Supplementary Material:
Learning by Aligning Videos in Time

Sanjay Haresh∗  Sateesh Kumar∗  Huseyin Coskun  Shahram N. Syed
Andrey Konin  M. Zeeshan Zia  Quoc-Huy Tran

Retrocausal, Inc.
Seattle, WA

www.retrocausal.ai

Figure S1: Kendall’s Tau results by TCC.

In this supplementary material, we first present results on a subset of 11 actions of Penn Action in Sec. 1 and fine-grained frame retrieval results on IKEA ASM in Sec. 2. We then show training-from-scratch results in Sec. 3 and ablation results of α, σ, and p in Sec. 4. Next, in Sec. 5 we show results of combining LAV with TCC and TCN while we show results on a recent frame-shuffling method in Sec. 6. Moreover, we visualize our embeddings and provide our labels for Penn Action in Secs. 7 and 8 respectively. Finally, we describe our additional implementation details in Sec. 9.

1. Results on a Subset of 11 Actions of Penn Action

Among the 3 datasets that we use in Sec. 5 of the main paper (i.e., Pouring, Penn Action, and IKEA ASM), we notice that for Penn Action while TCC performs well on most actions, it struggles on 2 actions i.e., Baseball Swing and Tennis Forehand. As we can see in Fig. S1, the Kendall’s Tau results by TCC on the above 2 actions (red and orange curves) do not go higher than ∼ 0.2, which hurts its overall performance on Penn Action in Tabs. 2-4 of the main paper. This might be due to the fact that the beginning and ending frames of the above 2 actions are visually similar, and TCC

Figure S2: Example alignment errors by TCC. On the left is the reference frame in one video, and on the right is the aligned frame in another video by TCC (i.e., via nearest neighbor search in the embedding space). The frame on the left is among the beginning frames, while the frame on the right in among the ending frames. TCC incorrectly aligns them together due to their similar appearances/poses.

∗ indicates joint first author.
{sanjay,sateesh,huseyin,shahram,andrey,zeeshan,huy}@retrocausal.ai
Table S1: Phase classification, phase progression, and Kendall’s Tau results on a subset of 11 actions of Penn Action. Best results are in bold, while second best ones are underlined.

| Method      | Classification | Progress | $\tau$ |
|-------------|----------------|----------|--------|
| PENN ACTION | 82.03          | 0.7054   | 0.7783 |
| SAL [4]     | 82.99          | 0.7281   | 0.8275 |
| TCC [2]     | 83.94          | 0.7394   | 0.8001 |
| LAV (Ours)  | **84.47**      | 0.6654   | 0.8149 |
| LAV+TCC (Ours) | 84.14   | 0.7111   | 0.7892 |

Table S2: Fine-grained frame retrieval results on a subset of 11 actions of Penn Action. Best results are in bold, while second best ones are underlined.

| Method      | AP@5  | AP@10 | AP@15 |
|-------------|-------|-------|-------|
| PENN ACTION | 76.83 | 76.52 | 76.35 |
| SAL [4]     | 78.09 | 77.74 | 77.52 |
| TCC [2]     | **79.49** | **79.19** | **79.00** |
| LAV (Ours)  | 79.34 | **79.23** | **79.08** |
| LAV+TCC (Ours) | 79.29 | 79.15 | 79.03 |

Table S3: Fine-grained frame retrieval results on IKEA ASM. Best results are in bold, while second best ones are underlined.

| Method      | AP@5  | AP@10 | AP@15 |
|-------------|-------|-------|-------|
| SAL [4]     | 15.15 | 14.90 | 14.72 |
| TCN [5]     | 19.15 | 19.19 | 19.33 |
| TCC [2]     | 19.80 | 19.64 | 19.68 |
| LAV (Ours)  | **23.89** | **23.65** | **23.56** |
| LAV+TCC (Ours) | 22.95 | 22.80 | 22.70 |

