Supplementary Material

LA Specific Parameters

For the LA-specific parameters, we use cluster size $m = 8000$ for constructing close neighbors $\mathbf{C}_i$ and nearest neighbor size $k = 512$ for constructing background neighbors $\mathbf{B}_i$. These parameters depart somewhat from the optimal parameters found in [7], due to the substantial difference in size, and thus density in the embedding space, between the Kinetics training set (240K points) and the ImageNet dataset used in [7] (1.2M points).

Network Implementation Details

For VIE-Single, we directly apply the ResNet-18 architecture and follow exactly the same preprocessing pipeline as described in the main text.

For VIE-3DResNet, in order to be comparable to other works [4, 3] which use a smaller input resolution for their networks, we correspondingly scale down our input image size. More specifically, during training, we first resize the chosen frames so that their shortest edges are between 128 and 160px and then get 112×112 images through random crops. We then apply the same color noise and random horizontal flip to get the final inputs to the networks. During testing, the frames are resized so that their shortest side is 128px, and then the center 112×112 crops are chosen as inputs. Same as in [4, 3], the input clip contains 16 consecutive frames.

For VIE-TRN, we sample four consecutive half-second bins, and then one frame from each bin, using ResNet-18 as the shared 2D-CNN across multiple frames, with the outputs of the **Conv5** concatenated channel-wise and input into a fully-connected layer to generate the final embedding. This is a simplified version of the TRN, which runs faster and achieves only slightly lower supervised action-recognition performance than the full 8-frame TRN introduced in [6].

For VIE-Slow and VIE-SlowFast, we follow [1] but modify it to use ResNet-18 rather than ResNet-50. The Slow model/pathway evenly samples one frame from every 16 to assemble a 4-frame input sequence, while the Fast pathway samples one frame from every 4 to assemble a 16-frame input sequence.

Single-frame Models with Multi-frame Inputs

To control for the fact that multi-frame models received more total inputs than single-frame models, we also built models which, for any given multi-frame model, takes VIE-Single model, applies it to multiple frames using the same sampling strategy as for the multi-frame model, and then averages across the per-frame outputs before training the softmax classifier. These models are denoted with Input-Single. And their performance is shown in Table S1.

Models	Conv3	Conv4	Conv5
TRN-Input-Single	25.52	39.25	44.27
Slow-Input-Single	26.17	39.24	44.62
Sf-Input-Single	25.72	39.38	44.29

Table S1: Top-1 transfer learning accuracy (%) on Kinetics for Input-Single models.

Reimplementation Details

We reimplemented OPN [5], RotNet [2], and 3DRotNet [3] methods and train them on Kinetics videos, as controls for VIE. The implementation of OPN follows the procedure described in the paper as closely as possible, including input size, motion-related frame sampling, the use of frame-wise spatial jittering and channel dropping, and the learning rate schedule.

However, for a fair comparison, we use ResNet-18 as the OPN backbone. Our OPN implementation achieves approximately 40% in the order prediction training task on Kinetics, similar to that reported in the original OPN paper, suggesting it is functioning as intended. As for RotNet and 3DRotNet, we use ResNet-18 and 3DResNet-18 as the backbones respectively. The input resolution for 3DResNet-18 is set as 112×112, matching the input resolution of VIE-3DResNet. Other details follow the procedure described in the original papers.

Fine-tuning Implementation Details

In testing for both preprocessing pipelines, each video is split into consecutive 16-frame clips and the outputs of all clips are averaged to get the final prediction. As for other parameters, the initial learning rate is 0.01 and the weight decay is 1e-4 for the training from scratch. For finetuning, the initial learning rate is 0.0005 and the weight decay is 1e-5. The learning rate is dropped by 10 after validation performance saturates. We report the results on the first split for both UCF101 and HMDB51, which should be close to the 3-split average result.

References

[1] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *arXiv preprint arXiv:1812.03982*, 2018. 1

[2] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 1

[3] Longlong Jing, Xiaodong Yang, Jingen Liu, and Yingli Tian. Self-Supervised Spatiotemporal Feature Learning via Video Rotation Prediction. 2018. 1

[4] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-Supervised Video Representation Learning with Space-Time Cubic Puzzles. 2018. 1

[5] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 667–676, 2017. 1

[6] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–818, 2018. 1

[7] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. *arXiv preprint arXiv:1903.12355*, 2019. 1