LSDL Lecture 04
Contrastive Learning

Ildus Sadrtdinov, 07.10.24

Self-supervised pre-training

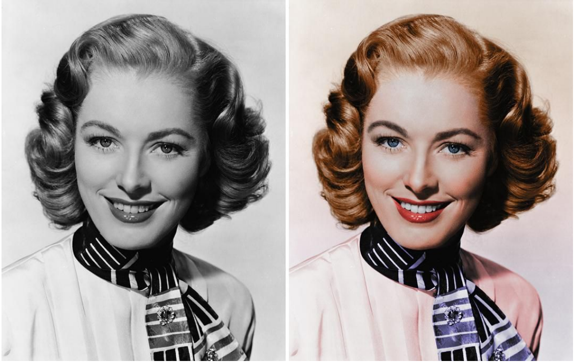
generative

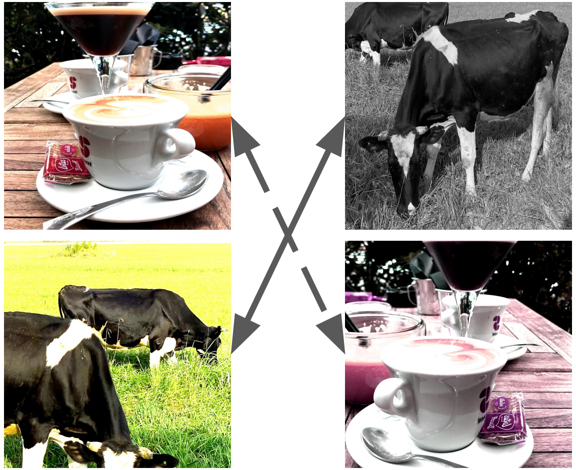
discriminative

generative pre-text tasks

discriminative pre-text tasks

contrastive tasks





Plan

- **Introduction to information theory**
  - Mutual information & InfoNCE
- **Contrastive learning with negative examples**
  - SimCLR, MoCo
- **Contrastive learning without negative examples**
  - BYOL, SimSiam
- **Contrastive learning inspired by clustering**
  - DeepCluster, SwAV
- **Bonus**
  - Dense Contrastive Learning
  - Supervised Contrastive Learning

Plan

- **Introduction to information theory**
  - Mutual information & InfoNCE
- **Contrastive learning with negative examples**
  - SimCLR, MoCo
- **Contrastive learning without negative examples**
  - BYOL, SimSiam
- **Contrastive learning inspired by clustering**
  - DeepCluster, SwAV
- **Bonus**
  - Dense Contrastive Learning
  - Supervised Contrastive Learning

Information theory

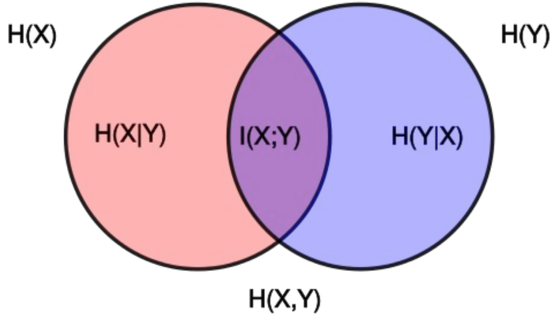
Is it possible to quantify information stored in a random variable?

- (Differential) Entropy: $H(X) = -\mathbb{E}_{p(X)}\Big[\log p(X)\Big] = -\int p(x)\log p(x)dx$
- Joint Entropy: $H(X,Y) = -\mathbb{E}_{p(X,Y)}\Big[\log p(X,Y)\Big] = -\int p(x,y)\log p(x,y)dxdy$
- Conditional Entropy: $H(Y|X) = -\mathbb{E}_{p(X,Y)}\Big[\log p(Y|X)\Big] = -\int p(x,y)\log \frac{p(x,y)}{p(x)}dxdy$

Information is measured in bits (if ***log = log***$_2$) or in nats (if ***log = ln***)

Wikipedia

Mutual information



$$I(X;Y) = KL\Big(P(X,Y)\Big|\Big|P(X)P(Y)\Big) = \int p(x,y)\log\frac{p(x,y)}{p(x)p(y)}dxdy$$

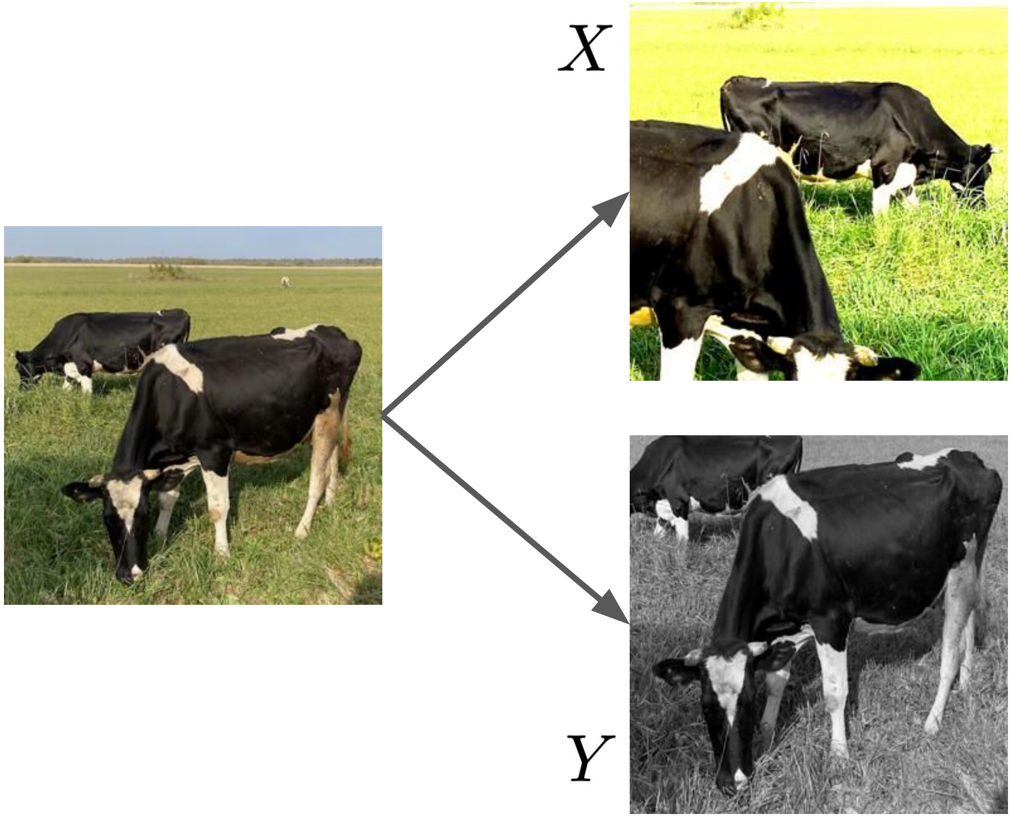
Properties:

- $\mathrm{I}(X;Y) \geq 0$
- $\mathrm{I}(X;Y) = \mathrm{I}(Y;X)$
- $$\begin{aligned}\mathrm{I}(X;Y) &\equiv \mathrm{H}(X) - \mathrm{H}(X \mid Y) \\ &\equiv \mathrm{H}(Y) - \mathrm{H}(Y \mid X) \\ &\equiv \mathrm{H}(X) + \mathrm{H}(Y) - \mathrm{H}(X,Y) \\ &\equiv \mathrm{H}(X,Y) - \mathrm{H}(X \mid Y) - \mathrm{H}(Y \mid X)\end{aligned}$$

- When $I(X;Y) = 0$?
- When $I(X;Y) \to \max$?