Table S4: Training-from-scratch results on Pouring. Best results are in bold, while second best ones are underlined.

| Method      | Classification | Progress | $\tau$ |
|-------------|----------------|----------|--------|
| PENN ACTION | 85.86          | 0.6422   | 0.7329 |
| SAL [4]     | 85.98          | 0.6732   | 0.7500 |
| TCC [2]     | **88.59**      | 0.7104   | 0.7774 |
| LAV (Ours)  | 87.70          | **0.7320** | **0.7867** |

2. Fine-Grained Frame Retrieval Results on IKEA ASM

We now conduct fine-grained frame retrieval experiments on IKEA ASM and report the quantitative results of different self-supervised methods in Tab. S3. It is evident from the results that LAV consistently achieves the best performance across different values of $K$, outperforming other methods by significant margins. For example, for AP@5, LAV obtains 23.89%, while TCC, TCN, and SAL get 19.80%, 19.15%, and 15.15% respectively. Furthermore, the combined LAV+TCC leads to significant performance increase over TCC. For instance, for AP@5, LAV+TCC achieves 22.95%, while TCC obtains 19.80%. The above observations on IKEA ASM are similar to those on Penn Action and Pouring reported in Sec. 5.4 of the main paper, confirming the utility of our self-supervised representation for fine-grained frame retrieval.

3. Training-from-Scratch Results

All of the experiments in Sec. 5 of the main paper utilize an encoder network initialized with pre-trained weights from ImageNet classification. For completeness, we now experiment with learning from scratch. We use a smaller backbone network, i.e., VGG-M [1], (instead of ResNet-50) for this experiment. Tab. S4 shows the quantitative results of different self-supervised methods when learning from scratch on Pouring. It can be seen from Tab. S4 that the performance of all methods drops as compared to Tabs. 2 and 3 of the main text. Moreover, SAL and TCN are inferior to TCC and LAV across all metrics. Lastly, although
LA V has slightly lower classification accuracy than TCC, LA V outperforms TCC on both progression and Kendall’s Tau.

Next, Tab. S5 shows training-from-scratch results on Penn Action, using a single joint model for all actions (similar as Sec. 5.5 of the main paper). For all methods, the performance in Tab. S5 is lower than Tab. 5 of the main text. Also, SAL and TCN are inferior to TCC and LA V. TCC performs the best on progression, while LA V performs the best on the other two metrics.

Finally, we obtain training-from-scratch results on IKEA ASM, which show LA V achieves the best performance (i.e., for classification, 23.84 for LA V vs. 22.04, 20.45, and 20.42 for TCC, TCN, and SAL respectively).

4. Ablation Results of \( \alpha \), \( \sigma \), and \( p \)

We first present ablation results of \( \alpha \) on Pouring in Fig. S3(a). We observe that the performance is generally stable across values of \( \alpha \), and \( \alpha = 1.0 \) yields the best results. Next, Figs. S3(b) and S3(c) illustrate ablation results of \( \sigma \) and \( p \) respectively on Pouring. From the results, the performance is generally stable across values of \( \sigma \) and \( p \). Particularly, \( \sigma = 15 \) performs the best, and large \( p \) is preferred.

5. Performance of LA V+TCC and LA V+TCN

We note that LA V+TCC does not consistently perform better than LA V in Tabs. 2 and 3 of the main paper. This might be attributed to the fact that LA V works on L2-normalized embeddings while TCC does not. Since the two components operate on different embedding spaces, combining the two might not always lead to better results.

In addition, we evaluate LA V+TCN on Pouring. We notice LA V+TCN suffers from the same problem as LA V+TCC (i.e., normalized/unnormalized embeddings). LA V+TCN obtains 91.22, 0.7866, and 0.7925 for classification, progression, and Kendall’s Tau respectively, which are comparable to TCN but lower than LA V.

