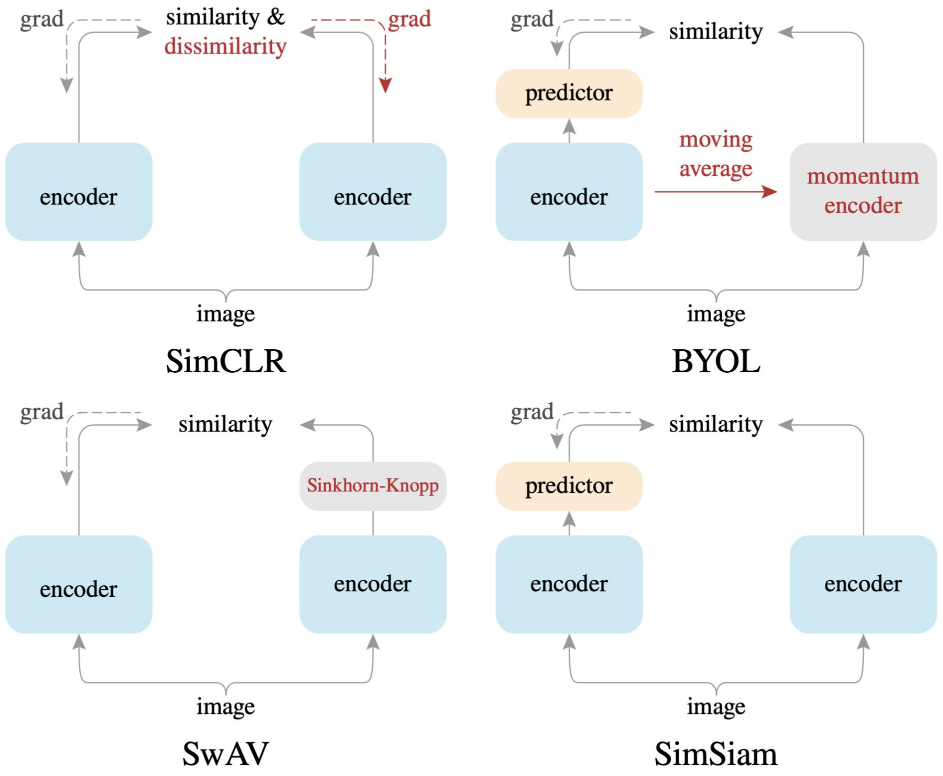
Current state of self-supervised learning

A presentation concerning:

- BYOL
- SimSiam


Let me have a try.
OPEN SESAME!
Well, I've done all I can do.

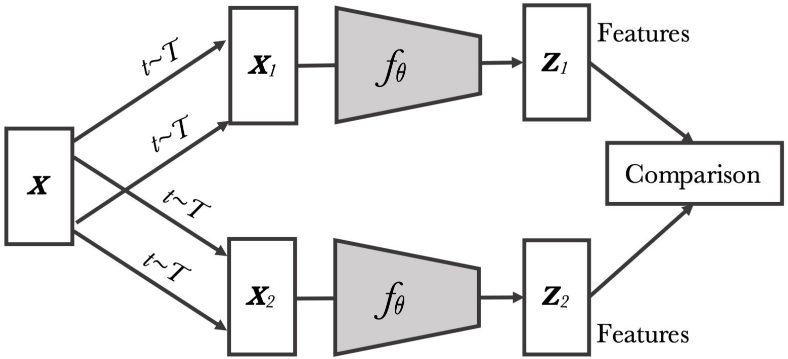

grad
similarity &
dissimilarity
grad
encoder
encoder
image
SimCLR
grad
similarity
predictor
moving
average
encoder
momentum
encoder
image
BYOL
grad
similarity
Sinkhorn-Knopp
encoder
encoder
image
SwAV
grad
similarity
predictor
encoder
encoder
image
SimSiam