Wikipedia

So what? (or the main idea of contrastive learning)



$$I\Big(f_\theta(X), f_\theta(Y)\Big) \to \max_\theta$$

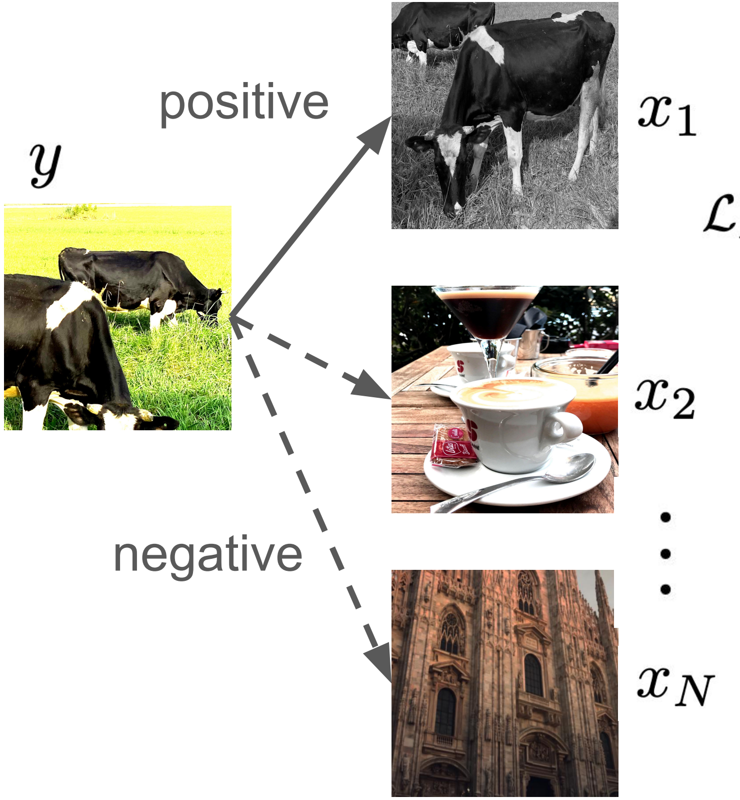
f_θ – our neural network with weights θ

Let’s go training! Oh, wait…

$$I(X,Y) = \int p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) dxdy = \mathbb{E}_{p(x,y)}\left[\log\frac{p(x,y)}{p(x)p(y)}\right]$$

We do not know any of these PDFs! :(

InfoNCE loss and negative examples



$$\mathcal{L}_{NCE}(\theta) = \mathbb{E}_{p(x_{1:N}, y)} \left[-\log \frac{e^{f_\theta(x_1, y)}}{\sum_{n=1}^{N} e^{f_\theta(x_n, y)}} \right] \to \min_\theta$$

Noise **C**ontrastive **E**stimation

Oord et al., 2018

InfoNCE is a lower bound for mutual information

$$I(X_1; Y) \geq \log N - \mathcal{L}_{NCE}$$

$$\mathcal{L}_{NCE}(\theta) = \mathbb{E}_{p(x_{1:N}, y)} \left[-\log \frac{e^{f_\theta(x_1, y)}}{\sum_{n=1}^{N} e^{f_\theta(x_n, y)}} \right]$$

Oord et al., 2018 Poole et al., 2019

Plan

- **Introduction to information theory**
  - Mutual information & InfoNCE
- **Contrastive learning with negative examples**
  - SimCLR, MoCo
- **Contrastive learning without negative examples**
  - BYOL, SimSiam
- **Contrastive learning inspired by clustering**
  - DeepCluster, SwAV
- **Bonus**
  - Dense Contrastive Learning
  - Supervised Contrastive Learning

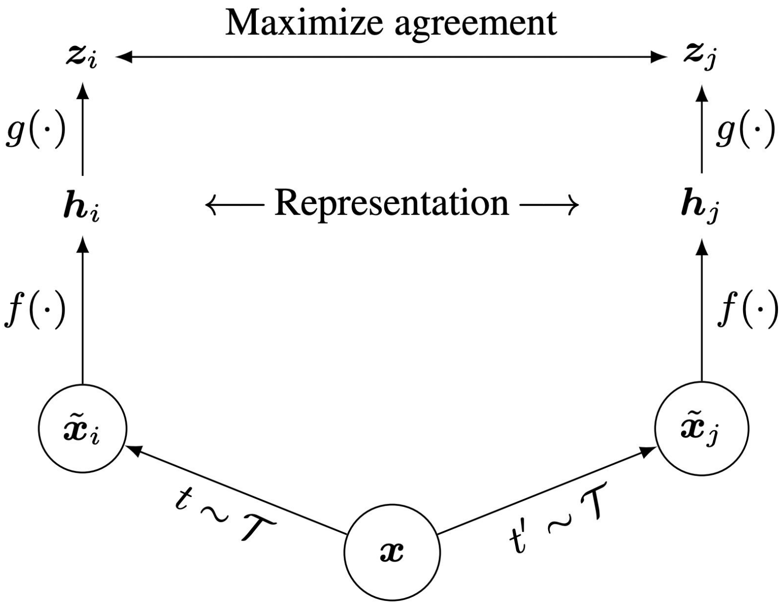
SimCLR

A **Sim**ple Framework for **C**ontrastive **L**earning of Visual **R**epresentations
Google Research

- Make a double batch of augmented images (each image has 2 view)
- Use each of two views both as an anchor ($\boldsymbol{y}$) and a positive example ($\boldsymbol{x_1}$)
- Very intense augmentations to make the task meaningful

Chen et al., 2020

SimCLR: scheme

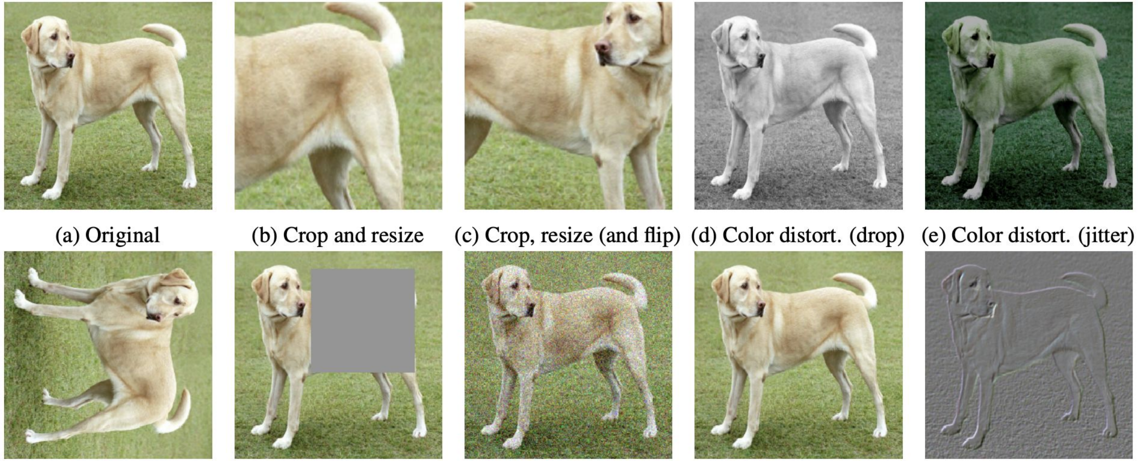


$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

$$\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$$

Chen et al., 2020

SimCLR: augmentations



(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)
(f) Rotate $\{90°, 180°, 270°\}$ (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

Figure 4. Illustrations of the studied data augmentation operators. Each augmentation can transform data stochastically with some internal parameters (e.g. rotation degree, noise level). Note that we *only* test these operators in ablation, the *augmentation policy used to train our models* only includes *random crop (with flip and resize)*, *color distortion*, and *Gaussian blur*. (Original image cc-by: Von.grzanka)

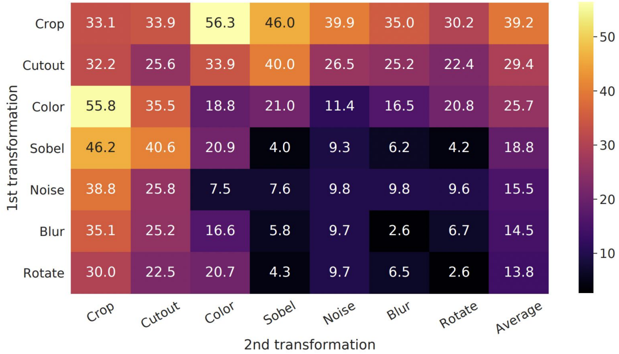
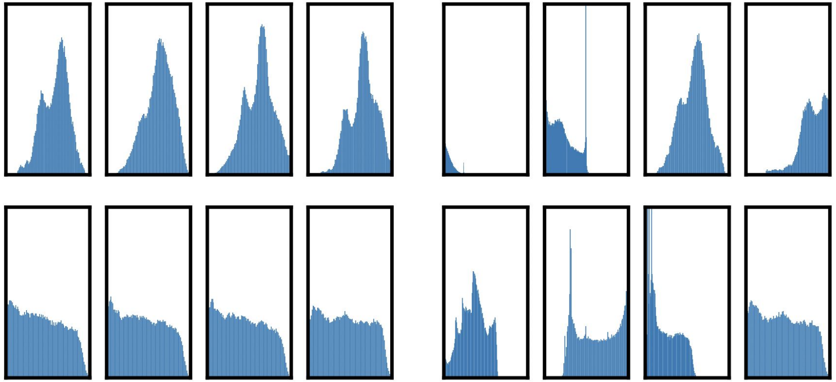


Figure 5. Linear evaluation (ImageNet top-1 accuracy) under individual or composition of data augmentations, applied only to one branch. For all columns but the last, diagonal entries correspond to single transformation, and off-diagonals correspond to composition of two transformations (applied sequentially). The last column reflects the average over the row.

