Self-Supervised Learning

Marina Munkhoeva
Research Scientist, AIRI

HSE, 2025-10-09

00

Outline

Outline

01

Recap and Preliminaries

Recap

Previously, we covered various pretext tasks in vision:

Temporal coherence

- Slow Feature Analysis
- Frame order
- Visual odometry
- Ranking tracking frames

Spatial coherence

- Relative patch prediction
- Jigsaw puzzles
- Rotation prediction

Domain knowledge

- Inpainting
- Patch-based pseudo-classes
- Colorization

Overview

From now on we will cover contemporary methods, **roughly** grouped into:

- Sample Contrastive Learning
  (SimCLR & friends)
- Clustering-based
- Self-Distillation
- Feature Contrastive Learning
- Whitening-based
- Masked Image Modelling

Mutual Information

Differential Entropy $h(X) = -\int_{\mathcal{X}} p(x) \log p(x) dx$

Can $h(X) < 0$?

Mutual Information

Differential Entropy $h(X) = -\int_{\mathcal{X}} p(x) \log p(x) dx$

Can $h(X) < 0$? Yes, consider $X \sim U(0, a)$, then $h(X) = \log a < 0$ when $a < 1$.

Generally, it's due p being density, i.e. no longer a probability as in discrete case.

Mutual Information

Differential Entropy $h(X) = -\int_{\mathcal{X}} p(x) \log p(x) dx$

Mutual Information

$$
\begin{aligned}
I(X;Y) &= D_{\mathrm{KL}}(P(X,Y) \parallel P(X)P(Y)) \\
&= \int p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy = \int p(x,y) \log \frac{p(x|y)}{p(x)} dxdy \\
I(X;Y) &= h(X) - h(X|Y) = h(X) + h(Y) - h(X,Y)
\end{aligned}
$$

Mutual Information

Differential Entropy $h(X) = -\int_{\mathcal{X}} p(x) \log p(x) dx$

Mutual Information

$$I(X;Y) = D_{\mathrm{KL}}(P(X,Y) \parallel P(X)P(Y))$$

$$= \int p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy = \int p(x,y) \log \frac{p(x|y)}{p(x)} dxdy$$

$$I(X;Y) = h(X) - h(X|Y) = h(X) + h(Y) - h(X,Y)$$

$I(X;Y)$ measures how much knowing Y reduces uncertainty about X.

Properties:

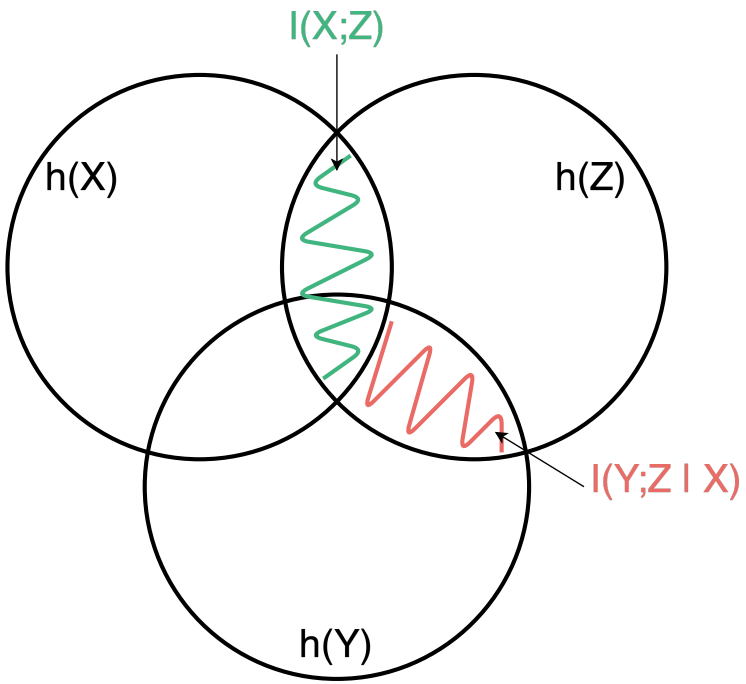
- Non-negativity: $I(X;Y) \geq 0$ with equality iff X and Y are independent
- Symmetry: $I(X;Y) = I(Y;X)$
- for $Z = f(X)$ with deterministic f, $I(X;Z) = H(Z) - H(Z \mid X) = H(Z)$

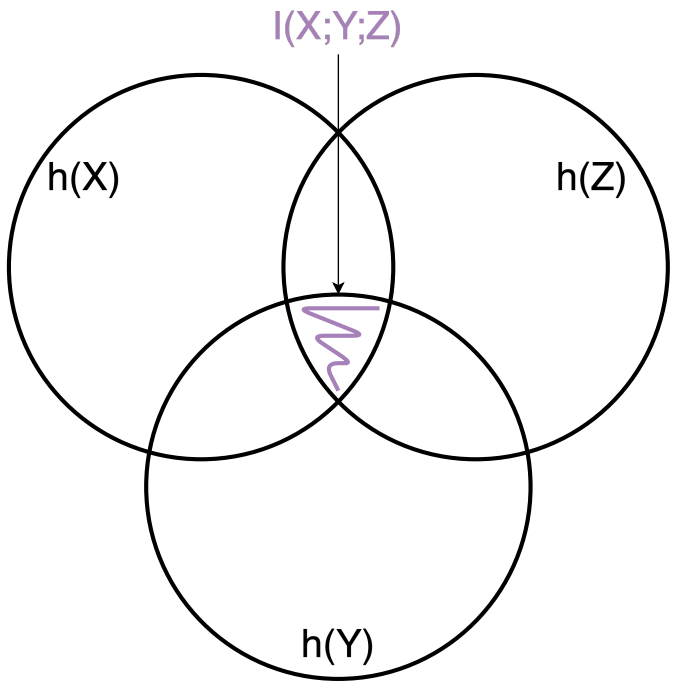
Mutual Information

Conditional Mutual Information $I(X;Y \mid Z) = h(X \mid Z) + h(Y \mid Z) - h(X,Y \mid Z)$