6. Performance of a Recent Frame-Shuffling Method

We evaluate the clip order prediction (COP) method of Xu et. al. [6] on Pouring. As it is a clip-based method, we use sliding windows to generate embeddings for frames at window centers. As mentioned in Sec. 4 of the main paper, the network is first trained for the pretext task and then frozen while we train SVM classifier/linear regressor for the main tasks. It achieves 79.44, 0.5309, and 0.6656 for classification, progression, and Kendall’s Tau respectively, which are lower than SAL in Tabs. 2 and 3 of the main paper. This is likely because the pretext task (i.e., COP) is clip-based, whereas the main tasks are frame-based and require capturing fine-grained frame-based details. Further, since we freeze the network while training SVM classifier/linear regressor, it could not disregard irrelevant clip-based details to focus on the one frame that matters.

7. Visualization of Embeddings

We present the t-SNE visualization [3] of the embeddings learned by LA V on 3 example actions of Penn Action in Fig. S4. For each action, we show 4 videos with each plotted using a unique color. In addition, we use different shades of the same color to distinguish different frames of the same video, i.e., beginning frames have light shades, while later frames have progressively darker shades. The visualization in Fig. S4 shows that LA V encodes each video as an overall smooth trajectory in the embedding space, where temporally close frames are mapped to nearby points in the embedding space and vice versa. Moreover, corresponding frames from different videos are generally aligned in the embedding space, e.g., points of different colors but similar shades are nearby in the embedding space and vice versa. We also sample one random time-step (highlighted by a black circle), and plot corresponding frames from different videos (each bordered by a distinct color), which are shown to belong to the same action phase. The above observations show the potential application of our self-supervised representation for temporal video alignment.

8. Labels for Penn Action

We have made our dense per-frame labels for 2123 videos of Penn Action publicly available at https://bit.ly/3f73e2W. Please refer to Tab. 2 of TCC for
| Hyperparameter               | Value                                      |
|-----------------------------|--------------------------------------------|
| # of sampled frames ($p$)   | 40 (P), 20 (PA, IA)                        |
| Batch size                  | 1 (P), 2 (PA, IA)                          |
| Learning rate               | $10^{-4}$                                  |
| Weight decay                | $10^{-5}$                                  |
| Soft-DTW smoothness ($\gamma$) | 0.1                                      |
| Window size ($\sigma$)      | 15 (P, IA), 7 (PA)                         |
| Margin ($\lambda$)          | 2                                         |
| Regularization weight ($\alpha$) | 1.0 (P), 0.5 (PA, IA)                  |
| # of context frames ($k$)   | 1                                         |
| Context stride              | 15 (P, PA), 8 (IA)                         |

Table S6: Hyperparameter settings for LAV. Here, P denotes Pouring, PA represents Penn Action, and IA denotes IKEA ASM. For batch size, 1 means 1 pair of videos (or 2 videos per batch).

For fair evaluations, we use the same data augmentation techniques and encoder networks for all the competing methods. More specifically, we follow the same data augmentation procedures and borrow the encoder networks from TCC [2]. Please refer to the supplementary material of TCC for more details on data augmentation techniques and encoder networks. In addition, we list the hyperparameter settings for our method in Tab. S6. For other methods, we use the same hyperparameter settings suggested by TCC.

9. Implementation Details

For fair evaluations, we use the same data augmentation techniques and encoder networks for all the competing methods. More specifically, we follow the same data augmentation procedures and borrow the encoder networks from TCC [2]. Please refer to the supplementary material of TCC for more details on data augmentation techniques and encoder networks. In addition, we list the hyperparameter settings for our method in Tab. S6. For other methods, we use the same hyperparameter settings suggested by TCC.

References

[1] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014. 2

[2] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1801–1810, 2019. 2, 4

[3] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 3

[4] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016. 2

[5] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1134–1141. IEEE, 2018. 2

[6] 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 Conference on Computer Vision and Pattern Recognition*, pages 10334–10343, 2019. 3
Figure S4: Visualization of embeddings for LAV.