Chen et al., 2020

SimCLR: augmentations



(a) Without color distortion. (b) With color distortion.

Figure 6. Histograms of pixel intensities (over all channels) for different crops of two different images (i.e. two rows). The image for the first row is from Figure 4. All axes have the same range.

Methods	Color distortion strength 1/8	1/4	1/2	1	1 (+Blur)	AutoAug
SimCLR	59.6	61.0	62.6	63.2	64.5	61.1
Supervised	77.0	76.7	76.5	75.7	75.4	77.1

Table 1. Top-1 accuracy of unsupervised ResNet-50 using linear evaluation and supervised ResNet-50[5], under varied color distortion strength (see Appendix A) and other data transformations. Strength 1 (+Blur) is our default data augmentation policy.

Chen et al., 2020

SimCLR: projection head

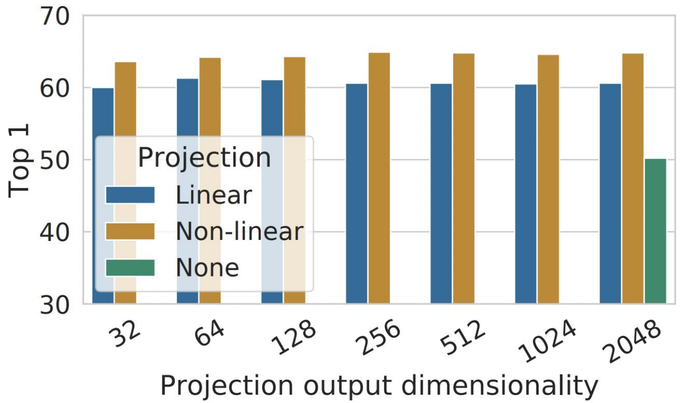


Figure 8. Linear evaluation of representations with different projection heads $g(\cdot)$ and various dimensions of $\boldsymbol{z} = g(\boldsymbol{h})$. The representation $\boldsymbol{h}$ (before projection) is 2048-dimensional here.

What to predict?	Random guess	Representation $\boldsymbol{h}$	Representation $g(\boldsymbol{h})$
Color vs grayscale	80	99.3	97.4
Rotation	25	67.6	25.6
Orig. vs corrupted	50	99.5	59.6
Orig. vs Sobel filtered	50	96.6	56.3

Table 3. Accuracy of training additional MLPs on different representations to predict the transformation applied. Other than crop and color augmentation, we additionally and independently add rotation (one of $\{0°, 90°, 180°, 270°\}$), Gaussian noise, and Sobel filtering transformation during the pretraining for the last three rows. Both $\boldsymbol{h}$ and $g(\boldsymbol{h})$ are of the same dimensionality, i.e. 2048.

Chen et al., 2020

SimCLR: model size and training time

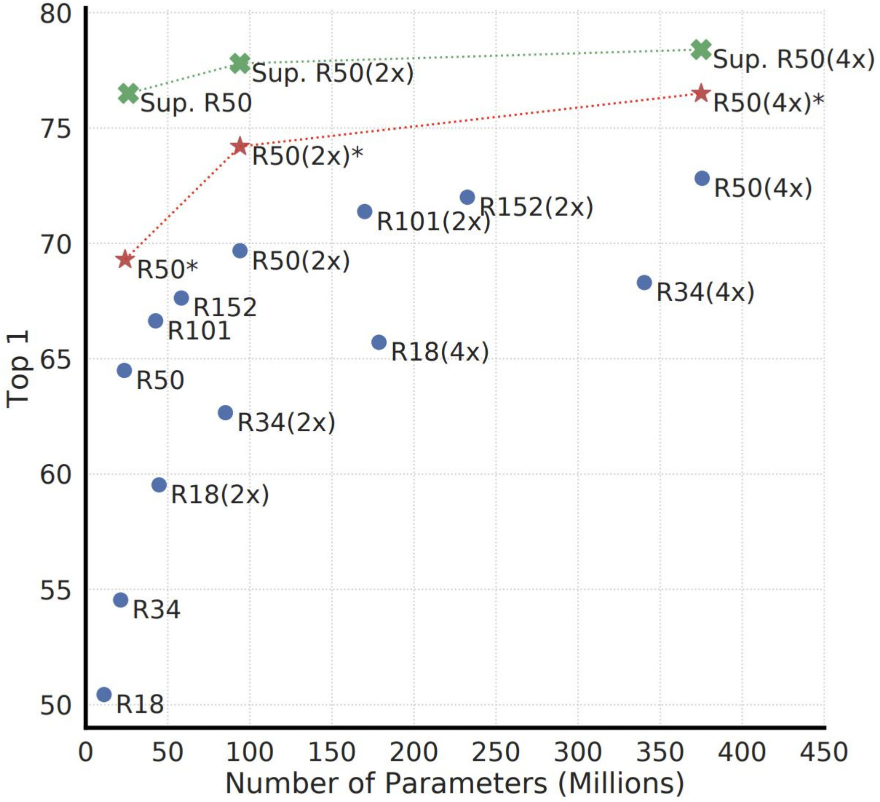


Figure 7. Linear evaluation of models with varied depth and width. Models in blue dots are ours trained for 100 epochs, models in red stars are ours trained for 1000 epochs, and models in green crosses are supervised ResNets trained for 90 epochs[7] (He et al., 2016).

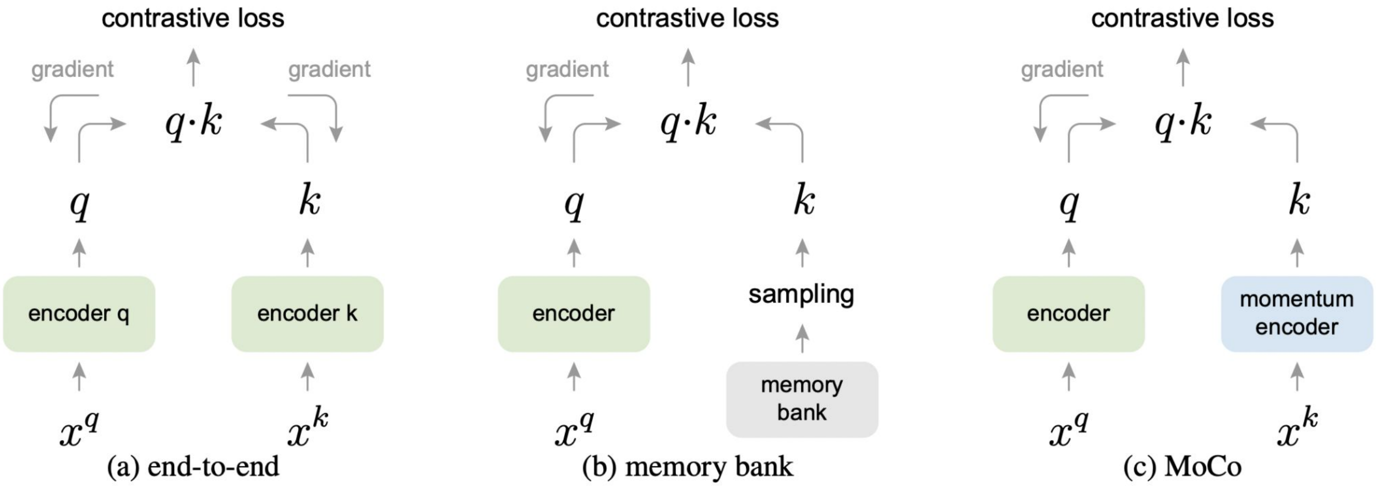
Chen et al., 2020

MoCo

Momentum **Co**ntrast, FAIR

- Momentum encoder makes contrastive learning asymmetric
- “Memory bank” to increase the number of negatives

$$\theta_{\mathrm{k}} \leftarrow m\theta_{\mathrm{k}} + (1-m)\theta_{\mathrm{q}}$$