Multivariate MI for 3 random variables $I(X;Y;Z) = I(Y,Z) - I(Y;Z \mid X)$

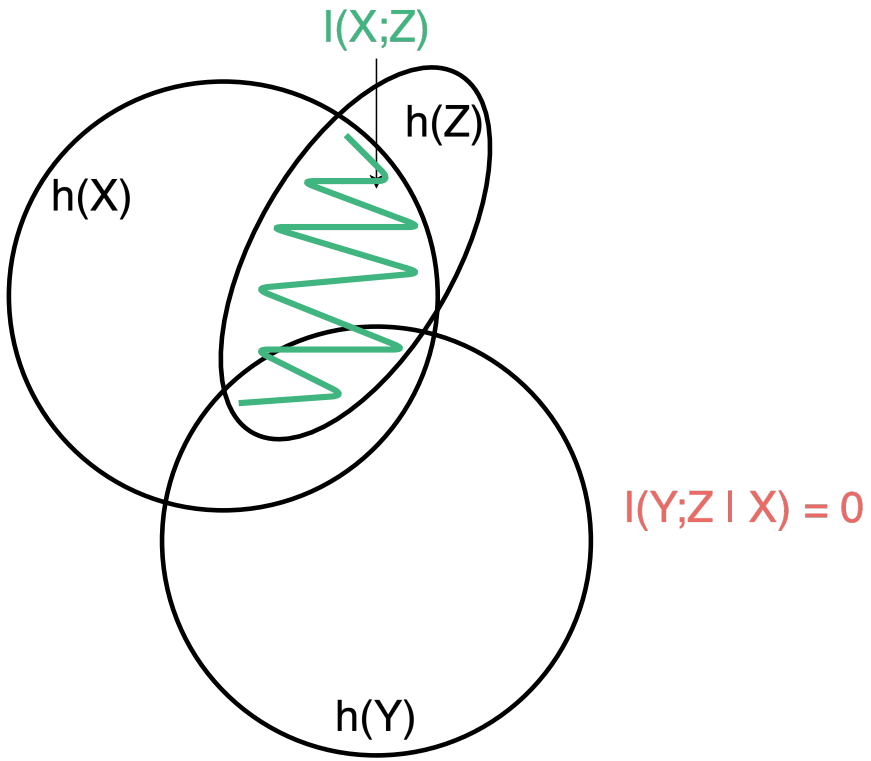
Venn diagram illustration for measures of information

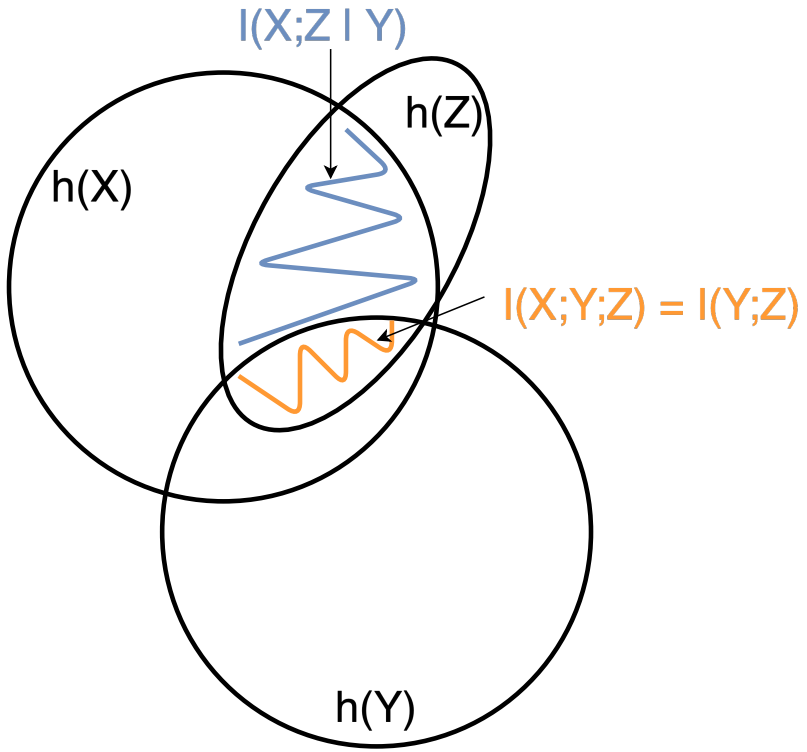




Mutual Information

$$I(Y; Z \mid X) = 0$$





Multi-View Redundancy

Multi-view redundancy assumption (Sridharan & Kakade, 2008)

Let Y — some task targets, X_1, X_2 — two views of the same data

There exists an $\varepsilon > 0$ such that

$$I(Y; X_1|X_2) \leq \varepsilon \text{ and}$$
$$I(Y; X_2|X_1) \leq \varepsilon$$

When $\varepsilon = 0$, we will learn nothing new about Y by observing X_1 if we already know X_2, i.e. $Y \perp\!\!\!\perp X_1 \mid X_2$.

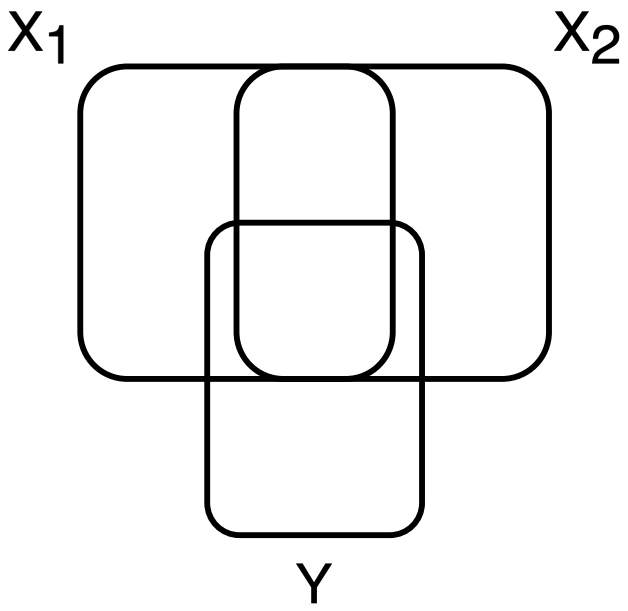
NB small ε **doesn't** mean Y is perfectly predictable from either X_i.

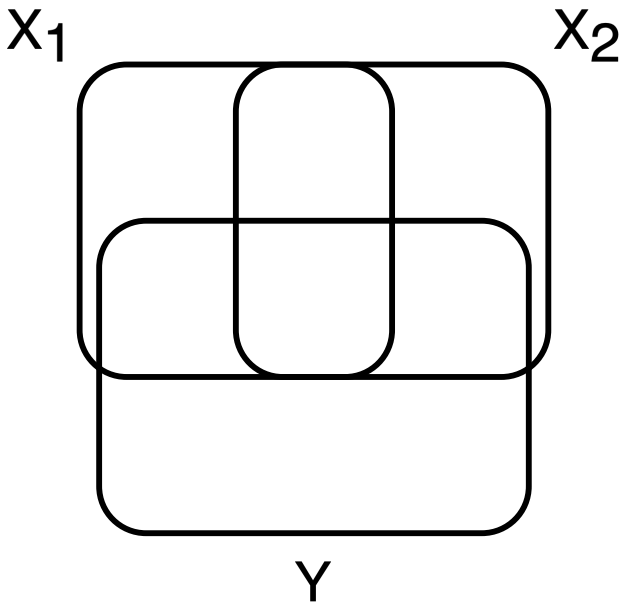
Multi-View Redundancy

There exists an $\varepsilon > 0$ such that

$$I(Y; X_1 | X_2) \leq \varepsilon \text{ and}$$
$$I(Y; X_2 | X_1) \leq \varepsilon$$

Which diagram better suits the assumption?





