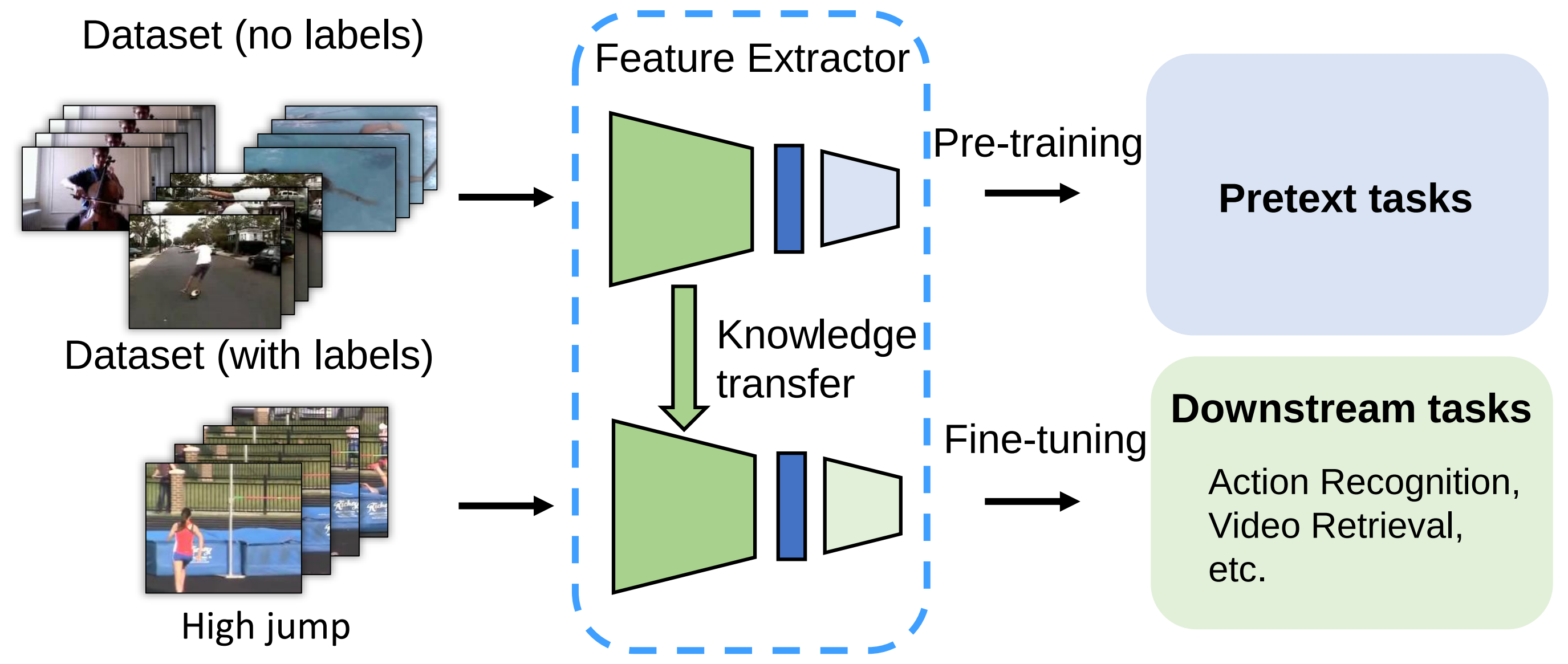
ASCNet: Self-supervised Video Representation Learning with Appearance-Speed Consistency

Deng Huang[†], Wenhao Wu[†], Weiwen Hu, Xu Liu, Dongliang He, Zhihua Wu, Xiangmiao Wu, Mingkui Tan[*] and Errui Ding
sehuangdeng@mail.scut.edu.cn, wuwenhao17@mails.ucas.edu.cn

2021 ICCV OCTOBER 11-17 VIRTUAL

Background

Self-supervised video representation learning aims to learn video features from unlabeled videos. The learned model weights can be transferred to downstream tasks, such as action recognition.



Challenges

- Videos contains unstructured and noisy visual information. It is hard to learn all information with single task.
- Videos are unlabeled. It is hard to find sufficient supervision for model training.