He et al., 2019

MoCo: ablations

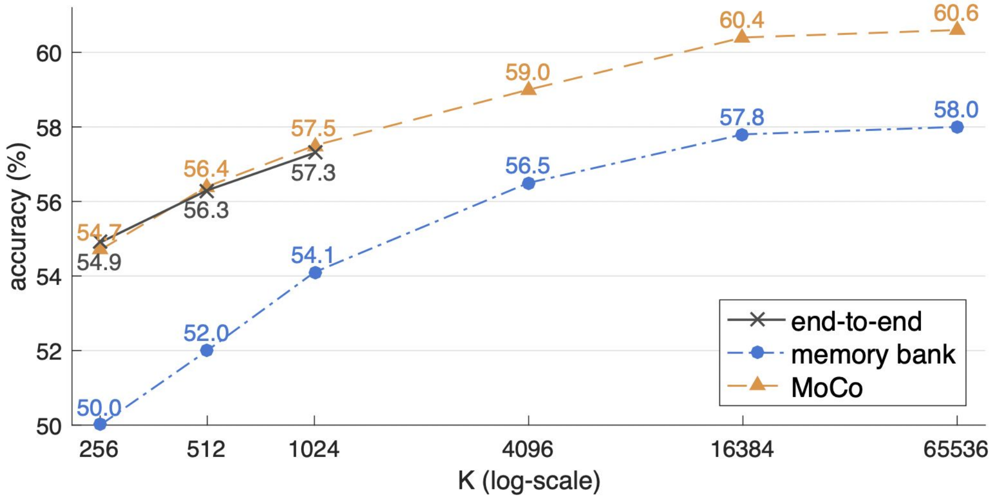


Figure 3. **Comparison of three contrastive loss mechanisms** under the ImageNet linear classification protocol. We adopt the same pretext task (Sec. 3.3) and only vary the contrastive loss mechanism (Figure 2). The number of negatives is K in memory bank and MoCo, and is $K-1$ in end-to-end (offset by one because the positive key is in the same mini-batch). The network is ResNet-50.

Ablation: momentum. The table below shows ResNet-50 accuracy with different MoCo momentum values (m in Eqn.(2)) used in pre-training ($K = 4096$ here) :

momentum m	0	0.9	0.99	0.999	0.9999
accuracy (%)	*fail*	55.2	57.8	59.0	58.9

He et al., 2019

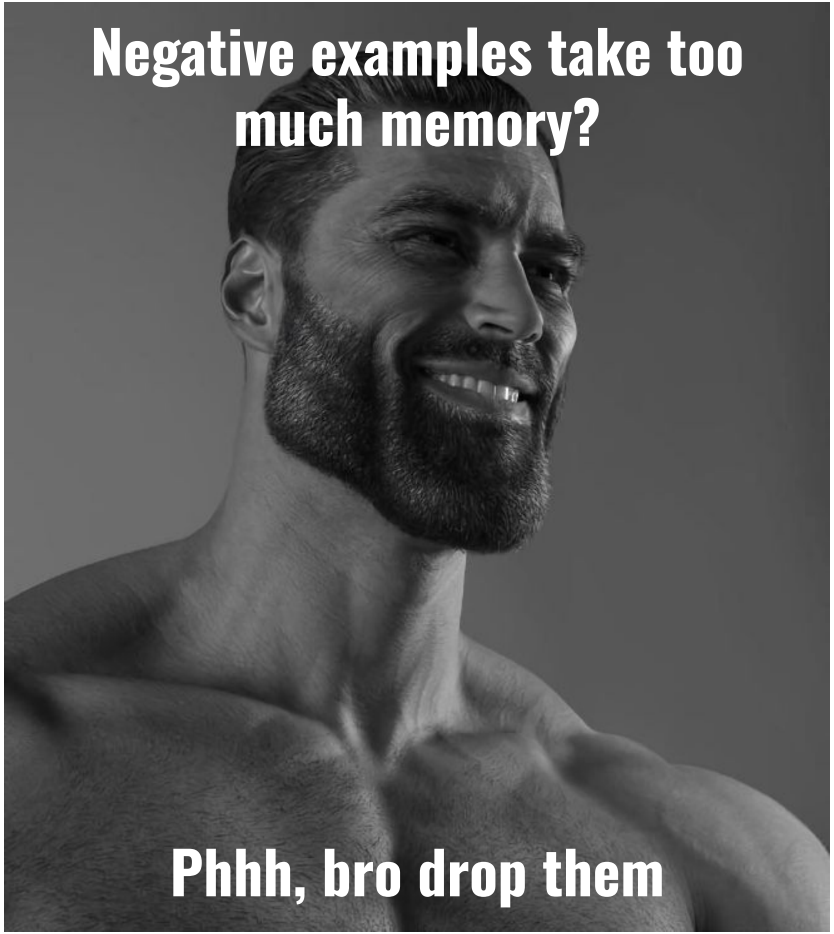
Plan

- **Introduction to information theory**
  - Mutual information & InfoNCE
- **Contrastive learning with negative examples**
  - SimCLR, MoCo
- **Contrastive learning without negative examples**
  - BYOL, SimSiam
- **Contrastive learning inspired by clustering**
  - DeepCluster, SwAV
- **Bonus**
  - Dense Contrastive Learning
  - Supervised Contrastive Learning

BYOL

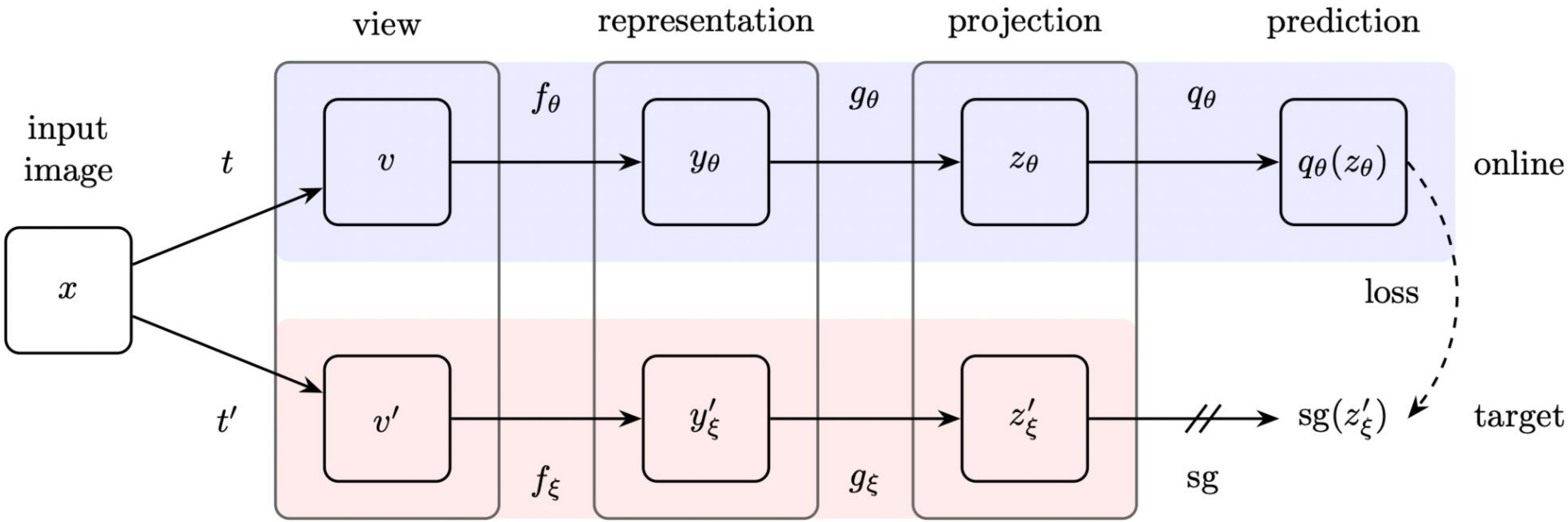
Bootstrap **Y**our **O**wn **L**atent, DeepMind