Multi-View Redundancy

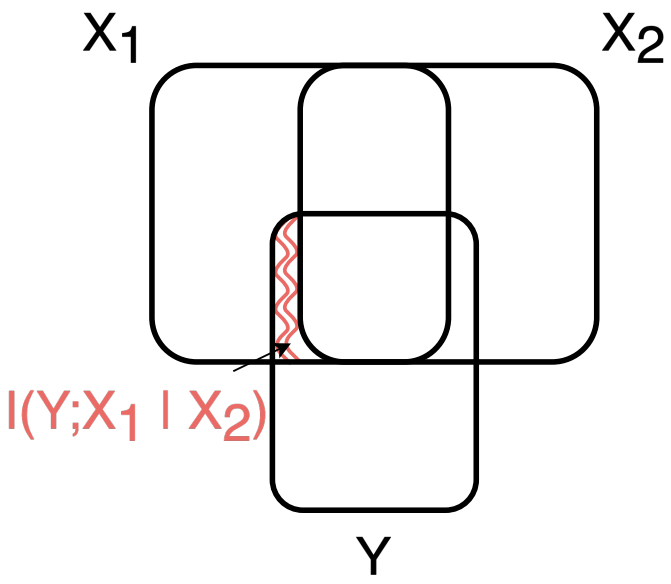
There exists an $\varepsilon > 0$ such that

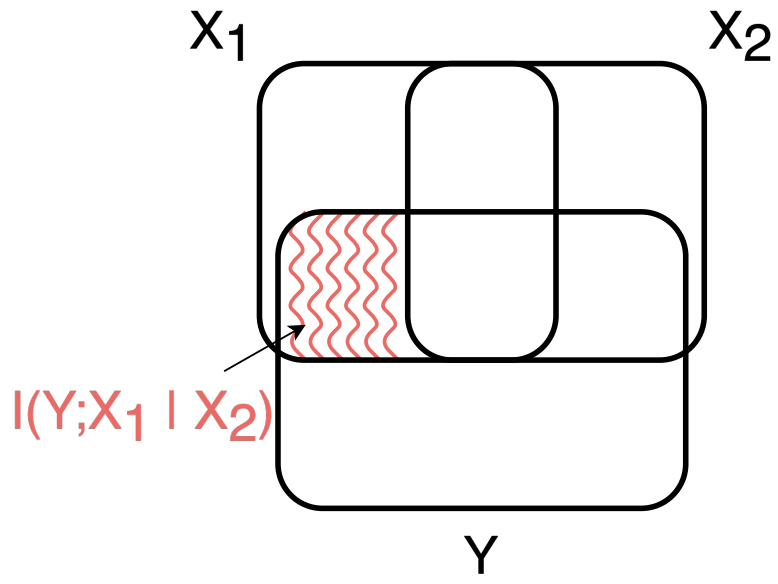
$$I(Y; X_1|X_2) \leq \varepsilon \text{ and}$$
$$I(Y; X_2|X_1) \leq \varepsilon$$

Which diagram better suits the assumption?

– both seem okay if ε is large enough

– where ε is smaller?





Low/high shared info

High shared information (views preserve semantics) << this lecture

- Examples: standard image augs of same object, multi-crop, light jitter.
- Properties: strong, dense signal; many predictive invariances exist.

Low shared info (views share only sparse/abstract bits)

- Examples: cross-modal (image–text, audio–video), heavy corruptions, large domain gaps.
- Properties: weak or fragmentary alignment; many false correspondences.

02

Sample CL

Contrastive Predictive Coding

Given context c, predict observation x **without** directly modelling conditional $p(x|c)$

Maximally preserve MI between x and c: $I(x, c) = \sum_{x,c} p(x, c) \log \frac{p(x|c)}{p(x)}$

CPC (Oord et al., 2018):

1. Encode $z_t = g_{\text{enc}}(x_t)$; summarize context $c_t = g_{\text{ar}}(z_{\leq t})$
2. Model density ratio $f_k(x_{t+k}, c_t) \propto \frac{p(x_{t+k}|\ c_t)}{p(x_{t+k})}$ as $f_k(x_{t+k}, c_t) := \exp(z_{t+k}^\top W_k c_t)$
3. Noise-Contrastive Estimation:

$$\mathcal{L}_{\text{InfoNCE}} = -\mathbb{E}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right]$$

InfoNCE

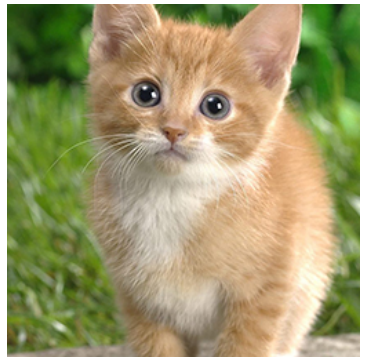
$(x, c) \sim p(x|c)$ — positive pair

$x' \sim p(x)$, (x', c) — negative pair

Classify c into N classes, where $y = i$ if (x_i, c) — positive pair

c

x_1



x_2

...

x_N

InfoNCE

$(x, c) \sim p(x|c)$ — positive pair

$x' \sim p(x)$, (x', c) — negative pair

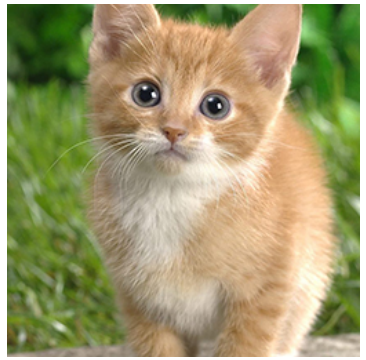
Classify c into N classes, where $y = i$ if (x_i, c) — positive pair

Cross-entropy: $\mathcal{L}_{\text{InfoNCE}} = -\mathbb{E}_X \left[\log \frac{f(x,c)}{\sum_{x' \in X} f(x',c)} \right]$

where $\frac{f_k}{\sum_X f_k}$ is model prediction

c

x_1



x_2

...

x_N

InfoNCE

$(x, c) \sim p(x|c)$ — positive pair

$x' \sim p(x)$, (x', c) — negative pair

Classify c into N classes, where $y = i$ if (x_i, c) — positive pair

Cross-entropy: $\mathcal{L}_{\text{InfoNCE}} = -\mathbb{E}_X \left[\log \frac{f(x_i, c)}{\sum_{x_j \in X} f(x_j, c)} \right]$

where $\frac{f_k}{\sum_X f_k}$ is model prediction

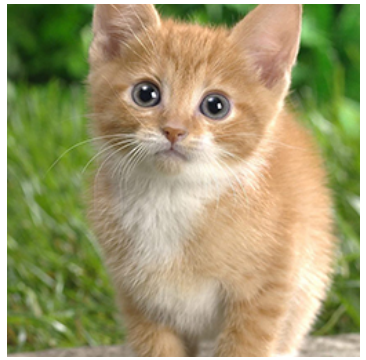
Optimal probability of $x_i \sim p(x|c)$ rather than proposal $p(x)$

$$p(y = i \mid X, c) = \frac{p(x_i|c) \prod_{l \neq i} p(x_l)}{\sum_{j=1}^{N} p(x_j|c) \prod_{l \neq j} p(x_l)} \propto \frac{p(x_i|c)}{p(x_i)}$$

$\min \mathcal{L}_{\text{InfoNCE}}$ yields optimal $f(x, c)$ estimating density ratio $\frac{p(x|c)}{p(x)}$

c

x_1



x_2

...