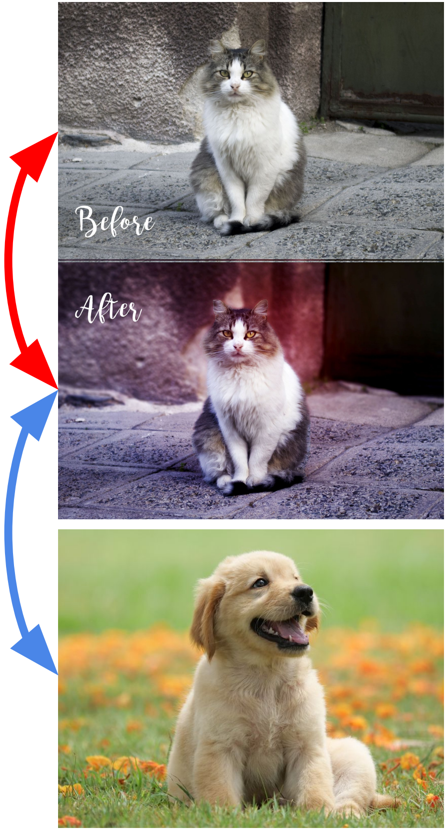
SimCLR

- Contrastive learning
- https://arxiv.org/abs/2002.05709

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

$$\frac{\textbf{Attract}}{\textbf{Repulse}}$$





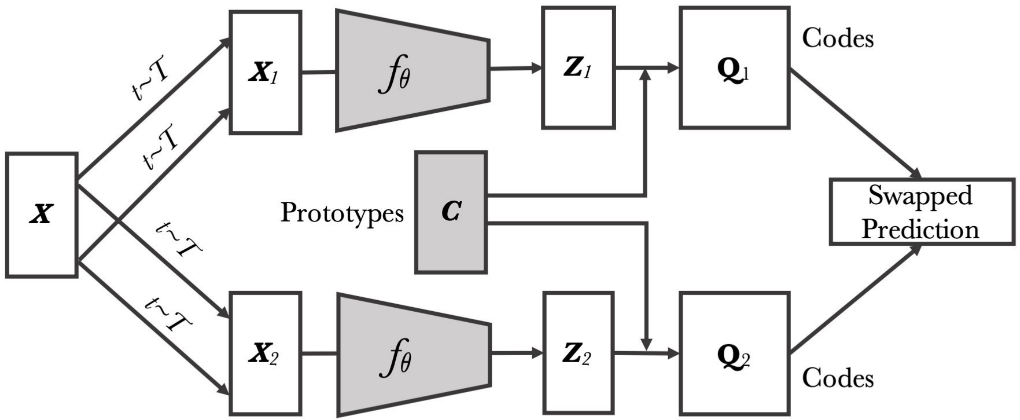
SwAV

- Contrastive Learning with online clustering
- https://arxiv.org/abs/2006.09882

$$L(\mathbf{z}_t, \mathbf{z}_s) = \ell(\mathbf{z}_t, \mathbf{q}_s) + \ell(\mathbf{z}_s, \mathbf{q}_t)$$

$$\ell(\mathbf{z}_t, \mathbf{q}_s) = -\sum_k \mathbf{q}_s^{(k)} \log \mathbf{p}_t^{(k)}$$

$$\mathbf{p}_t^{(k)} = \frac{\exp\left(\frac{1}{\tau}\mathbf{z}_t^\top \mathbf{c}_k\right)}{\sum_{k'} \exp\left(\frac{1}{\tau}\mathbf{z}_t^\top \mathbf{c}_{k'}\right)}$$



$$\max_{\mathbf{Q} \in \mathcal{Q}} \operatorname{Tr}\left(\mathbf{Q}^\top \mathbf{C}^\top \mathbf{Z}\right) + \varepsilon H(\mathbf{Q})$$

$$\mathcal{Q} = \left\{ \mathbf{Q} \in \mathbb{R}_+^{K \times B} \mid \mathbf{Q}\mathbf{1}_B = \frac{1}{K}\mathbf{1}_K, \mathbf{Q}^\top \mathbf{1}_K = \frac{1}{B}\mathbf{1}_B \right\}$$

