition	3D R18	512	num_class
	R(2+1)D	512	
	S3D-G	1024	
Video retrieval	3D R18	512	$*$
	R(2+1)D	512	
	S3D-G	1024	

Table 1: **Configuration of the linear layer**. The numbers in Input column are the dimensions of the linear layer. num_class denotes the number of actions in the target dataset. $*$ indicates that the output dimensions is the same as input dimensions for video retrieval task.

C. More Details of Memory Bank

Inspired by MoCo [6], we adopt the memory bank to facilitate the appearance-based feature retrieval strategy described in section 3.3.1. Instead of collecting negative samples, we use the memory bank to retrieve positive samples for SCP task. The memory bank is a dictionary that maps the appearance features to their corresponding motion features. During the training process, we extract the appearance features from the target network and find the indices of the most similar (measured by dot-product similarity) appearance features in the memory bank. Then we use the indices to retrieve corresponding motion features for SCP task. For each iteration, we update the memory bank with both appearance and motion features extracted from the target network and keep the most recent K samples. We use $K = 32768$ in all our experiments. We maintain two separate memory banks for different playback speeds.

D. Linear Protocol for Action Recognition

Apart from fine-tuning all layers in the network on UCF-101 [11] and HMDB-51 [8] as in Section 4.4, we also provide the results under common linear protocol [6, 1, 2, 10] for action recognition tasks in Table 2. In this protocol, the entire video encoder is frozen, and only a single linear layer is trained with cross-entropy loss. We train the linear classifier on top of the video backbone for 100 epochs with an initial learning rate of 0.1. We use SGD as optimizer and the weight decay of SGD is set to 0, following the same settings as MoCo [6]. ASCNet shows competitive performance compared with single and multi-modal methods [3, 4], by using only RGB modality on Kinetics-400.

Method	Date	Dataset	Res.	Arch.	Mod.	UCF	HMDB
CBT [12]	2019	K600+	112	S3D-G	R	54.0	29.5
MemDPC [3]	2020	K400	224	3D R34	R+F	54.1	30.5
CoCLR [4]	2020	K400	128	S3D-G	R+F	74.5	46.1
ASCNet		K400	112	3D R18	R	68.2	44.9

Table 2: Comparison with state-of-the-art approaches. We report the top-1 accuracy (%) on linear evaluation for action recognition task on UCF-101 and HMDB-51 datasets. For input, 'R' refers to RGB only, 'F' is optical flow.

E. Structures of Video Encoders

The flexibility of ASCNet allows us to study a variety of video backbones. In this paper, we encode a video sequence using various common architectures as backbones. We show the detailed structure of different video backbones used in the experiments: R(2+1)D [13] (Table 3) and 3D ResNet-18 [5] (Table 4). We refer the readers to [14] for the detailed structure of S3D-G. We denote spatio-temporal size by $T \times S^2$ where T is the temporal length and S is the height and width of a square spatial crop.

Stage	Layer	Output size
raw	-	16 ×112 ×112
conv_1	1×7×7, 83, stride 1, 2, 2 3×1×1, 64, stride 1, 1, 1	16×56×56
res_2	[1×3×3, 144 3×1×1, 64 1×3×3, 144 3×1×1, 64]	16×56×56
res_3	[1×3×3, 230 3×1×1, 128 1×3×3, 288 3×1×1, 128]	8×28×28
res_4	[1×3×3, 460 3×1×1, 256 1×3×3, 576 3×1×1, 256]	4×14×14
res_5	[1×3×3, 921 3×1×1, 512 1×3×3, 1152 3×1×1, 512]	2×7×7
global average pool, fc		1×1×1

Table 3: The structure of the video encoder $f(\cdot;\theta)$ with R(2+1)D. Note that both output size and kernel size are in T×W×H shape.

Stage	Layer	Output size
raw	-	16 ×112 ×112
conv_1	7×7×7, 64, stride 1, 2, 2	16×56×56
pool_1	3×3×3 max, stride 2, 2, 2	8×28×28
res_2	[3×3×3, 64 3×3×3, 64] ×2	8×28×28
res_3	[3×3×3, 128 3×3×3, 128] ×2	4×14×14
res_4	[3×3×3, 256 3×3×3, 256] ×2	2×7×7
res_5	[3×3×3, 512 3×3×3, 512] ×2	1×4×4
global average pool, fc		1×1×1

Table 4: The structure of the video encoder $f(\cdot;\theta)$ with 3D ResNet-18. Note that both output size and kernel size are in T×W×H shape.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020. 1, 2

[2] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In *NeurIPS*, 2020. 2

[3] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *ECCV*, pages 312–329, 2020. 2

[4] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *NeurIPS*, 2020. 2

[5] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, pages 6546–6555, 2018. 1, 2

[6] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9726–9735, 2020. 2

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 1

[8] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso A. Poggio, and Thomas Serre. HMDB: A large video database for human motion recognition. In *ICCV*, pages 2556–2563, 2011. 1, 2

[9] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010. 1

[10] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge J. Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. *arXiv preprint arXiv:2008.03800*, 2020. 2

[11] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 1, 2

[12] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*, 2019. 2

[13] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pages 6450–6459, 2018. 1, 2

[14] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, pages 318–335, 2018. 1, 2