x_N

InfoNCE - a lower bound for MI

Plug in the optimal value for $f(x, c)$ into the loss (Oord et al., 2018):

$$\mathcal{L}^{\text{opt}}_{\text{InfoNCE}} = -\mathbb{E}_X \left[\log \frac{\frac{p(x|c)}{p(x)}}{\frac{p(x|c)}{p(x)} + \sum_{x' \in X_{\text{neg}}} \frac{p(x'|c)}{p(x')}} \right] = \mathbb{E}_X \log \left[1 + \frac{p(x)}{p(x|c)} \sum_{x' \in X_{\text{neg}}} \frac{p(x'|c)}{p(x')} \right]$$

$$\approx \mathbb{E}_X \log \left[1 + \frac{p(x)}{p(x|c)} (N-1) \mathbb{E}_{x'} \frac{p(x'|c)}{p(x')} \right] = \mathbb{E}_X \log \left[1 + \frac{p(x)}{p(x|c)} (N-1) \right]$$

$$\geq \mathbb{E}_X \log \left[\frac{p(x)}{p(x|c)} N \right] = -\mathbb{E}_X \log \frac{p(x|c)}{p(x)} + \log(N) = \log(N) - I(x, c)$$

$$I(x, c) \geq \log(N) - \mathcal{L}^{\text{opt}}_{\text{infoNCE}}$$

NB approximation, assumption $\frac{p(x)}{p(x|c)} < 1$

otherwise $\mathbb{E}_X \log \left[1 + \frac{p(x)}{p(x|c)} (N-1) \right] \geq \mathbb{E}_X \log \left[\frac{p(x)}{p(x|c)} (N-1) \right]$

Lower Bounds on MI (I)

A better derivation (Poole et al., 2019) via variational LB:

$$
\begin{aligned}
I(X;Y) &= \mathbb{E}_{p(x,y)} \log \frac{p(x,y)}{p(x)p(y)} = \mathbb{E}_{p(x,y)} \log \frac{p(x \mid y)}{p(x)} \\
&= \mathbb{E}_{p(x,y)} \log \frac{q(x|y)}{p(x)} \frac{p(x|y)}{q(x|y)} \\
&= \mathbb{E}_{p(x,y)} \log \frac{q(x|y)}{p(x)} + \mathbb{E}_{p(x|y)p(y)} \log \frac{p(x|y)}{q(x|y)} \\
&= \mathbb{E}_{p(x,y)} \log \frac{q(x|y)}{p(x)} + \mathbb{E}_{p(y)} D_{\mathrm{KL}}(p(x|y) \parallel q(x|y)) \\
&\geq \mathbb{E}_{p(x,y)} \log q(x|y) - \mathbb{E}_{p(x,y)} \log p(x) \\
&= \mathbb{E}_{p(x,y)} \log q(x|y) + h(X) \triangleq I_{\mathrm{BA}}
\end{aligned}
$$

Lower Bounds on MI (II)

I_{BA} remains intractable as $h(X)$ intractable

Energy-based variational family: $q(x|y) = \frac{p(x)}{Z(y)} e^{f(x,y)}$, where $Z(y) = \mathbb{E}_{p(x)}\left[e^{f(x,y)}\right]$,
$f(x, y)$ - critic function

$$\begin{aligned}
\mathbb{E}_{p(x,y)} \log q(x|y) + h(X) &= \mathbb{E}_{p(x,y)} \log \frac{p(x)}{Z(y)} e^{f(x,y)} + h(X) \\
&= \mathbb{E}_{p(x,y)} \log p(x) + \mathbb{E}_{p(x,y)} f(x,y) - \mathbb{E}_{p(y)} \log Z(y) + h(X) \\
&= \mathbb{E}_{p(x,y)} f(x,y) - \mathbb{E}_{p(y)} \log Z(y) \triangleq I_{\mathrm{UBA}}
\end{aligned}$$

Donsker-Varadhan bound obtained via Jensen's inequality for $\mathbb{E}_{p(y)}[\log Z(y)]$ term:

$$I_{\mathrm{UBA}} \geq \mathbb{E}_{p(x,y)}[f(x,y)] - \log \mathbb{E}_{p(y)p(x)}\left[e^{f(x,y)}\right] \triangleq I_{\mathrm{DV}}$$

$$I_{\mathrm{DV}} = \mathbb{E}_{P_{XY}} T(x,y) - \log \mathbb{E}_{P_X \otimes P_Y} e^{T(x,y)}$$

Lower Bounds on MI (III)

So far: $I(X;Y) \geq \mathbb{E}_{p(x,y)} f(x,y) - \mathbb{E}_{p(y)}[\log Z(y)] \triangleq I_{\mathrm{UBA}}$

$\log Z(y)$ still intractable :(How to get tractable UBA?

$\log(x) \leq \frac{x}{a} + \log(a) - 1 \ \forall x, a > 0 \log(Z(y)) \leq \frac{Z(y)}{a(y)} + \log(a(y)) - 1$

$$I \geq I_{\mathrm{UBA}} \geq \mathbb{E}_{p(x,y)}[f(x,y)] - \mathbb{E}_{p(y)}\left[\frac{\mathbb{E}_{p(x)} e^{f(x,y)}}{a(y)} + \log(a(y)) - 1\right] \triangleq I_{\mathrm{TUBA}}$$

Maximize I_{TUBA} wrt both $a(y)$ and f

Set $a = e$ to get **Nguyen-Wainwright-Jordan** (NWJ) bound:

$$\mathbb{E}_{p(x,y)}[f(x,y)] - e^{-1}\mathbb{E}_{p(y)}[Z(y)] \triangleq I_{\mathrm{NWJ}}$$

Optimal critic $f^*(x,y) = 1 + \log \frac{p(x|y)}{p(x)}$

InfoNCE lower bounds MI

Upper-bound on partition $Z(y)$ is high-variance! Reduce variance with multiple samples.

$x_1, y \sim p(x_1, y)$ — positive samples, $x_{2:N} \sim r^{K-1}$ — other than X_1

$X, Y \perp Z \to I(X, Z; Y) = I(X; Y)$

$I(X_1; Y) = \mathbb{E}_{r^{N-1}}[I(X_1; Y)] = I(X_1, X_{2:N}; Y)$

Estimate "new" multi-sample mutual information

$$I(X_1; Y) = I(X_1, X_{2:N}; Y) \geq I_{\mathrm{NWJ}}$$

$f^*(x_{1:N}, y) = 1 + \log \frac{e^{f(x_1, y)}}{a(y; x_{1:N})}$ $\qquad$ $a(y; x_{1:N}) = \frac{1}{N} \sum_i e^{f(x_i, y)} (\mathrm{MC})$