- Further development of momentum encoder from MoCo
- Contrastive learning **without negative examples**
- SOTA in pure contrastive methods



Grill et al., 2020

BYOL



$$\mathcal{L}_{\theta,\xi} \triangleq \left\| \overline{q_\theta}(z_\theta) - \bar{z}'_\xi \right\|_2^2 = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\left\| q_\theta(z_\theta) \right\|_2 \cdot \left\| z'_\xi \right\|_2}$$

$$\theta \leftarrow \mathrm{optimizer}\left(\theta, \nabla_\theta \mathcal{L}^{\texttt{BYOL}}_{\theta,\xi}, \eta\right),$$
$$\xi \leftarrow \tau\xi + (1-\tau)\theta,$$

Grill et al., 2020

BYOL: representation collapse

Why do we even need negative examples in contrastive learning?

$$\mathcal{L}_{\theta,\xi} \triangleq \left\|\overline{q_\theta}(z_\theta) - \overline{z}'_\xi\right\|_2^2$$

What if we take constant representations? We end up in a **global minimum** of the loss function with a **representation collapse**!

Grill et al., 2020

BYOL: representation collapse

Why does BYOL not collapse in practice?

- Optimization steps are not in the direction of joint gradient $\nabla_{\theta,\xi} L$

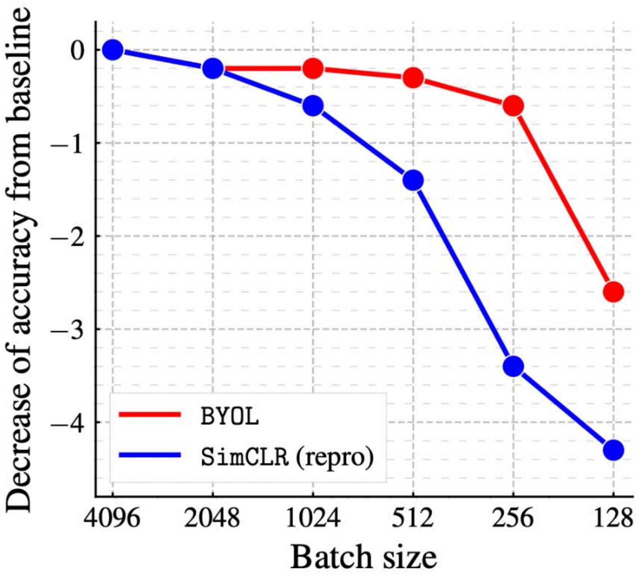
$$\theta \leftarrow \text{optimizer}\left(\theta, \nabla_\theta \mathcal{L}^{\texttt{BYOL}}_{\theta,\xi}, \eta\right),$$
$$\xi \leftarrow \tau\xi + (1-\tau)\theta,$$

- Collapsed global minimum is unstable

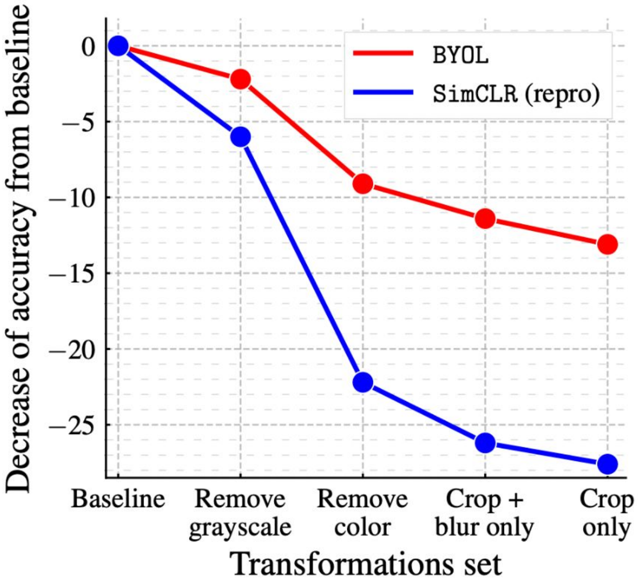
- Hyperparameters (EMA, weight decay) are important

Grill et al., 2020

BYOL: ablations



(a) Impact of batch size



(b) Impact of progressively removing transformations

Figure 3: Decrease in top-1 accuracy (in % points) of BYOL and our own reproduction of SimCLR at 300 epochs, under linear evaluation on ImageNet.

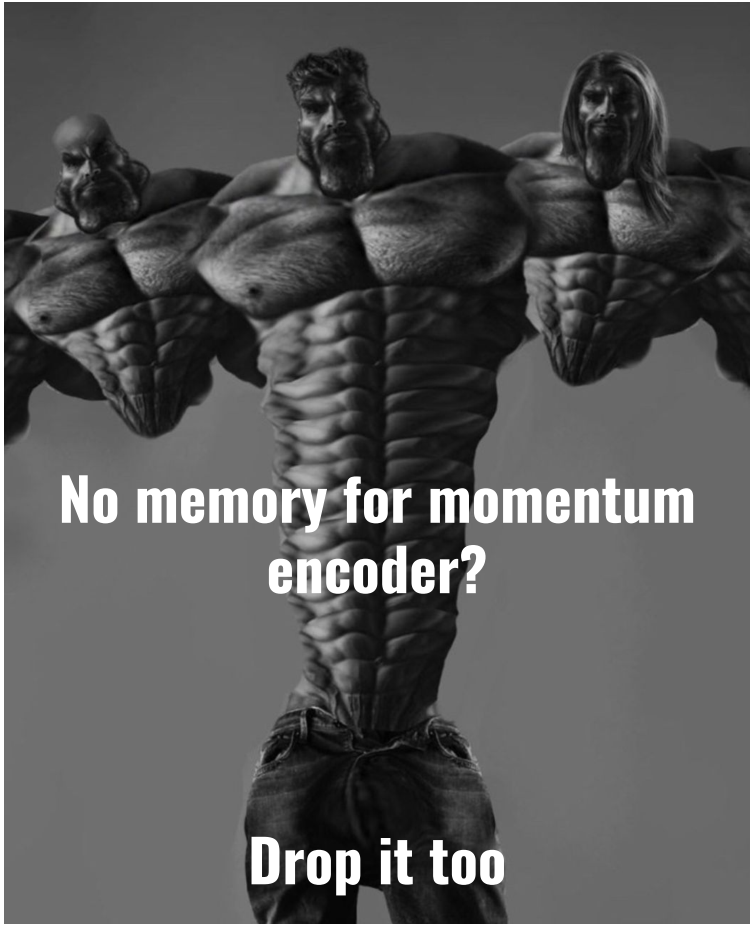
Grill et al., 2020

SimSiam

Simple **Siam**ese networks, FAIR

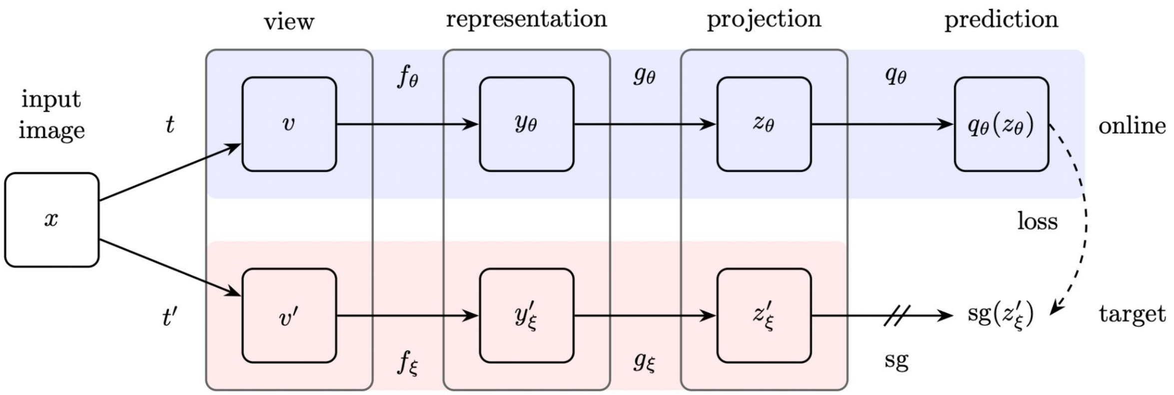
- Momentum encoder is **unnecessary** for contrastive learning