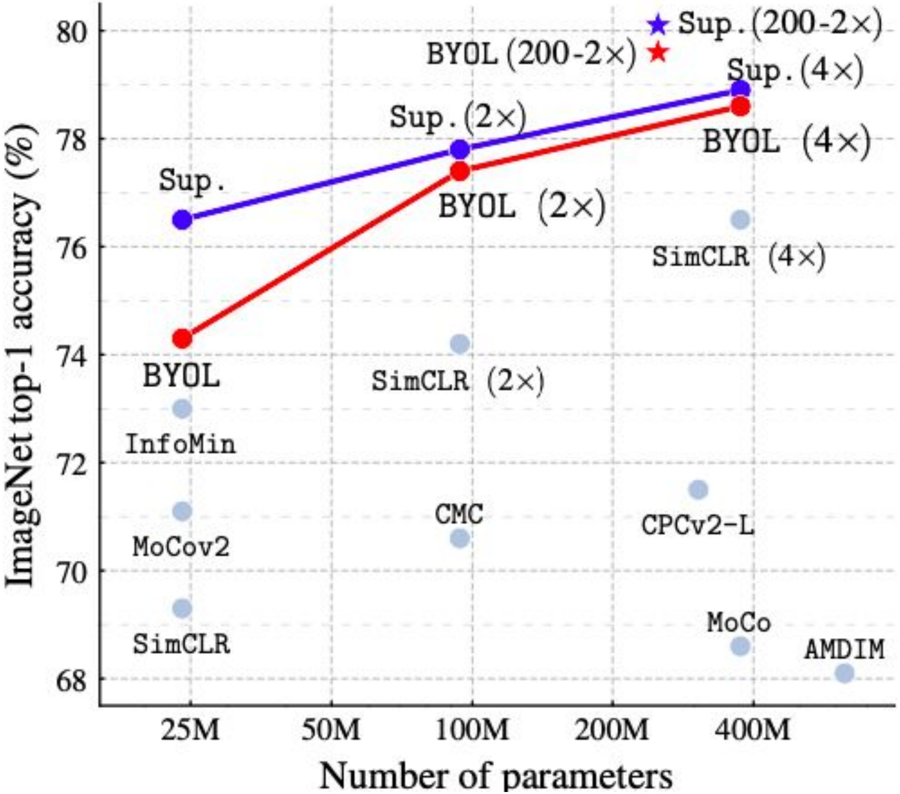
$$\mathbf{Q}^* = \operatorname{Diag}(\mathbf{u}) \exp\left(\frac{\mathbf{C}^\top \mathbf{Z}}{\varepsilon}\right) \operatorname{Diag}(\mathbf{v})$$

$$-\frac{1}{N} \sum_{n=1}^{N} \sum_{s,t \sim \mathcal{T}} \left[\frac{1}{\tau} \mathbf{z}_{nt}^\top \mathbf{C} \mathbf{q}_{ns} + \frac{1}{\tau} \mathbf{z}_{ns}^\top \mathbf{C} \mathbf{q}_{nt} - \log \sum_{k=1}^{K} \exp\left(\frac{\mathbf{z}_{nt}^\top \mathbf{c}_k}{\tau}\right) - \log \sum_{k=1}^{K} \exp\left(\frac{\mathbf{z}_{ns}^\top \mathbf{c}_k}{\tau}\right) \right]$$

Bootstrap your own latent

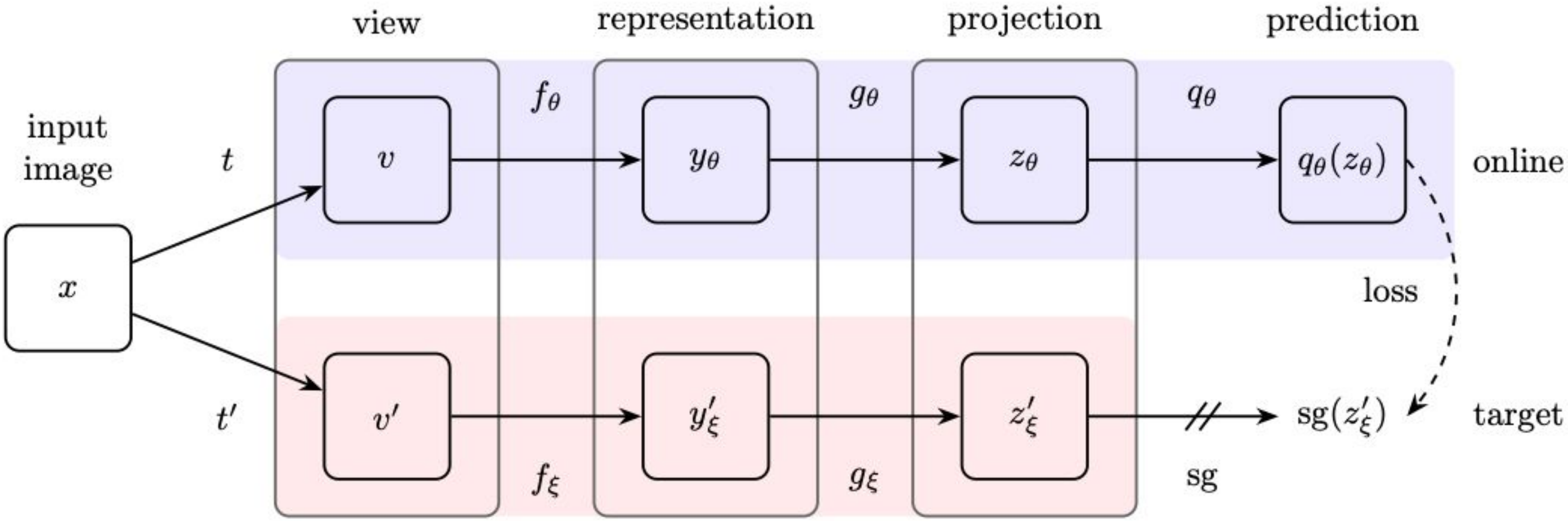
Grill, Strub, Altche, Tallec, Richemond
DeepMind
https://arxiv.org/abs/2006.07733