$r^{N-1}(x_{2:N}) = \prod_{j=2}^{N} p(x_j)$

InfoNCE lower bounds MI

$f(x_{1:N}, y) = 1 + \log \frac{e^{f(x_1,y)}}{a(y;x_{1:N})}$

substitute into partition function $Z(y) = \mathbb{E}_{p(x_{1:N})} e^{f(x_{1:N},y)} = e\mathbb{E}_{p(x_{1:N})} \frac{e^{f(x_1,y)}}{a(y;x_{1:N})}$

finally

$$I(X_1;Y) \geq \mathbb{E}_{p(x_{1:N},y)}[f(x_{1:N},y)] - e^{-1}\mathbb{E}_{p(y)}[Z(y)]$$

$$= 1 + \mathbb{E}_{p(x_{1:N},y)}\left[\log \frac{e^{f(x_1,y)}}{a(y;x_{1:N})}\right] - \mathbb{E}_{p(y)}\mathbb{E}_{p(x_{1:N})}\left[\frac{e^{f(x_1,y)}}{a(y;x_{1:N})}\right]$$

Red right term — normalized probability density with partition function $Z(y) = a(y;x_{1:N})$

InfoNCE lower bounds MI

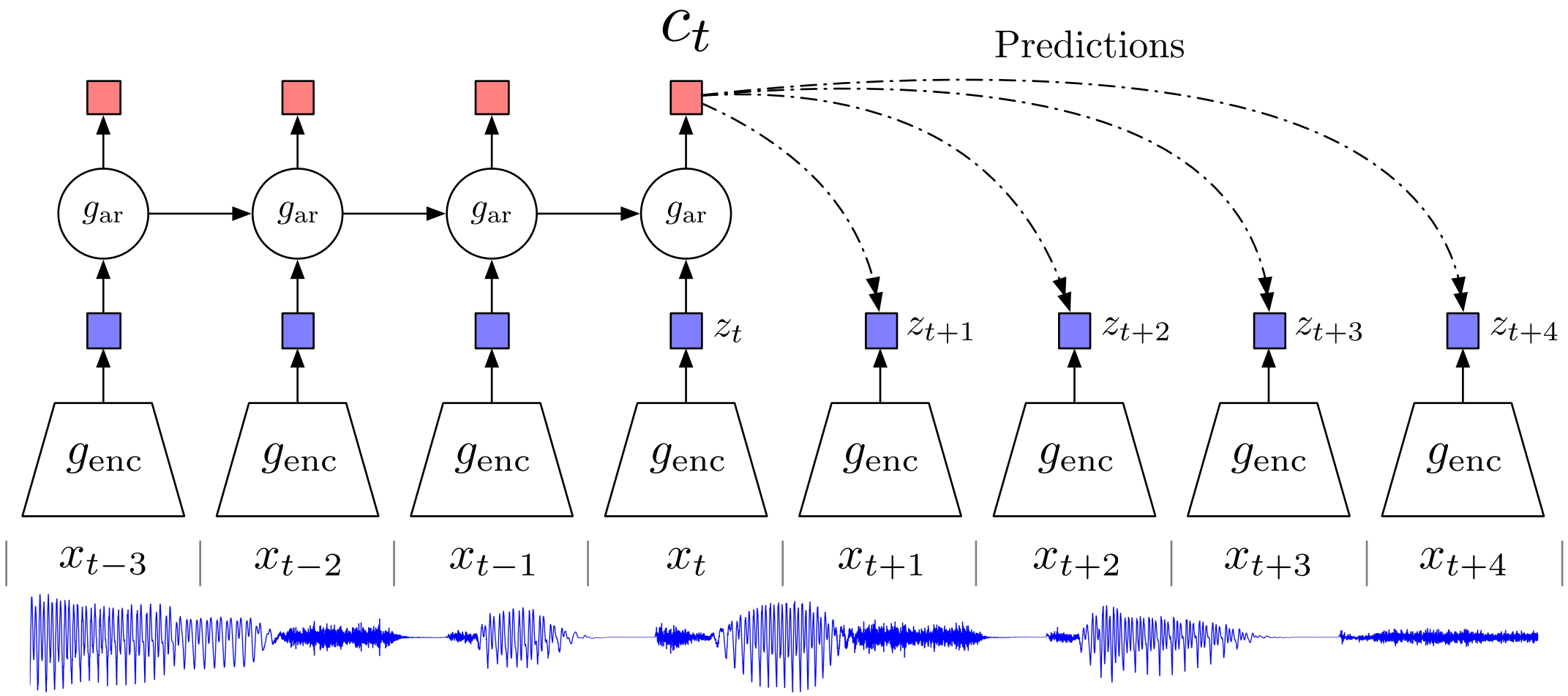
Finally,

$$I(X_1; Y) \geq \mathbb{E}_{p(x_{1:N}, y)} \log \frac{e^{f(x_1, y)}}{\frac{1}{N} \sum_{i=1}^{N} e^{f(x_i, y)}}$$

$$= \log(N) - \mathcal{L}_{\text{InfoNCE}}$$

For detailed exposition see (Poole et al., 2019)

CPC Audio

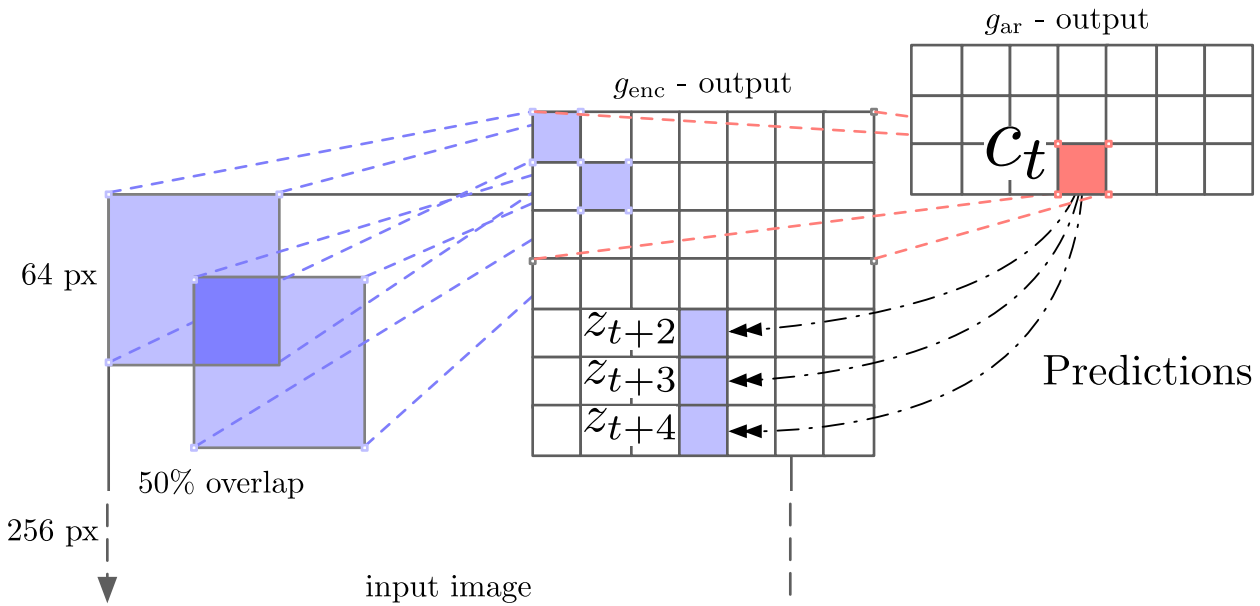


(Oord et al., 2018)