- Overall performance worse than BYOL but still comparable to other methods

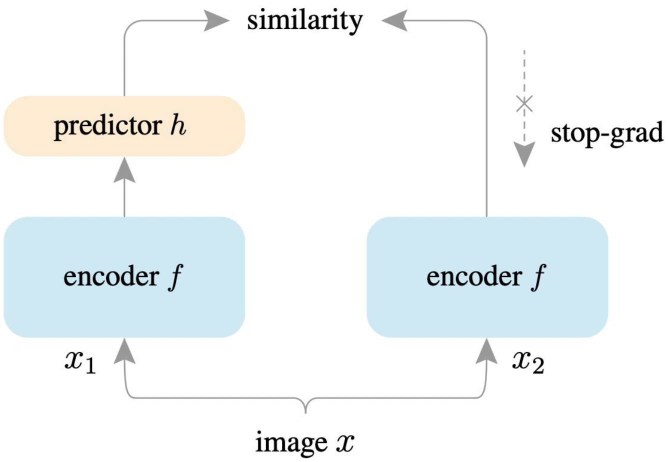


[Chen and He, 2020](#)

SimSiam vs. BYOL



BYOL

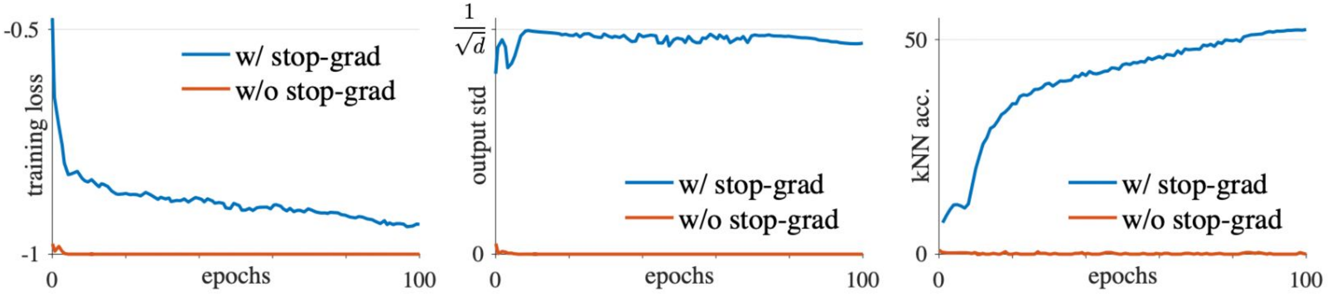


SimSiam*

*SimSiam encoder ***f*** includes both the convolutional part (BYOL's $\boldsymbol{f_\theta}$) and the MLP projection (BYOL's $\boldsymbol{g_\theta}$)

Chen and He, 2020

SimSiam: ablations



	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).

	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).

Chen and He, 2020

SimSiam is EM algorithm???

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

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

x – image
$\mathcal{T}(\cdot)$ – augmentation
$\mathcal{F}_\theta(\cdot)$ – encoder
η_x – representation of image x

[Chen and He, 2020]

SimSiam is EM algorithm???

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

$$\theta^t \quad \leftarrow \quad \arg\min_\theta \ \mathcal{L}(\theta, \eta^{t-1})$$

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

x – image

$\mathcal{T}(\cdot)$ – augmentation

$\mathcal{F}_\theta(\cdot)$ – encoder

η_x – representation of image x

Chen and He, 2020

SimSiam is EM algorithm???

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

$$\eta_x^t \leftarrow \cancel{\mathbb{E}_{\mathcal{T}}\left[\mathcal{F}_{\theta^t}(\mathcal{T}(x))\right]} \longrightarrow \eta_x^t \leftarrow \mathcal{F}_{\theta^t}(\mathcal{T}'(x))$$

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

x – image

$\mathcal{T}(\cdot)$ – augmentation

$\mathcal{F}_\theta(\cdot)$ – encoder

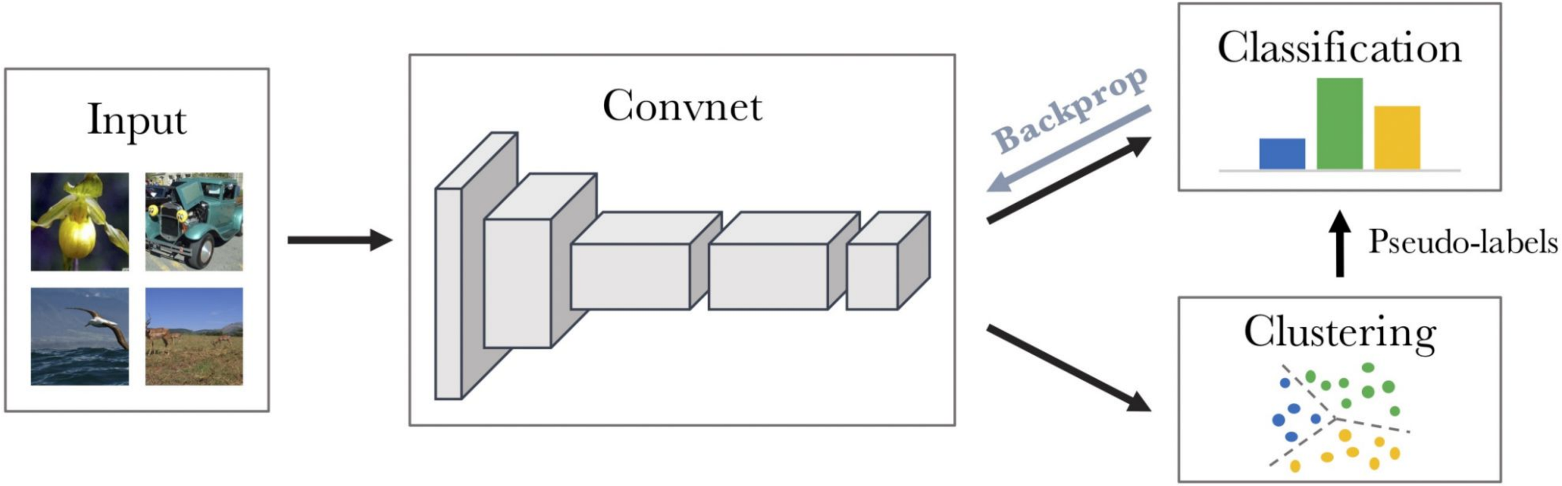
η_x – representation of image x

Chen and He, 2020

Plan

- **Introduction to information theory**
  - Mutual information & InfoNCE
- **Contrastive learning with negative examples**
  - SimCLR, MoCo
- **Contrastive learning without negative examples**
  - BYOL, SimSiam
- **Contrastive learning inspired by clustering**
  - DeepCluster, SwAV
- **Bonus**
  - Dense Contrastive Learning
  - Supervised Contrastive Learning

DeepCluster



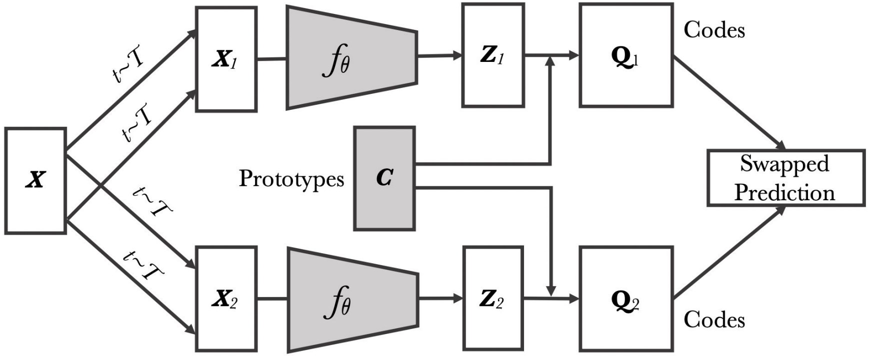
$$\min_{C \in \mathbb{R}^{d \times k}} \frac{1}{N} \sum_{n=1}^{N} \min_{y_n \in \{0,1\}^k} \| f_\theta(x_n) - C y_n \|_2^2 \quad \text{such that} \quad y_n^\top 1_k = 1$$