Novelty & Contributions

- We propose the ACP and SCP tasks for unsupervised video representation learning. In this sense, negative samples no longer affect the quality of learned representations, making the training more robust.
- We propose the appearance-based feature retrieval strategy to select the more effective positive sample for speed consistency perception. In this way, we can bridge the gap between two pretext tasks.
- We verify the effectiveness of our method for learning meaningful video representations on two downstream tasks, namely, action recognition and retrieval, on the UCF-101 and HMDB51 datasets. In all cases, we demonstrate state-of-the-art performance over other self-supervised methods.

Model Overview

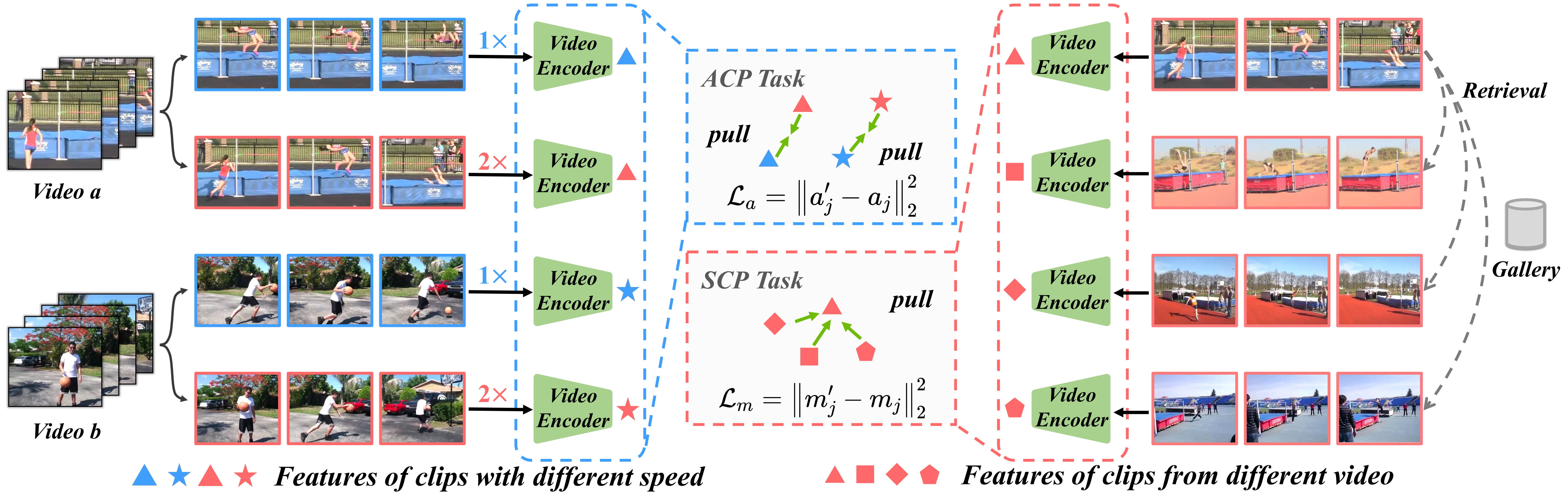

Video a
Video b
1×
2×
1×
2×
Video Encoder
ACP Task
pull
pull
$\mathcal{L}_a = \left\|a'_j - a_j\right\|_2^2$
SCP Task
pull
$\mathcal{L}_m = \left\|m'_j - m_j\right\|_2^2$
Retrieval
Gallery
▲ ★ ▲ ★ Features of clips with different speed
▲ ■ ◆ ⬟ Features of clips from different video

Illustration of the proposed framework. Given a set of video clips with different playback speed (*i.e.*, 1× and 2×), we use a video encoder $f(\cdot;\theta)$ to map the clips into appearance and speed embedding space. For the ACP task, we pull the appearance features from the same video closer. For the SCP task, we first retrieve the same speed video with similar contents and then pull the speed features closer. All of the video encoders share the parameters.

