Supplementary Material: Understanding Collapse in Non-Contrastive Siamese Representation Learning

Alexander C. Li[1], Alexei A. Efros[2], and Deepak Pathak[1]

[1]Carnegie Mellon University [2] University of California, Berkeley

1 ResNet-34 Results

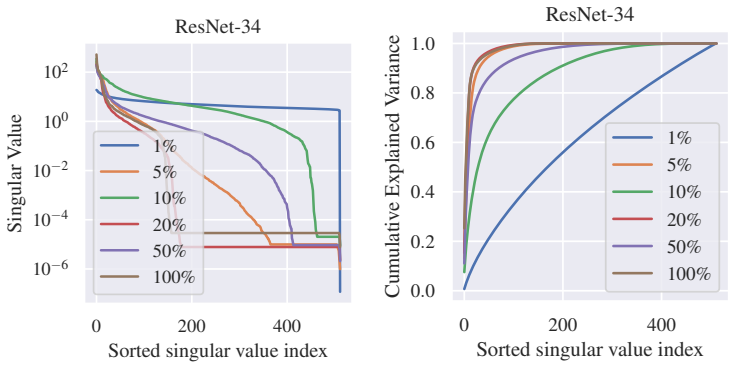

Fig. 1: **Partial dimensional collapse for large subsets.** Just as we did in Figure 3 for ResNet-18, we show the singular values of the representations computed by ResNet-34 models trained on different size subsets of ImageNet. The model trained on 1% exhibits no collapse, whereas more data (other than the 10% model) tends to lead to more collapse. Note that the degree of collapse for ResNet-34 is in general much worse than it is for ResNet-18. We hypothesize that this is due to its increased depth but equivalent width, which makes it easier for SimSiam to lose information at every layer and compute collapsed representations.

2 Distillation

Table 1: Distillation yields strong performance using smaller networks, regardless of pre-training algorithm. We show the linear probe accuracy of the ResNet-50 teachers trained by SimSiam or MoCo-v3 and the student networks obtained by distilling each teacher network.

Pre-training Algorithm	ResNet-50 Teacher	ResNet-18	ResNet-34
SimSiam	68.1	58.9	62.5
MoCo-v3	66.4	56.4	57.5

Model distillation [7,1] is a technique for compressing the knowledge in a large model into a smaller model. During the distillation process, the smaller student network learns to match the outputs of the larger teacher network. Training a large teacher model on a dataset (e.g. with cross-entropy loss) and distilling it into a smaller model typically performs better than directly training the small model. If we have a lot of compute available and only care about obtaining good small models, training a ResNet-50 with SimSiam (which does not collapse) and distilling it into a smaller model is an effective alternative approach for preventing partial dimensional collapse.

We distill a ResNet-50 into ResNet-18 and ResNet-34 by adding a fully connected layer that predicts the 2048-dimensional ResNet-50 representation and minimizes the mean-squared error (MSE). Note that this is equivalent to learning the top singular vectors of the teacher network representations, as the student network tries to learn a low-rank approximation of the teacher. Training the student is fairly straightforward. We train for 100 epochs on ImageNet-1k using the same hyperparameters used for SimSiam training. We find that the ResNet-50 teacher outputs are typically very small, on the order of 0.001 - 0.1, so minimizing the MSE with respect to the raw outputs leads to small gradient values and extremely long training times. Thus, we employ the standard trick of computing the mean and standard deviation of each dimension of the teacher output and using them to normalize the teacher output to have a mean of 0 and a variance of 1 in each dimension. We do this using an exponential moving average that updates the mean and standard deviation online.

As expected, Table 1 shows that distillation produces ResNet-18 and ResNet-34 networks with strong linear probing performance. Note that this outcome is orthogonal to our work. First, distillation is incredibly compute-heavy. Training and distilling ResNet-50 into ResNet-18 takes as much as 4× the compute as directly training the ResNet-18. Second, distillation is effective regardless of the pre-training algorithm – MoCo and SimSiam both benefit from distillation. Finally, distillation performance does not resolve the fact that vanilla SimSiam uniquely has the partial dimensional collapse problem and cannot be used to train smaller networks.

3 Additional Experiments

3.1 Vision Transformers

Self-supervised algorithms are typically evaluated using ResNets, but different architectures have qualitatively different behaviors. For example, self-supervised training with DINO [2] leads Vision Transformers [4] to learn features corresponding to semantic segmentation, whereas ResNets trained with DINO do not. Thus, we experimented with using SimSiam to train Vision Transformers of varying sizes, in order to look for further architecture-related qualitative differences. Due to computational constraints, we tried ViT-Small, which was used in [3], as well as variants with fewer layers or attention heads. We trained these models using SimSiam for 100 epochs using the following hyperparameters from [3]: AdamW optimizer, learning rate of 1.5×10^{-4}, weight decay of 0.1, learning rate warmup for 10 epochs, and frozen linear patch projection.

Surprisingly, these ViTs only achieve about 6-10% linear probe accuracy. There could be several reasons for their poor performance. We could have used bad hyperparameters, although this indicates that SimSiam is very sensitive to hyperparameter values. This also could be due to their limited representation size (384 dim), which makes it less likely that they have learned many useful features. Finally, this could indicate that ViT fundamentally lacks some architectural inductive bias that makes non-contrastive algorithms like SimSiam or BYOL work with ResNet. Further work in this area could be illuminating.

3.2 Nearest Neighbors SimSiam

We test whether a queue-based nearest neighbors loss (NNSiam, [5]) improves SimSiam training for ResNet-18. Given a pair of augmentations x_1 and x_2, the NNSiam objective is use x_1 to predict the nearest neighbor of x_2's projected representation in a MoCo-style queue. We train for 100 epochs on ImageNet with the same hyperparameters as vanilla SimSiam and a queue of length 25600.

This achieves a linear probe accuracy of 34.4% on ImageNet, which is better than the vanilla "multiple pass" baseline (30.0%), but still vastly underperforms our proposed methods, including "single pass" (44.5%) or hybrid training (48.3%).

3.3 Additional Baseline: Learning Rate Warmup

Table 2: Additional baseline for comparing validation top-1 linear finetuning accuracy for different SimSiam training methods.

Training method	ResNet-18	ResNet-34	ResNet-50
IID	30.01	16.83	**68.09**
IID + 10-epoch lr warmup	28.38	35.82	-

Figure 1(b) showed that MoCo-v3 [3] and BYOL [6] achieve reasonable performance with ResNet-18, whereas SimSiam collapses. One potential source of this difference is the learning rate warmup: MoCo-v3 and BYOL both utilize a linear learning rate warmup over the first 10 epochs, whereas SimSiam uses no warmup. We add a 10-epoch learning rate warmup to SimSiam and show that this detail is not responsible for the huge deficit in SimSiam performance. Table 2 shows that warmup decreases ResNet-18 performance from 30.01% to 28.38% but increases ResNet-34 performance from 16.83% to 35.82%. This still falls quite short of the performance of our hybrid continual-IID method, which outperforms this baseline by 20 percentage points (ResNet-18) and 15 percentage points (ResNet-34).

References

1. Buciluǎ, C., Caruana, R., Niculescu-Mizil, A.: Model compression. In: SIGKDD (2006) 2
2. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9650–9660 (2021) 3
3. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9640–9649 (2021) 3, 4
4. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. preprint arXiv:2010.11929 (2020) 3
5. Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9588–9597 (2021) 3
6. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M.: Bootstrap your own latent: A new approach to self-supervised learning. In: NeurIPS (2020) 4
7. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. preprint arXiv:1503.02531 (2015) 2