Self-Supervised Learning of Pose-Informed Latents

Raphaël Jean
Menya Solutions, Mila
raphael.jean@rocketmail.com

Pierre-Luc St-Charles
Mila, AMLRT

Sören Pirk
Google Research

Simon Brodeur
Menya Solutions

1. Supplementary Material

In the following sections, we provide supplemental detail on the network architecture used in our experiments, additional qualitative and quantitative evaluation results for both UCF101 and Objectron, and a discussion on failure cases in pose estimation.

The code used for our experiments is available online.[1]

1.1. Network Architecture

We show in Figure 1 the architecture used in our experiments. In (a), we show how we adapt the SimSiam architecture to train on view pairs from object-centric videos. The *projector* and *predictor* networks of our framework are shown in (b) and (c), respectively. We use 2048-dimension embeddings and a bottleneck structure for the *predictor*. These are in line with the original description in SimSiam but they differ from popular implementations such as PyTorch Lightning's.[2] Note also that in our implementation the *predictor* learning rate is fixed throughout training while the encoder and projector are trained with the learning rate schedule described in the main paper.

1.2. Additional Results

Qualitative Evaluation Results on Objectron. Our preliminary experiments on pose estimation using the Objectron dataset showed that commonly used pre-trained models are insensitive to pose variations due to feature suppression. In contrast with the structured space presented in Figure 2 of the main paper, models pre-trained on ImageNet will collapse different frame embeddings across a video into a single location. Our methodology instead results in an embedding structure that evokes the motion of the camera in the original video sequence. We provide a video that shows this effect along with this document.[3]

The embeddings produced by SimSiam with our proposed training approach also seem to be fairly robust to background clutter and appearance variations while remaining sensitive to object poses. We illustrate this in Figure 2 for a variety of videos. There, for different frames of a given video (shown on the top row of each double-row segment), we find the nearest neighboring frame in the validation set based on embedding similarity. The sequence of corresponding results is shown in the bottom row of each segment. We can observe that although the nearest correspondences sometimes vary in appearance and absolute location in image space, the object orientations (shown using the 3D bounding boxes in green) are almost always comparable.

Quantitative Evaluation Results on UCF101. Next, we provide in Table 1 additional results on the UCF101 action recognition benchmark for different model pre-training strategies. The results for the "Nearby frame pairs" and "Single frame" approaches are already discussed in the main paper. Here, the "Distant frame pairs" approach is added; interestingly, it shows significantly better performance in the single-neighbor retrieval regime (R@1), but much worse performance under other regimes. We also provide the evaluation results when using a model pre-trained on ImageNet in a supervised fashion ("ImageNet embeddings"): in this case, the results far outperform those of self-supervised training strategies. This shows that UCF101 pre-training can be easily eclipsed by supervised pre-training on an image dataset that contains object labels relevant to the UCF101 actions. This provides an interesting upper bound for single-frame representations. Finally, we provide a lower bound on this benchmark based on the embeddings of a randomly initialized model ("Random embeddings").

Note that we use majority voting to compare sequences of image embeddings and avoid the need to reduce the dimensionality of sequences with a temporal pooling operation. This allows to better compute the point-to-point similarity of trajectories in embedding space. For each image embedding of the query sequence, we search for the k-nearest neighbours in the training set and aggregate them. The relative frequencies of the classes associated with the aggregated k-nearest neighbours are then computed. If at least one of the top-k most frequent classes matches the

[1] https://github.com/rjean/siampose/

[2] https://github.com/PyTorchLightning/lightning-bolts/ as of version 0.3.1.

[3] The video is named `pca_3d_trajectory_video.mp4`.

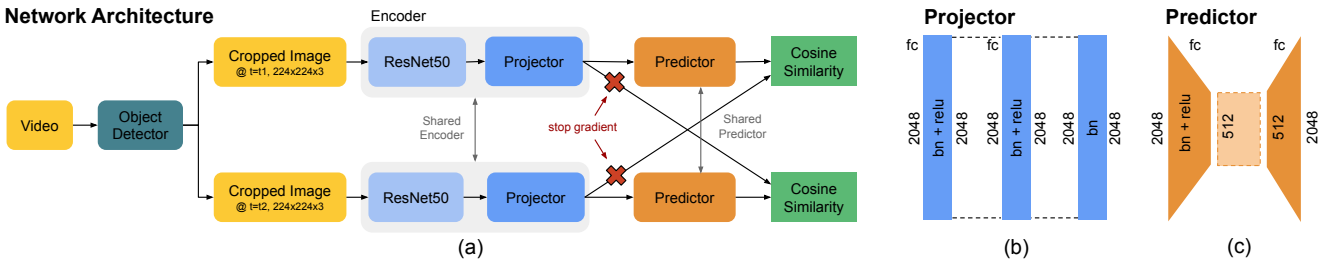

Figure 1. Our network architecture (a) is inspired by the Siamese framework of [1]. We sample frames from a video and use these as views to learn good representations by maximizing the similarity between their corresponding embeddings. The use of video frames makes the learned representations more motion- and geometry-aware which enables their use in tasks such as pose estimation and action recognition. The setups for the projector and predictor blocks are shown in (b) and (c), respectively.