CPC Vision

In CPC:

- positive pairs – overlapping patches in image
- negative pairs – patches from distinct images



CPC (Oord et al., 2018)

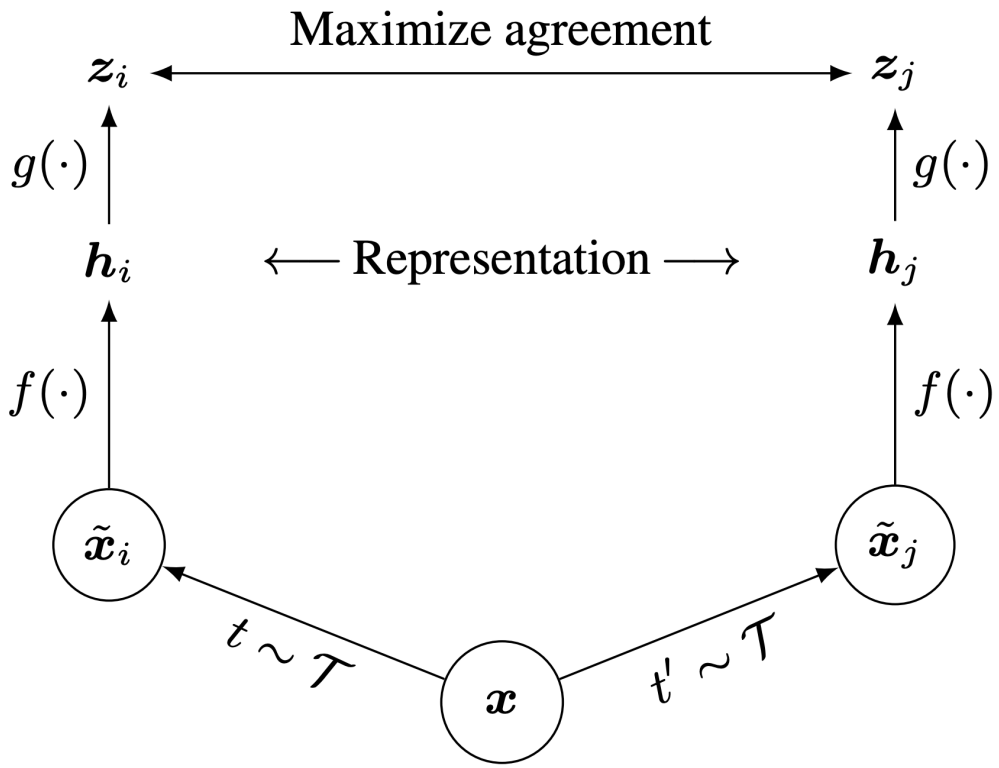
SimCLR

A Simple Framework for Contrastive Learning of Visual Representations (Chen et al., 2020)

SimCLR uses variant of InfoNCE loss

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

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



SimCLR insights

Many methods have been put forward architectural changes

- global to local view prediction
- neighbouring views
- context aggregation

How SimCLR stands out?

- decouples contrastive task from architectural choice by design

(no specialized architecture or memory bank — simply use random crops)

- show data augmentation is crucial for good representations
- show aggressive augmentation is beneficial

Architecture

SSL benefits more from increased training time and bigger model size

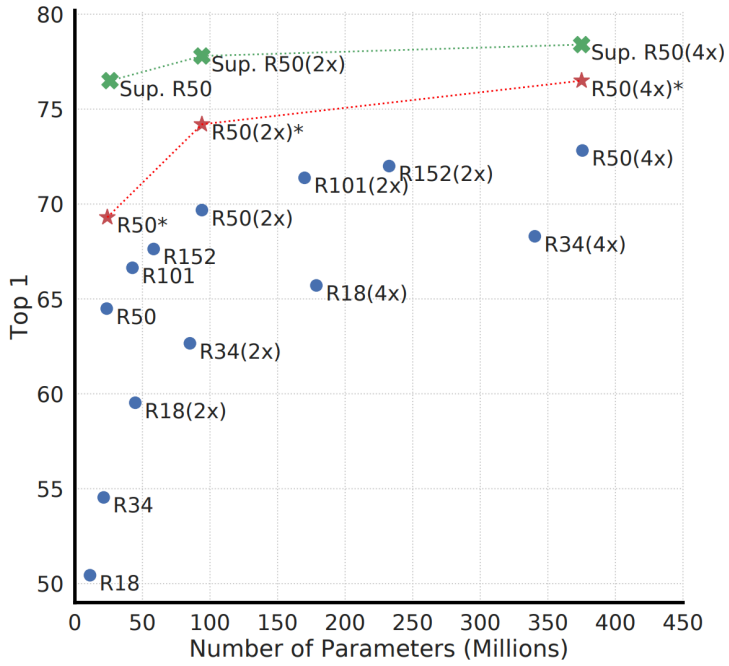


Figure 7. Linear evaluation of models with varied depth and width. Models in blue dots are ours trained for 100 epochs, models in red stars are ours trained for 1000 epochs, and models in green crosses are supervised ResNets trained for 90 epochs[7] (He et al., 2016).

Architecture

Projection head influence

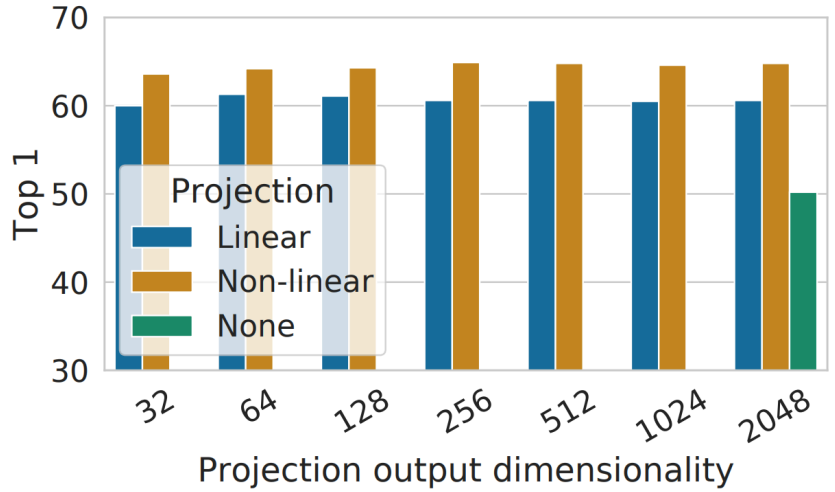


Figure 8. Linear evaluation of representations with different projection heads $g(\cdot)$ and various dimensions of $\boldsymbol{z} = g(\boldsymbol{h})$. The representation $\boldsymbol{h}$ (before projection) is 2048-dimensional here.