Caron et al., 2018

SwAV

Swapping **A**ssignments between **V**iews, FAIR

- Further development of DeepCluster
- Make cluster centroids learnable
- Do online clustering
- Swap cluster assignments between two augmented views of the image (i.e. bring contrastive component to the clustering setup)

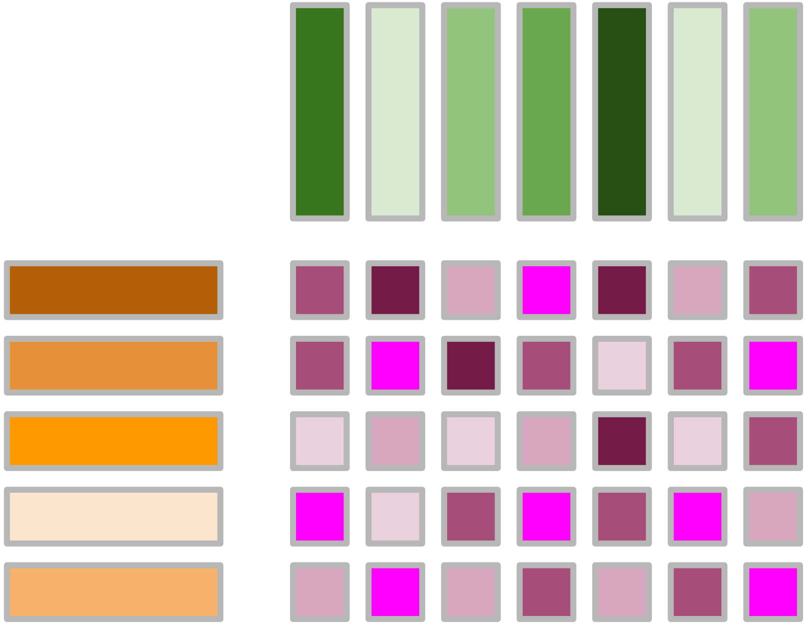


Caron et al., 2020

SwAV: scheme

Image representations

$$\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_B]$$



$$\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_K]$$

Cluster prototypes (“centroids”)

$$\mathbf{Q} \in \mathbb{R}_+^{K \times B}$$

Cluster assignments

Caron et al., 2020

SwAV: scheme

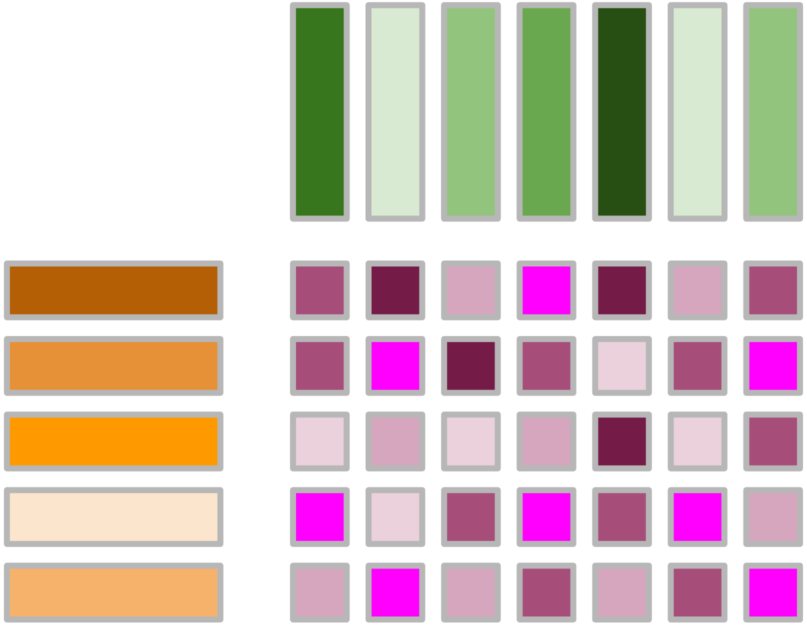
Image representations

$$\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_B]$$

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

$$H(\mathbf{Q}) = -\sum_{ij} \mathbf{Q}_{ij} \log \mathbf{Q}_{ij}$$

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



$$\mathbf{Q} \in \mathbb{R}_+^{K \times B}$$

Cluster assignments

$$\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_K]$$

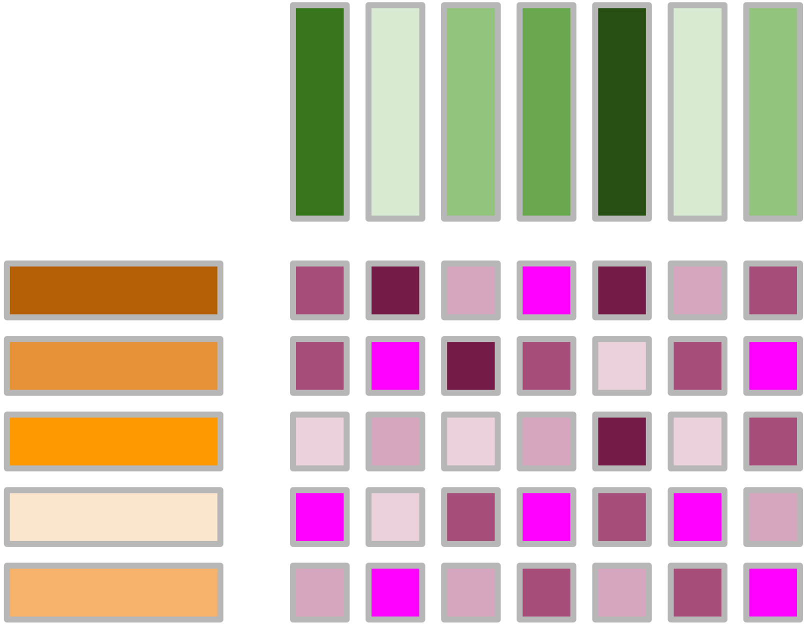
Cluster prototypes (“centroids”)

Caron et al., 2020

SwAV: scheme

Image representations

$$\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_B]$$



$$\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_K]$$

Cluster prototypes (“centroids”)

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

$$H(\mathbf{Q}) = -\sum_{ij} \mathbf{Q}_{ij} \log \mathbf{Q}_{ij}$$

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

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

$$\mathbf{Q} \in \mathbb{R}_+^{K \times B}$$

Cluster assignments

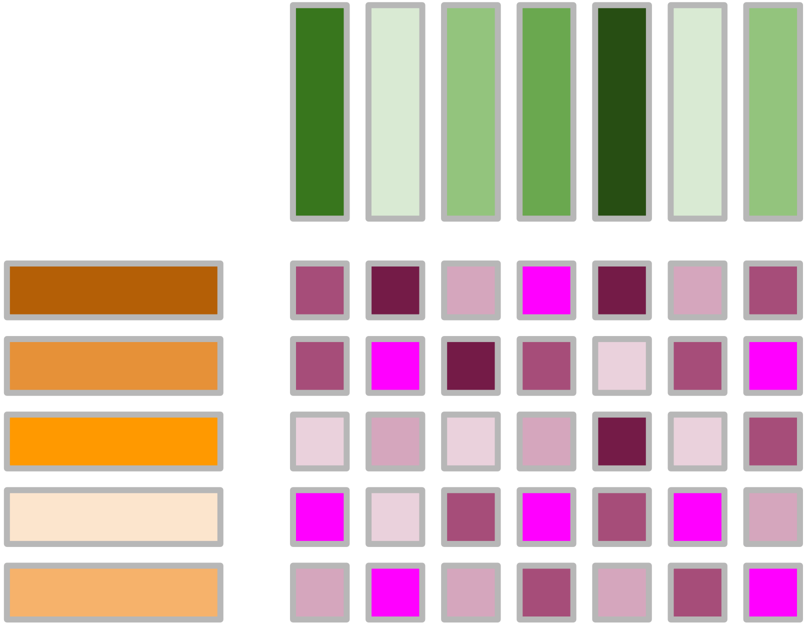
Sinkhorn–Knopp algorithm

Caron et al., 2020

SwAV: scheme

Image representations

$$\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_B]$$



$$\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_K]$$

Cluster prototypes (“centroids”)