BYOL Performance



How does BYOL work

Train on similarity between network and its moving average



How does BYOL work

Moving-average network updates is used to keep the target network as a stable version of the online network

We symmetrize the loss $\mathcal{L}_{\theta,\xi}$ in Eq. 2 by separately feeding v' to the online network and v to the target network to compute $\widetilde{\mathcal{L}}_{\theta,\xi}$. At each training step, we perform a stochastic optimization step to minimize $\mathcal{L}^{\text{BYOL}}_{\theta,\xi} = \mathcal{L}_{\theta,\xi} + \widetilde{\mathcal{L}}_{\theta,\xi}$ with respect to θ only, but *not* ξ, as depicted by the stop-gradient in Figure 2. BYOL's dynamics are summarized as

$$\theta \leftarrow \text{optimizer}\left(\theta, \nabla_\theta \mathcal{L}^{\text{BYOL}}_{\theta,\xi}, \eta\right), \tag{3}$$

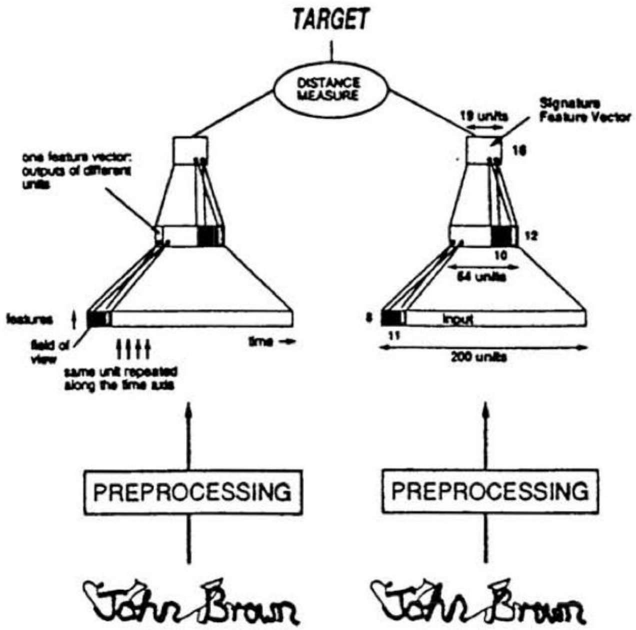
$$\xi \leftarrow \tau\xi + (1-\tau)\theta, \tag{1}$$

where optimizer is an optimizer and η is a learning rate.

