Supplementary Materials

Jiangliu Wang[1], Jianbo Jiao[2], and Yun-Hui Liu[1]

[1] The Chinese University of Hong Kong
[2] University of Oxford
{jlwang,yhliu}@mae.cuhk.edu.hk , jianbo@robots.ox.ac.uk

1 Overview

In this supplementary material we provide:

- More ablation studies on the pace prediction task design in Sec. 2
- Algorithm implementation details in Sec. 3
- More qualitative results of attention maps with different paces in Sec. 4
- A video file **suppleVideo.mp4** illustrating the basic idea of pace prediction. Note that we randomly sample videos with different paces from the UCF101 dataset [4]. See Table 1 for more details on the pace prediction accuracy.

2 Additional Ablation Studies on Pace Prediction Task

Here we provide additional ablation studies on the design of the pace prediction task, including (1) Pace prediction performance (Table 1). (2) Evaluation of the performance on *slow* pace as described in our paper (Table 2). (3) Investigation on different pace steps (Table 3). (4) Analysis on video play direction, *i.e.*, forwards or backwards (Table 4).

Pace prediction accuracy. We report the pretext task performance (*i.e.*, pace prediction accuracy) and the downstream task performance (*i.e.*, action recognition accuracy) on UCF101 dataset in Table 1. It can be seen from the table that with the increase of the maximum pace, the pretext task becomes harder for the network to solve, which leads to degradation of the downstream task. This further validate our claim in the paper that a pretext task should be neither too simple nor too ambiguous.

Table 1. Pace prediction accuracy w.r.t. different pace design.

Pre-training	Method	# Classes	Pace rea. acc.	UCF acc.
×	Random	-	-	56.0
✓	$p = [1, 3]$	3	77.6	71.4
✓	$p = [1, 4]$	4	69.5	**72.0**
✓	$p = [1, 5]$	5	61.4	72.0
✓	$p = [1, 6]$	6	55.9	71.1

Slow pace. In our paper, we propose two different methods to generate video clips with slow pace: replication of previous frames or interpolation with existing algorithms [1]. We choose the replication in practice as most modern interpolation algorithms are based on supervised learning, while our work focuses on self-supervised learning, forbidding us to use any human annotations.

As shown in Table 2, compared with normal and *fast* paces, if we use normal and *slow* paces, the performance of the downstream task decreases (73.9→72.6). While when combining with both slow and fast pace (*absolute* pace as described in the paper), no performance change is observed, which again validates our choice of the pace configuration.

Table 2. Evaluation of slow pace.

Config.	Pace	# Classes	UCF10 Acc.
Baseline	[1,2,3,4]	4	73.9
Slow	$[\frac{1}{4},\frac{1}{3},\frac{1}{2},1]$	4	72.6
Slow-fast	$[\frac{1}{3},\frac{1}{2},1,2,3]$	5	73.9

Pace step. Based on the better performance achieved by the fast pace as shown above, we take a closer look into the fast pace design, by considering different interval steps, *i.e.*, frame skip. For simplicity, in the paper we showcase with the step that equals one (baseline) between each fast pace where the paces are {1,2,3,4}. Here we further explore the interval steps of two and three so as to introduce larger motion dynamics into the learning process. It can be observed from Table 3 that by increasing the interval steps, performance could be further improved, but tends to saturate when the step is too large.

Table 3. Evaluation of different pace steps.

Step	Pace	# Classes	UCF10 Acc.
1	[1,2,3,4]	4	73.9
2	[1,3,5,7]	4	74.9
3	[1,4,7,10]	4	74.7

Forwards *v.s.* backwards. It has been a long standing problem that whether a forward played video can be considered as the same as its backward played version, in self-supervised video representation learning. Some works [3,2] argue that these two versions should be attributed to the same semantic labels, while Wei *et al.* prone to distinguish the forwards and backwards video [5]. In the following, we investigate these two opinions based on our method as shown in Table 4.

As for the random backwards with four classes, we consider forwards and backwards videos as the same pace samples, while for backwards with eight classes, they are considered to be different samples. It can be seen from the table that, both configurations achieve lower performance than our baseline. We suspect the reason is that to distinguish the backwards from forwards, it is essentially a video order prediction task though in some order prediction work [2,3] they are considered to be the same. When combing the proposed pace reasoning task with such an order prediction task, the network will be confused towards an ambiguous target. As a result, the downstream task performance is deteriorated.