Figure 2. Overview of frame-to-frame nearest neighbor matching results obtained with the Objectron validation set for the bike, book, bottle, camera, cereal box, chair, laptop and shoe categories of Objectron. Each double-row segment has a corresponding high-resolution video provided with this document. Top shows the query frames (white bounding box is ground truth, green and red bounding boxes are the reprojected 3D box from the nearest neighbor). Bottom shows the result frames and the associated 3D bounding boxes in green. The full resolution version of the images in this figure are also provided with this document (as part of the `frame_matching` folder).

class of the query sentence, we consider it a match.

Impact of Data Augmentation Strength. We also study the impact of varying the strength of the data augmentation operations across different view generation strategies on the Objectron dataset. We split our experiments into three main categories: the first category has views generated from a single frame ("Same"), the second has views from nearby frames ("Nearby"), and the last has views from frames uni-

Table 1. UCF-101 nearest neighbor video retrieval performance using different pairing methods, and using backbones pre-trained on other datasets.

Method	R@1	R@5	R@10	R@20
Nearby frame pairs	41.4	**57.3**	**62.1**	**66.2**
Distant frame pairs	**42.6**	54.9	58.8	61.2
Single frame	41.6	57.1	61.7	65.2
Random embeddings	18.1	32.8	41.1	50.1
ImageNet embeddings	66.8	85.5	90.1	93.2

Table 2. Results for different data augmentation and view selection configurations. "PX" stands for pixel-only augmentations. "ULC" is ultra light cropping, "LC" is light cropping, and "Full" is the default pipeline with strong augmentation.

Method	Classif. Acc.	Re-ID mAP	3D mAP
Same - PX	44.2	0.02	0.29
Same - ULC + PX	83.1	0.24	0.59
Same - LC + PX	91.2	0.26	0.65
Same - Full	**96.8**	0.59	0.67
Nearby - PX	26.6	0.00	0.19
Nearby - ULC + PX	93.8	0.55	0.65
Nearby - LC + PX	95.7	0.53	**0.69**
Nearby - Full	94.5	0.71	0.66
Distant - PX	25.9	0.00	0.18
Distant - ULC + PX	93.8	0.22	0.58
Distant - LC + PX	91.4	0.82	0.60
Distant - Full	94.8	**0.85**	0.58

formly sampled from the video ("Distant"). Then, we use four different sets of data augmentation pipelines across all categories: pixel-only operations such as blur and color jitter ("PX"), ultra-light cropping (<20% rescale without aspect ratio changes) combined with pixel-only operations ("ULC+PX"), light cropping (<40% rescale without aspect ratio changes) combined with pixel-only operations ("LC+PX"), and strong augmentations ("Full"). Note that "Full" corresponds to the original setup proposed in SimCLR and used in the original version of SimSiam. We show the evaluation results in Table 2.

We can first observe in this table that all the pixel-only augmentation pipelines lead to bad representations (and thus bad downstream performance) due to the discovery of edge-level shortcuts during pre-training. This happens even with the "distant" strategy which suggests that the diversity of views itself is not sufficient to learn good representations. The use of random cropping at a large enough scale is important and seems to be a predictor of the overall performance. Besides, we can observe that generating views from frame pairs reduces the need for data augmentation to obtain similar performance levels.

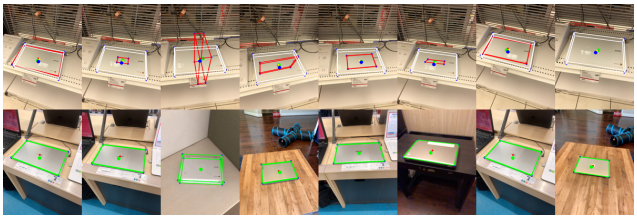

Figure 3. Examples of object thickness estimation failures combined with ground plane estimation failures. Query frames are shown in the top row, nearest neighbors in the bottom row.

Figure 4. Example of degraded book pose estimation. The orientation of the pattern on the book seems to confuse the pose estimation. Query frames are shown in the top row, nearest neighbors in the bottom row.

Figure 5. Example of failing nearest neighbor retrieval for a chair with a dot pattern (left). The other crops correspond to nearest neighbors across unique videos.

1.3. Failure Cases

Finally, we highlight cases where our pose estimation approach fails. In practice, about 69% of the 3D bounding boxes predicted using our zero-shot pipeline are within the 50% 3D IoU threshold of the ground-truth annotations of Objectron. The remainder are failure cases that fall into one of the two major categories described below.

Thickness and ground plane related failures. These failures happen when the predicted object orientation and size are correct but its thickness is incorrect. Such cases occur frequently with small books and closed laptops. They are sometimes also caused by bad ground plane estimations in the dataset that result in bounding box fitting errors. See Figure 3 for a few examples.

Optical illusions. These failures are related to the image content itself. For instance, in Figure 4, local features in the book's cover pattern suggest a 45° orientation. This leads to bad nearest neighbor matches (shown in the bottom row). In Figure 5, the grid pattern in the chair seems to dominate the embedding more than the pose information. In this case, the query frame is the first image on the left, and several nearest neighbors are shown consecutively after it.

References

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