Xi are the parameters of the target network

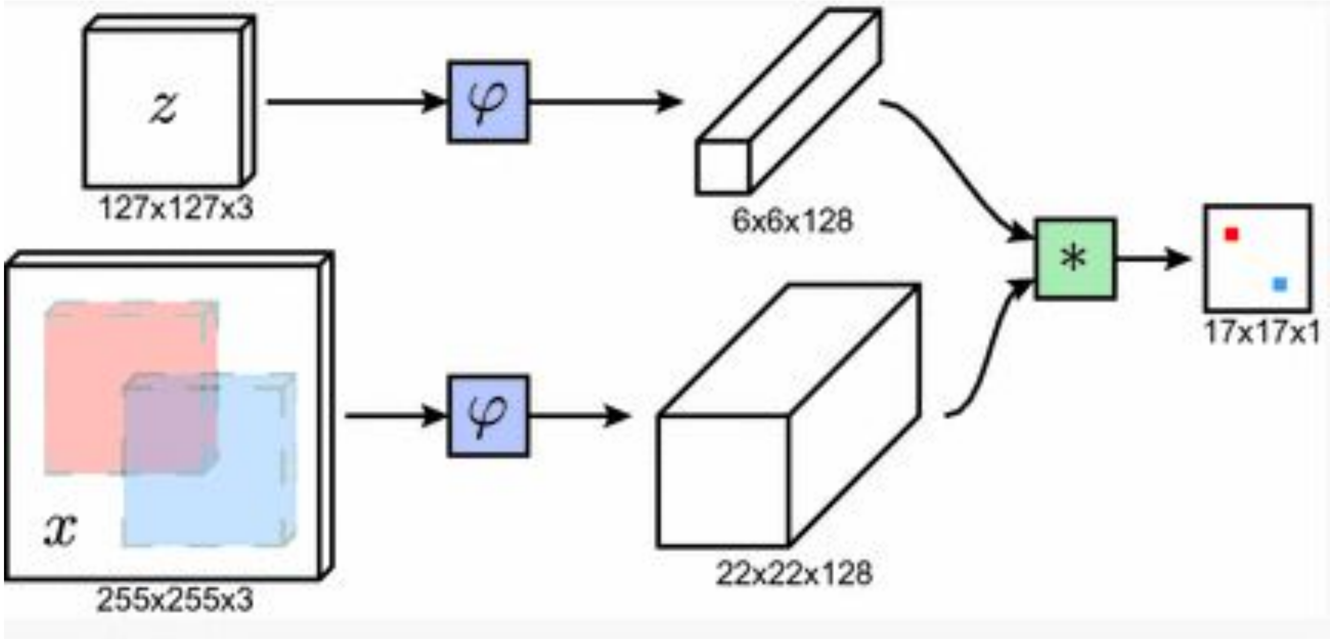
What have siamese networks been used for

The original siamese network was used for signature verification



What have siamese networks been used for

Recently a large body of literature uses siamese networks for object tracking



Exploring Simple Siamese Representation Learning

Xinlei Chen Kaiming He
Facebook AI Research (FAIR)
https://arxiv.org/abs/2011.10566

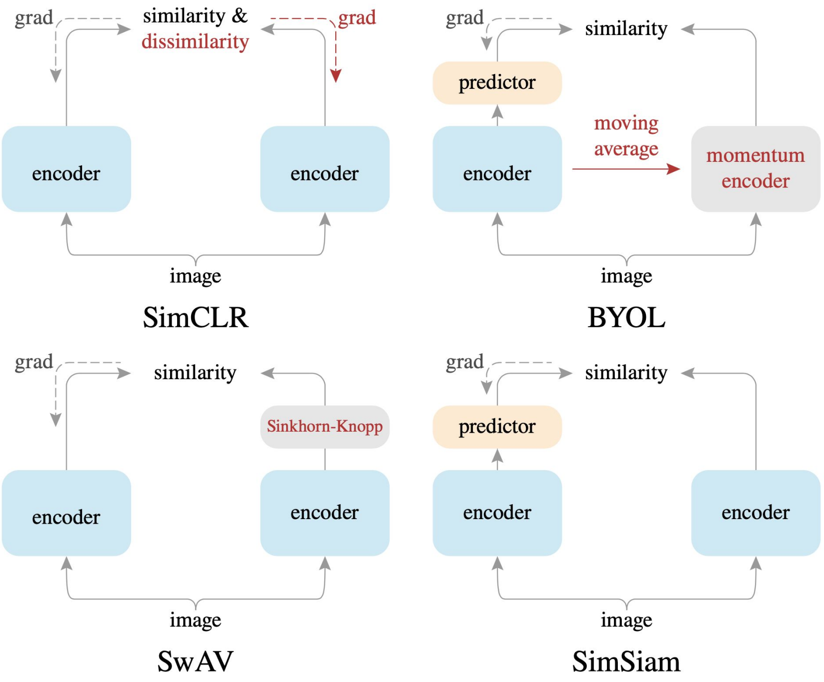
SimSiam

- “SimCLR without negatives”
- “SwAV without online clustering”
- “BYOL without the momentum encoder”