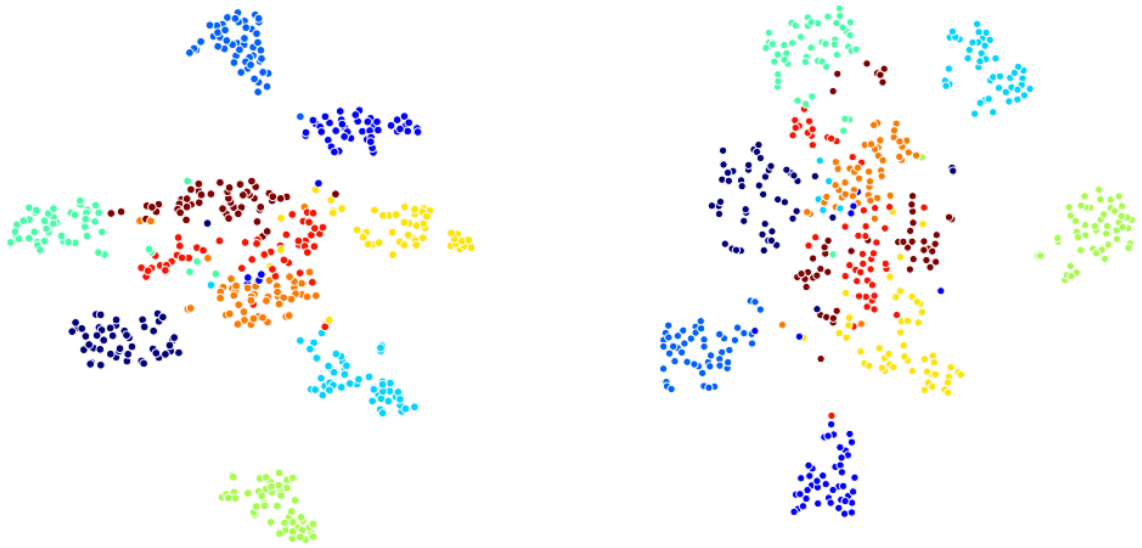
Embeddings lose information on transformation

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.

Architecture

Pre-projection head features are better separated



(a) $\boldsymbol{h}$ (b) $\boldsymbol{z} = g(\boldsymbol{h})$

Figure B.4. t-SNE visualizations of hidden vectors of images from a randomly selected 10 classes in the validation set.

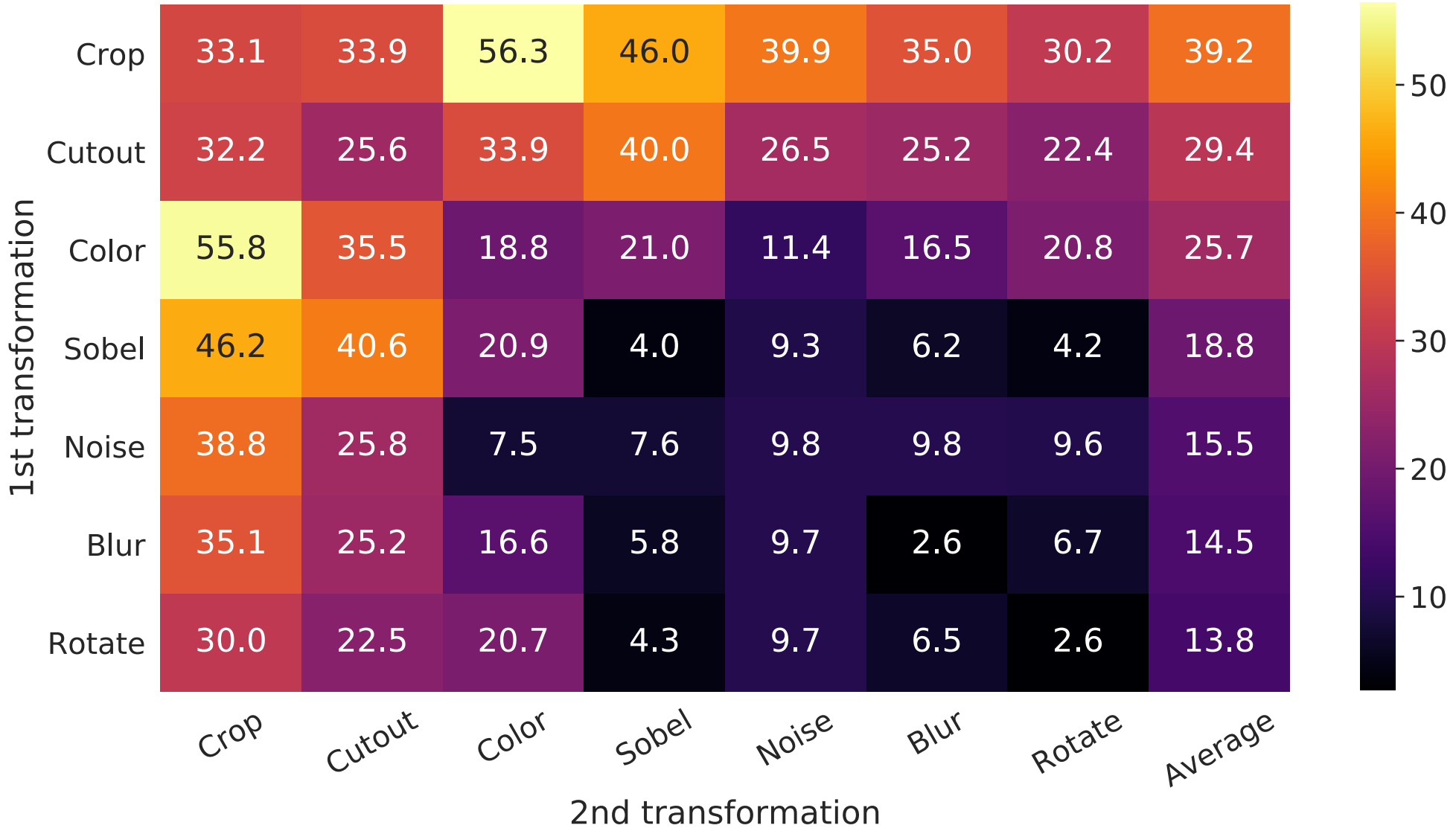
Augmentations matter

(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

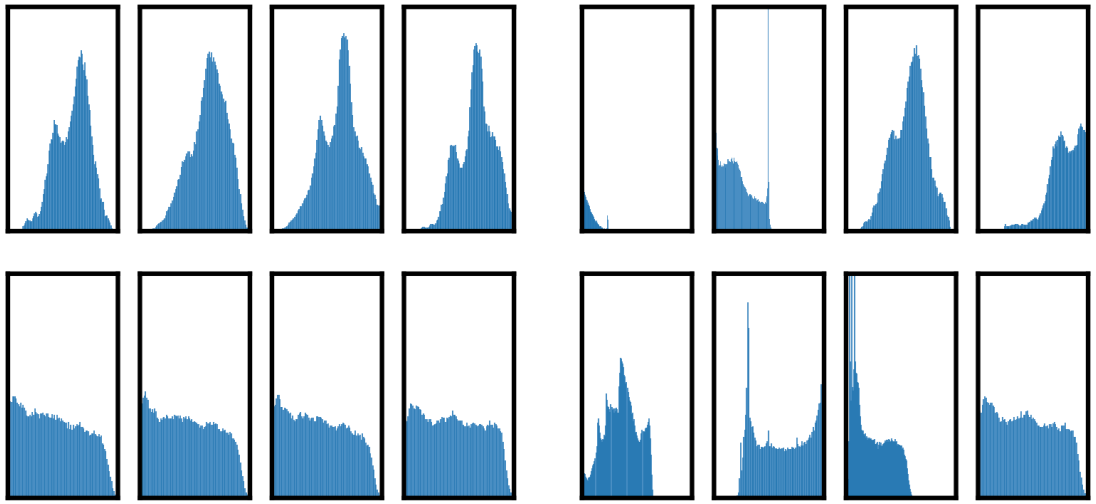
(f) Rotate $\{90°, 180°, 270°\}$ (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

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)

Which augmentations bring more value?



