SimSiam: Exploring Simple Siamese Representation Learning

General idea behind this paper is to prove that all contrastive learning method are result of siamese network in one way or other and all other techniques used in MoCo, BYOL, SimCLR or SwAV are just design choices.

SimSam just uses stop gradient in order to train a good enough contrastive model with far less batch size.

Focus of paper

This paper focuses on employing simple Siamese networks to learn meaningful representation even in the absence of

1) negative sample pairs (SimCLR)
2) large batches
3) momentum encoders

They show collapsing solutions do exist for the loss and structure, but a stop-gradient operation plays an essential role in preventing collapsing.

Dissimilarities with other CL methods

SimSam can be thought of as

1) BYOL without the momentum encoder
2) SimCLR without negative pairs
3) SwAV without online clustering

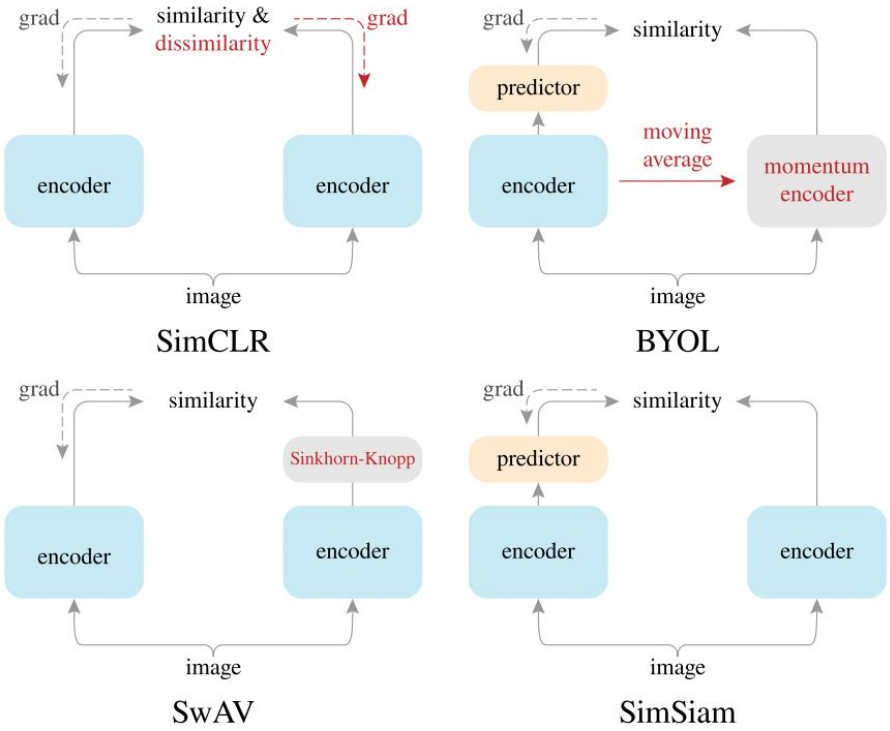


Figure 3. **Comparison on Siamese architectures**. The encoder includes all layers that can be shared between both branches. The dash lines indicate the gradient propagation flow. In BYOL, SwAV, and SimSiam, the lack of a dash line implies stop-gradient, and their symmetrization is not illustrated for simplicity. The components in red are those missing in SimSiam.