Experiments on Action Recognition

Method	Date	Backbone	UCF	HMDB
CBT	2019	S3D-G	79.5	44.6
SpeedNet	2020	S3D-G	81.1	48.8
Pace	2020	S3D-G	87.1	52.6
CoCLR-RGB	2020	S3D-G	87.9	54.6
RSPNet	2021	S3D-G	89.9	59.6
Ours		3D R18	80.5	52.3
Ours		S3D-G	**90.8**	**60.5**
Supervised (K400)		3D R18	84.4	56.4
Supervised (ImageNet)		S3D-G	86.6	57.7
Supervised (K400)		S3D-G	96.8	75.9

Table 1: Comparison with state-of-the-art self-supervised learning methods on the UCF-101 and HMDB-51 datasets. Our ASCNet outperforms the ImageNet supervised pre-trained model over two datasets (90.8% vs. 86.6%, 60.5% vs. 57.7%)

Experiments on Video Retrieval

Method	Architecture	Top-k		
		$k=1$	$k=10$	$k=50$
OPN	CaffeNet	19.9	34.0	51.6
Buchler *et al.*	CaffeNet	25.7	42.2	59.5
ClipOrder	3D R18	14.1	40.0	66.5
SpeedNet	S3D-G	13.0	37.5	65.0
VCP	3D R18	18.6	42.5	68.1
	R(2+1)D	19.9	42.0	64.4
Pace	3D R18	23.8	46.4	69.8
	C3D	31.9	59.2	80.2
RSPNet	C3D	36.0	66.5	87.7
	3D R18	41.1	68.4	88.7
Ours	3D R18	**58.9**	**82.2**	**93.4**

Table 2: Comparison with state-of-the art methods for nearest neighbor retrieval task on the UCF-101 dataset as measured by the top-k retrieval accuracy (%).

Our Methods

Algorithm 1 Training method of ASCNet.

```
Require: Video set $\mathcal{V}=\{v_i\}_{i=1}^N$, the encoder $f$ with paramaters $\theta$ or $\xi$, the projection heads $g_a$ and $g_m$ with parameters $\theta_a$, $\xi_a$, $\theta_m$ and $\xi_m$, the predictors $h_a$ and $h_m$ with parameters $\theta'_a$ and $\theta'_m$, the hyperparameter $\gamma$.
Randomly initialize parameters $\theta, \theta_a, \theta_m, \theta'_a, \theta'_m$.
Initialize parameters $\xi \leftarrow \theta, \xi_a \leftarrow \theta_a, \xi_m \leftarrow \theta_m$.
while not convergent do
    Randomly sample a video $v$ from $\mathcal{V}$.
    Sample two clips $c_i, c_j$ from $v$.
    Extract features $\mathbf{x}_i=f(c_i,\theta)$, $\mathbf{x}_j=f(c_j,\xi)$.
    // Learn appearance features with ACP task
    Obtain $\mathbf{a}_i=g_a(\mathbf{x}_i,\theta_a)$, $\mathbf{a}_j=g_a(\mathbf{x}_j,\xi_a)$.
    Obtain $\mathbf{a}'_i = h_a(\mathbf{a}_i,\theta'_a)$.
    Compute $\mathcal{L}_a = \|\mathbf{a}'_i - \mathbf{a}_j\|_2^2$.
    // Conduct appearance-based feature retrieval
    Construct $\mathcal{C}=\{\mathbf{a}_t\}_{t=1}^{N-1}$ using $g_a(\cdot,\theta_a)$ from $\mathcal{V}\setminus v$.
    // Obtain $\mathcal{C}$ efficiently from memory bank
    Select the video $\hat{v}$ corresponding to features $\mathbf{a}\in\mathcal{A}$ with the highest dot product similarity with $\mathbf{a}_j$.
    Sample one clip $c_k$ from $\hat{v}$.
    Extract features $\mathbf{x}_k=f(c_k,\xi)$.
    // Learn speed features with SCP task
    Obtain $\mathbf{m}_i=g_m(\mathbf{x}_i,\theta_m)$, $\mathbf{m}_k=g_m(\mathbf{x}_k,\xi_m)$.
    Obtain $\mathbf{m}'_i=h_m(\mathbf{x}_i,\theta'_m)$.
    Compute $\mathcal{L}_m=\|\mathbf{m}'_i - \mathbf{m}_k\|_2^2$.
    Compute $\mathcal{L}=\gamma\mathcal{L}_m+(1-\gamma)\mathcal{L}_a$.
    Update parameters $\theta, \theta_a, \theta_m, \theta'_a, \theta'_m$ via SGD.
    Compute exponential moving average $\xi, \xi_a, \xi_m$.
end while
```

Contact Information

- Deng Huang
  South China University of Technology
  sehuangdeng@mail.scut.edu.cn
- Wenhao Wu
  Baidu Inc.
  wuwenhao17@mails.ucas.edu.cn