Table 4. Evaluation of video forwards *v.s.* backwards.

Config.	Pace	# Classes	UCF10 Acc.
Baseline	[1,2,3,4]	4	73.9
Rnd backwards	[1,2,3,4]	4	73.0
Backwards	[±1, ±2, ±3, ±4]	8	73.7

3 Implementation Details

Here we present the algorithms of the proposed approach, with two possible solutions as mentioned in the paper: pace prediction with contrastive learning on same video context and pace prediction with contrastive learning on same video pace.

4 Attention Visualization

Finally, we provide the attention map visualization on more video clips with different paces. Starting from the same initial frame, we sample 16-frames clips with different paces $p = 1, 2, 3, 4$. Then we show the attention maps for every 3 frames. Note that only one attention map is generated based on a 16-frame video clip. It can be seen from Fig. 1, clips with larger pace p contain larger motion dynamics as they span more frames. The attention maps are also becoming active in larger motion areas with the increase of pace p.

Algorithm 1 Pace prediction with contrastive learning on same video context.

Input: Video set X, pace transformation $g_{pac}(.)$, λ_{cls}, λ_{ctr}, backbone network f.
Output: Updated parameters of network f.

```
for sampled mini-batch video clips $\{x_1, \ldots, x_N\}$ do
    for $i = 1$ to $N$ do
        Random generate video pace $p_i$, ${p_i}'$
        $\widetilde{x}_i = g_{pac}(x_i|p_i)$
        $\widetilde{x}'_i = g_{pac}(x_i|{p_i}')$
        $z_i = f(\widetilde{x}_i)$
        ${z_i}' = f(\widetilde{x}'_i)$
    end for
    for $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ do
        $\text{sim}(z_i, z_j) = {z_i}^\top z_j$
    end for
    Define $\mathcal{L}_{ctr\_sc} = -\frac{1}{2N} \sum\limits_{i,\mathcal{J}} \log \frac{\exp(\text{sim}(z_i,{z_i}'))}{\sum\limits_{i} \exp(\text{sim}(z_i,{z_i}'))+\sum\limits_{i,\mathcal{J}} \exp(\text{sim}(z_i,z_{\mathcal{J}}))}$.

    $\mathcal{L}_{cls} = -\frac{1}{2N} \sum \sum\limits_{i=1}^{M} y_i (\log \frac{\exp(h_i)}{\sum_{j=1}^{M} \exp(h_j)})$
    $\mathcal{L} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{ctr}\mathcal{L}_{ctr\_sc}$
    Update $f$ to minimize $\mathcal{L}$
end for
```

Algorithm 2 Pace prediction with contrastive learning on same video pace.

Input: Video set X, pace transformation $g_{pac}(.)$, λ_{cls}, λ_{ctr}, backbone network f.
Output: Updated parameters of network f.

```
for sampled mini-batch video clips $\{x_1, \ldots, x_N\}$ do
    for $i = 1$ to $N$ do
        Random generate video pace $p_i$
        $\widetilde{x}_i = g_{pac}(x_i|p_i)$
        $z_i = f(\widetilde{x}_i)$
    end for
    for $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, N\}$ do
        $\text{sim}(z_i, z_j) = {z_i}^\top z_j$
    end for
    Define $\mathcal{L}_{ctr\_sp} = -\frac{1}{N} \sum\limits_{i,j,k} \log \frac{\exp(\text{sim}(z_i,z_j))}{\sum\limits_{i,j} \exp(\text{sim}(z_i,z_j))+\sum\limits_{i,k} \exp(\text{sim}(z_i,z_k))}$

    $\mathcal{L}_{cls} = -\frac{1}{N} \sum \sum\limits_{i=1}^{M} y_i (\log \frac{\exp(h_i)}{\sum_{j=1}^{M} \exp(h_j)})$
    $\mathcal{L} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{ctr}\mathcal{L}_{ctr\_sp}$
    Update $f$ to minimize $\mathcal{L}$
end for
```