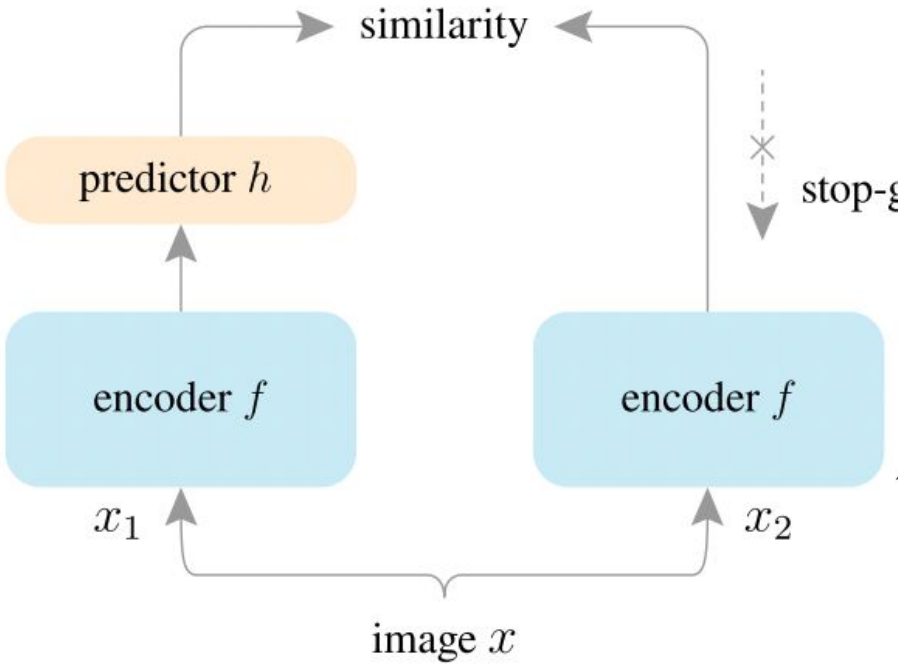
Finding

stop-gradient operation is critical.

This finding can be obscured with the usage of a momentum encoder, which is always accompanied with stop-gradient (as it is not updated by its parameters' gradients).

While the moving-average behavior may improve accuracy with an appropriate momentum coefficient, our experiments show that it is not directly related to preventing collapsing.

Architecture



$$p_1 \triangleq h(f(x_1)) \text{ and } z_2 \triangleq f(x_2),$$

$$\mathcal{D}(p_1, z_2) = -\frac{p_1}{\|p_1\|_2} \cdot \frac{z_2}{\|z_2\|_2},$$

$$\mathcal{L} = \frac{1}{2}\mathcal{D}(p_1, \texttt{stopgrad}(z_2)) + \frac{1}{2}\mathcal{D}(p_2, \texttt{stopgrad}(z_1))$$

$$\mathcal{L} = \frac{1}{2}\mathcal{D}(p_1, \texttt{stopgrad}(z_2)) + \frac{1}{2}\mathcal{D}(p_2, \texttt{stopgrad}(z_1))$$

The encoder on x_2 receives no gradient from z_2 in the first term, but it receives gradients from p_2 in the second term (and vice versa for x_1).

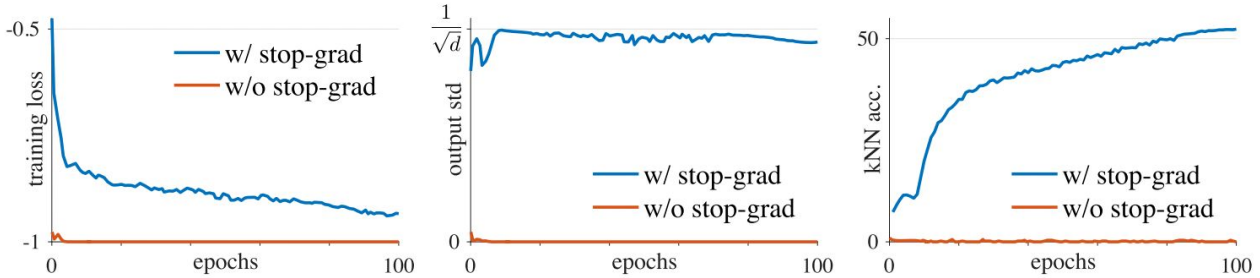
x_1 is firstly fed to trainable encoder and then x_2 is fed to it in one training step.

They also show doing the training way boosts the accuracy. They also trying using asymmetric loss by sampling two pairs for each image in the asymmetric version ("2×"). It makes the gap smaller.

	sym.	asym.	asym. 2×
acc. (%)	68.1	64.8	67.3

Training with and without stop gradient

Without stop-gradient, the optimizer quickly finds a degenerated solution and reaches the minimum possible loss of−1.



	acc. (%)
w/ stop-grad	67.7±0.1
w/o stop-grad	0.1

Figure 2. **SimSiam with *vs*. without stop-gradient**. **Left plot**: training loss. Without stop-gradient it degenerates immediately. **Middle plot**: the per-channel std of the ℓ_2-normalized output, plotted as the averaged std over all channels. **Right plot**: validation accuracy of a kNN classifier [34] as a monitor of progress. **Table**: ImageNet linear evaluation ("w/ stop-grad" is mean±std over 5 trials).