$$p_1 \triangleq h(f(x_1)) \qquad z_2 \triangleq f(x_2)$$

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

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



Comparisons on ImageNet linear classification

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

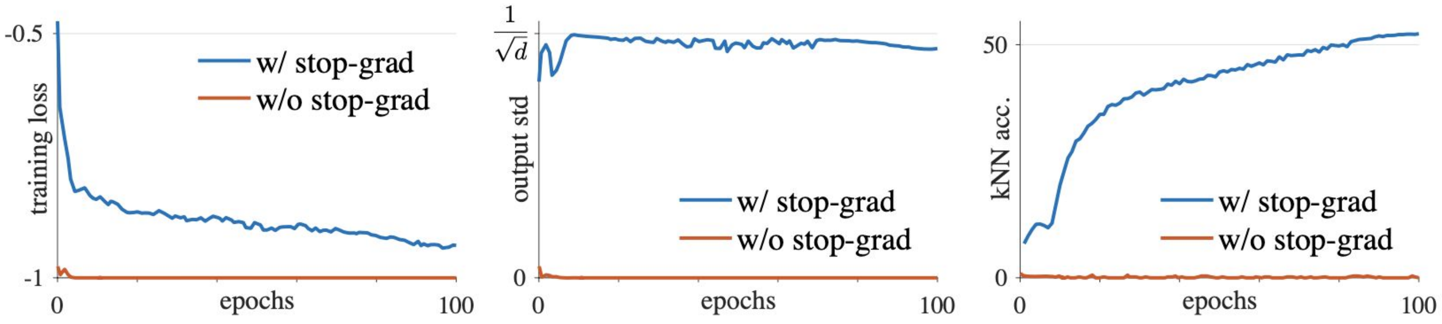
Table 4. **Comparisons on ImageNet linear classification**. All are based on **ResNet-50** pre-trained with **two 224×224 views**. Evaluation is on a single crop. All competitors are from our reproduction, and "+" denotes *improved* reproduction *vs*. original papers (see supplement).

Transfer Learning

pre-train	VOC 07 detection			VOC 07+12 detection			COCO detection			COCO instance seg.		
	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}	AP_{50}^{mask}	AP^{mask}	AP_{75}^{mask}
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**

Analysis of building blocks

Stop-gradient



	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 [36] as a monitor of progress. **Table**: ImageNet linear evaluation ("w/ stop-grad" is mean±std over 5 trials).

$$\mathcal{D}(p_1, \texttt{stopgrad}(z_2))$$

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

Predictor h

- The model does not work if removing h
- Stopgrad model is equivalent to $\mathcal{D}(z_1, z_2)$

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

	pred. MLP h	acc. (%)
baseline	lr with cosine decay	67.7
(a)	no pred. MLP	0.1
(b)	fixed random init.	1.5
(c)	lr not decayed	68.1

Table 1. **Effect of prediction MLP** (ImageNet linear evaluation accuracy with 100-epoch pre-training). In all these variants, we use the same schedule for the encoder f (lr with cosine decay).

Batch size

batch size	64	128	256	512	1024	2048	4096
acc. (%)	66.1	67.3	68.1	68.1	68.0	67.9	64.0

Table 2. **Effect of batch sizes** (ImageNet linear evaluation accuracy with 100-epoch pre-training).

Batch Normalization

	case	proj. MLP's BN hidden	proj. MLP's BN output	pred. MLP's BN hidden	pred. MLP's BN output	acc. (%)
(a)	none	-	-	-	-	34.6
(b)	hidden-only	✓	-	✓	-	67.4
(c)	default	✓	✓	✓	-	68.1
(d)	all	✓	✓	✓	✓	unstable

Table 3. **Effect of batch normalization on MLP heads** (ImageNet linear evaluation accuracy with 100-epoch pre-training).

Similarity Function

$$\mathcal{D}(p_1, z_2) = -\texttt{softmax}(z_2) \cdot \log \texttt{softmax}(p_1)$$

	cosine	cross-entropy
acc. (%)	68.1	63.2

Symmetrization

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

Hypothesis on Why SimSiam works

Expectation-Maximization

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

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

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

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

One-step alternation

$$\eta_x^t \leftarrow \mathcal{F}_{\theta^t}(\mathcal{T}'(x))$$

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

Multi-Step Alternation and Moving-Average (memory bank)

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

$$\eta_x^t \leftarrow m * \eta_x^{t-1} + (1-m) * \mathcal{F}_{\theta^t}(\mathcal{T}'(x))$$

Predictor

$$\mathbb{E}_z\left[\left\|h(z_1) - z_2\right\|_2^2\right]$$

$$h(z_1) = \mathbb{E}_z[z_2] = \mathbb{E}_{\mathcal{T}}\left[f(\mathcal{T}(x))\right]$$

Thank you for attention!