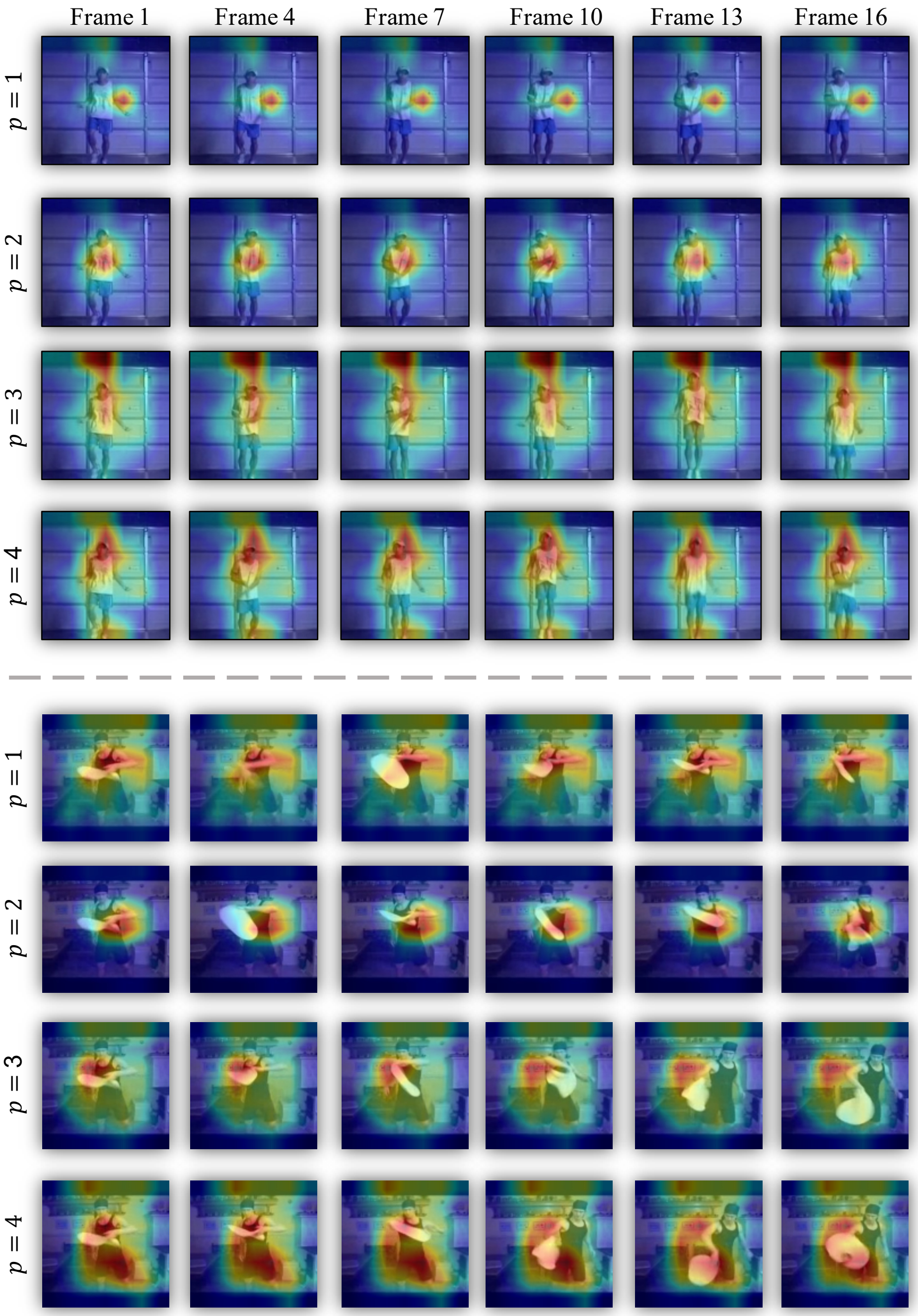

Fig. 1. Attention visualization (using tool from [6]) of the conv5 layer from self-supervised pre-trained model.

References

1. Jiang, H., Sun, D., Jampani, V., Yang, M.H., Learned-Miller, E., Kautz, J.: Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In: CVPR. pp. 9000–9008 (2018)
2. Lee, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Unsupervised representation learning by sorting sequences. In: ICCV. pp. 667–676 (2017)
3. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and learn: unsupervised learning using temporal order verification. In: ECCV. pp. 527–544. Springer (2016)
4. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
5. Wei, D., Lim, J.J., Zisserman, A., Freeman, W.T.: Learning and using the arrow of time. In: CVPR. pp. 8052–8060 (2018)
6. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In: ICLR (2017)