Clustering way of looking at SimSam

F is a network parameterized by θ. T is the augmentation. x is an image. The expectation E[·] is over the distribution of images and augmentations.

$$\mathcal{L}(\theta, \eta) = \mathbb{E}_{x,\mathcal{T}} \left[\left\| \mathcal{F}_\theta(\mathcal{T}(x)) - \eta_x \right\|_2^2 \right].$$

ηx is the representation of the image x, η is not necessarily the output of a network; it is the argument of an optimization problem

$$\min_{\theta,\eta} \mathcal{L}(\theta, \eta).$$

The variable θ is analogous to the clustering centers: it is the learnable parameters of an encoder. The variable ηx is analogous to the assignment vector of the sample x (a one-hot vector in k- means): it is the representation of x.

they alternate between these sub-problems:

$$\begin{aligned} \theta^t &\leftarrow \arg\min_{\theta} \mathcal{L}(\theta, \eta^{t-1}) \\ \eta^t &\leftarrow \arg\min_{\eta} \mathcal{L}(\theta^t, \eta) \end{aligned}$$

Solving for θ. use SGD to solve the sub-problem, η_{t-1} which is a constant in this subproblem.

Solving for η. The sub-problem can be solved independently for each ηx.

$$\mathbb{E}_{\mathcal{T}}\left[\|\mathcal{F}_{\theta^t}(\mathcal{T}(x)) - \eta_x\|_2^2\right]$$

$$\eta_x^t \leftarrow \mathbb{E}_{\mathcal{T}}\left[\mathcal{F}_{\theta^t}(\mathcal{T}(x))\right].$$

Alteration:

$$\theta^{t+1} \leftarrow \arg\min_{\theta} \mathbb{E}_{x,\mathcal{T}}\left[\left\|\mathcal{F}_{\theta}(\mathcal{T}(x)) - \mathcal{F}_{\theta^t}(\mathcal{T}'(x))\right\|_2^2\right]$$

	1-step	10-step	100-step	1-epoch
acc. (%)	68.1	68.7	68.9	67.0

Results

method	batch size	negative pairs	momentum encoder	100 ep	200 ep	400 ep	800 ep
SimCLR (repro.+)	4096	✓		66.5	68.3	69.8	70.4
MoCo v2 (repro.+)	**256**	✓	✓	67.4	69.9	71.0	72.2
BYOL (repro.)	4096		✓	66.5	**70.6**	**73.2**	**74.3**
SwAV (repro.+)	4096			66.5	69.1	70.7	71.8
SimSiam	**256**			**68.1**	70.0	70.8	71.3

Table 4. **Comparisons on ImageNet linear classification**. All are based on **ResNet-50** pre-trained with **two 224×224 views**. Evaluation is on a single crop. All competitors are from our reproduction, and "+" denotes *improved* reproduction *vs*. original papers (see supplement).

	VOC 07 detection			VOC 07+12 detection			COCO detection			COCO instance seg.		
pre-train	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}	AP^{mask}_{50}	AP^{mask}	AP^{mask}_{75}
scratch	35.9	16.8	13.0	60.2	33.8	33.1	44.0	26.4	27.8	46.9	29.3	30.8
ImageNet supervised	74.4	42.4	42.7	81.3	53.5	58.8	58.2	38.2	41.2	54.7	33.3	35.2
SimCLR (repro.+)	75.9	46.8	50.1	81.8	55.5	61.4	57.7	37.9	40.9	54.6	33.3	35.3
MoCo v2 (repro.+)	**77.1**	**48.5**	**52.5**	**82.3**	**57.0**	**63.3**	**58.8**	**39.2**	**42.5**	**55.5**	**34.3**	**36.6**
BYOL (repro.)	**77.1**	47.0	49.9	81.4	55.3	61.1	57.8	37.9	40.9	54.3	33.2	35.0
SwAV (repro.+)	75.5	46.5	49.6	81.5	55.4	61.4	57.6	37.6	40.3	54.2	33.1	35.1
SimSiam, base	75.5	47.0	50.2	**82.0**	56.4	62.8	57.5	37.9	40.9	54.2	33.2	35.2
SimSiam, optimal	**77.3**	**48.5**	**52.5**	**82.4**	**57.0**	**63.7**	**59.3**	**39.2**	**42.1**	**56.0**	**34.4**	**36.7**

Table 5. **Transfer Learning**. All unsupervised methods are based on 200-epoch pre-training in ImageNet. *VOC 07 detection*: Faster R-CNN [30] fine-tuned in VOC 2007 trainval, evaluated in VOC 2007 test; *VOC 07+12 detection*: Faster R-CNN fine-tuned in VOC 2007 trainval + 2012 train, evaluated in VOC 2007 test; *COCO detection* and *COCO instance segmentation*: Mask R-CNN [18] (1× schedule) fine-tuned in COCO 2017 train, evaluated in COCO 2017 val. All Faster/Mask R-CNN models are with the C4-backbone [13]. All VOC results are the average over 5 trials. **Bold entries** are within 0.5 below the best.