$$\mathbf{Q} \in \mathbb{R}_+^{K \times B}$$

Cluster assignments

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

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

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

Caron et al., 2020

SwAV: scheme

Image representations

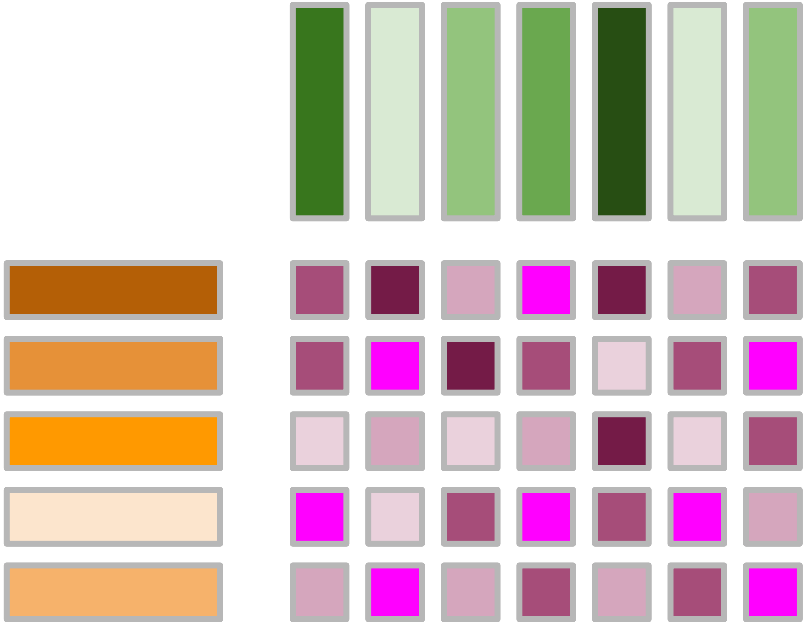
$$\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_B]$$

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

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

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

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



$$\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_K]$$

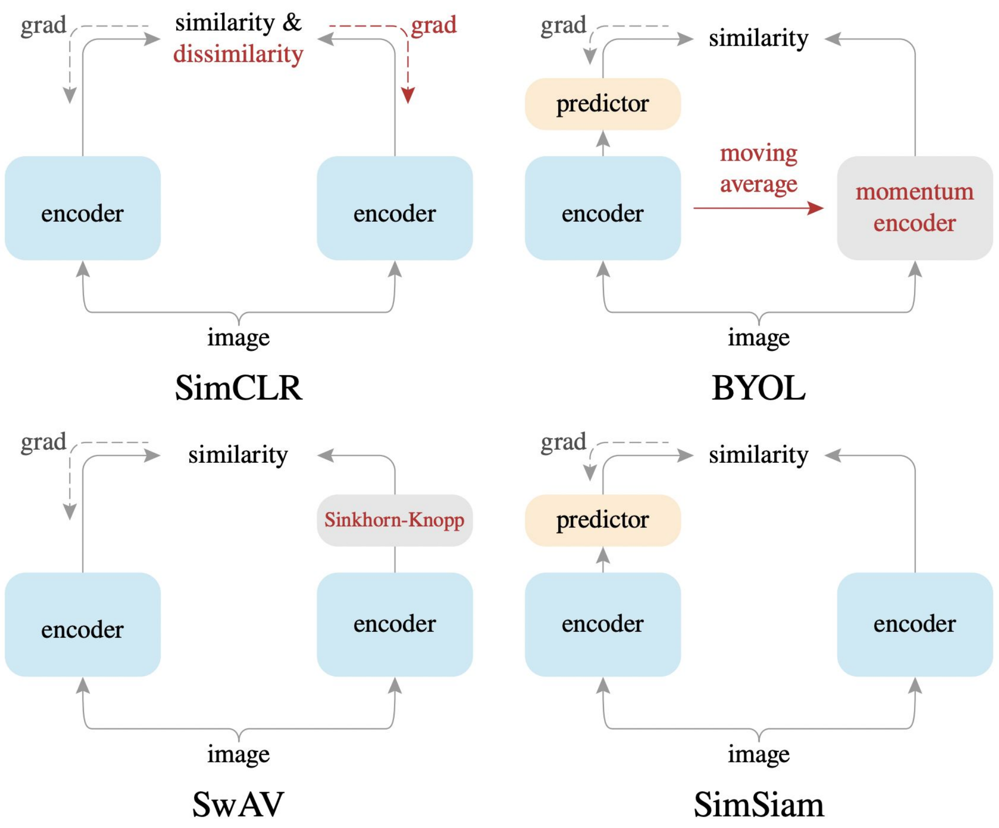
Cluster prototypes (“centroids”)

$$\mathbf{Q} \in \mathbb{R}_+^{K \times B}$$

Cluster assignments

Caron et al., 2020

Comparison



Chen and He, 2020

Results: ImageNet LP, VOC & COCO

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

	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**

Chen and He, 2020

Results: downstream classification

Method	Food101	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Linear evaluation:												
BYOL (ours)	**75.3**	91.3	**78.4**	**57.2**	**62.2**	**67.8**	60.6	82.5	75.5	90.4	94.2	**96.1**
SimCLR (repro)	72.8	90.5	74.4	42.4	60.6	49.3	49.8	81.4	**75.7**	84.6	89.3	92.6
SimCLR [8]	68.4	90.6	71.6	37.4	58.8	50.3	50.3	80.5	74.5	83.6	90.3	91.2
Supervised-IN [8]	72.3	**93.6**	78.3	53.7	61.9	66.7	**61.0**	**82.8**	74.9	**91.5**	**94.5**	94.7
Fine-tuned:												
BYOL (ours)	**88.5**	**97.8**	86.1	**76.3**	63.7	91.6	**88.1**	**85.4**	**76.2**	91.7	**93.8**	97.0
SimCLR (repro)	87.5	97.4	85.3	75.0	63.9	91.4	87.6	84.5	75.4	89.4	91.7	96.6
SimCLR [8]	88.2	97.7	85.9	75.9	63.5	91.3	88.1	84.1	73.2	89.2	92.1	97.0
Supervised-IN [8]	88.3	97.5	**86.4**	75.8	**64.3**	**92.1**	86.0	85.0	74.6	**92.1**	93.3	**97.6**
Random init [8]	86.9	95.9	80.2	76.1	53.6	91.4	85.9	67.3	64.8	81.5	72.6	92.0

Grill et al., 2020

Results: semi-supervised

Method	Top-1		Top-5	
	1%	10%	1%	10%
Supervised [77]	25.4	56.4	48.4	80.4
InstDisc	-	-	39.2	77.4
PIRL [35]	-	-	57.2	83.8
SimCLR [8]	48.3	65.6	75.5	87.8
BYOL (ours)	**53.2**	**68.8**	**78.4**	**89.0**

(a) ResNet-50 encoder.

Method	Architecture	Param.	Top-1		Top-5	
			1%	10%	1%	10%
CPC v2 [32]	ResNet-161	305M	-	-	77.9	91.2
SimCLR [8]	ResNet-50 (2×)	94M	58.5	71.7	83.0	91.2
BYOL (ours)	ResNet-50 (2×)	94M	62.2	73.5	84.1	91.7
SimCLR [8]	ResNet-50 (4×)	375M	63.0	74.4	85.8	92.6
BYOL (ours)	ResNet-50 (4×)	375M	69.1	75.7	87.9	92.5
BYOL (ours)	ResNet-200 (2×)	250M	**71.2**	**77.7**	**89.5**	**93.7**

(b) Other ResNet encoder architectures.

Table 2: Semi-supervised training with a fraction of ImageNet labels.

Grill et al., 2020