Color Distortion



(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.

← Color histograms suffice to distinguish images. Use DA to fix this.

SSL benefits from **stronger** DA ↓

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.

Performance

Method	Architecture	Param (M)	Top 1	Top 5
Methods using ResNet-50:				
Local Agg.	ResNet-50	24	60.2	-
MoCo	ResNet-50	24	60.6	-
PIRL	ResNet-50	24	63.6	-
CPC v2	ResNet-50	24	63.8	85.3
SimCLR (ours)	ResNet-50	24	**69.3**	**89.0**
Methods using other architectures:				
Rotation	RevNet-50 (4×)	86	55.4	-
BigBiGAN	RevNet-50 (4×)	86	61.3	81.9
AMDIM	Custom-ResNet	626	68.1	-
CMC	ResNet-50 (2×)	188	68.4	88.2
MoCo	ResNet-50 (4×)	375	68.6	-
CPC v2	ResNet-161 (∗)	305	71.5	90.1
SimCLR (ours)	ResNet-50 (2×)	94	74.2	92.0
SimCLR (ours)	ResNet-50 (4×)	375	**76.5**	**93.2**

Table 6. ImageNet accuracies of linear classifiers trained on representations learned with different self-supervised methods.

Method	Architecture	Label fraction	
		1%	10%
		Top 5	
Supervised baseline	ResNet-50	48.4	80.4
Methods using other label-propagation:			
Pseudo-label	ResNet-50	51.6	82.4
VAT+Entropy Min.	ResNet-50	47.0	83.4
UDA (w. RandAug)	ResNet-50	-	88.5
FixMatch (w. RandAug)	ResNet-50	-	89.1
S4L (Rot+VAT+En. M.)	ResNet-50 (4×)	-	91.2
Methods using representation learning only:			
InstDisc	ResNet-50	39.2	77.4
BigBiGAN	RevNet-50 (4×)	55.2	78.8
PIRL	ResNet-50	57.2	83.8
CPC v2	ResNet-161(∗)	77.9	91.2
SimCLR (ours)	ResNet-50	75.5	87.8
SimCLR (ours)	ResNet-50 (2×)	83.0	91.2
SimCLR (ours)	ResNet-50 (4×)	**85.8**	**92.6**

Table 7. ImageNet accuracy of models trained with few labels.

Transfer Learning

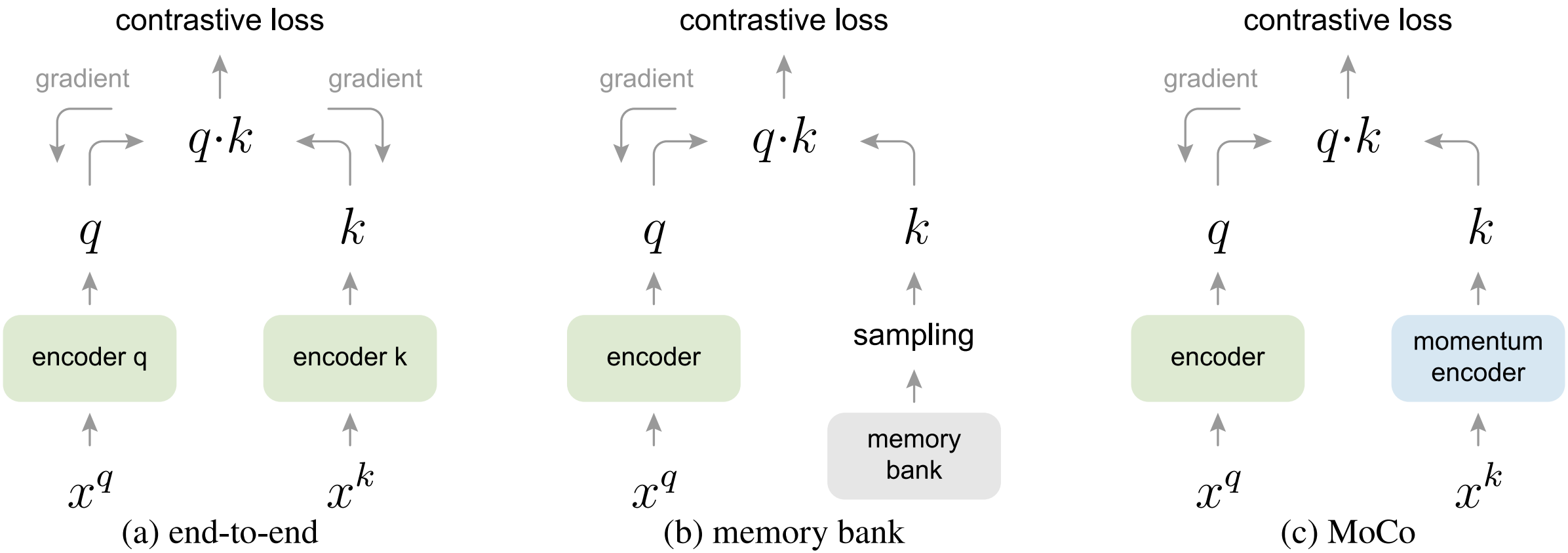
	Food	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Linear evaluation:												
SimCLR (ours)	**76.9**	**95.3**	80.2	48.4	**65.9**	60.0	61.2	**84.2**	**78.9**	89.2	**93.9**	**95.0**
Supervised	75.2	**95.7**	**81.2**	**56.4**	64.9	**68.8**	**63.8**	83.8	**78.7**	**92.3**	**94.1**	94.2
Fine-tuned:												
SimCLR (ours)	**89.4**	**98.6**	**89.0**	**78.2**	**68.1**	**92.1**	**87.0**	**86.6**	**77.8**	92.1	**94.1**	97.6
Supervised	88.7	98.3	**88.7**	**77.8**	67.0	91.4	**88.0**	86.5	**78.8**	**93.2**	**94.2**	**98.0**
Random init	88.3	96.0	81.9	**77.0**	53.7	91.3	84.8	69.4	64.1	82.7	72.5	92.5

Table 8. Comparison of transfer learning performance of our self-supervised approach with supervised baselines across 12 natural image classification datasets, for ResNet-50 $(4\times)$ models pretrained on ImageNet. Results not significantly worse than the best ($p > 0.05$, permutation test) are shown in bold. See Appendix B.8 for experimental details and results with standard ResNet-50.

MoCo

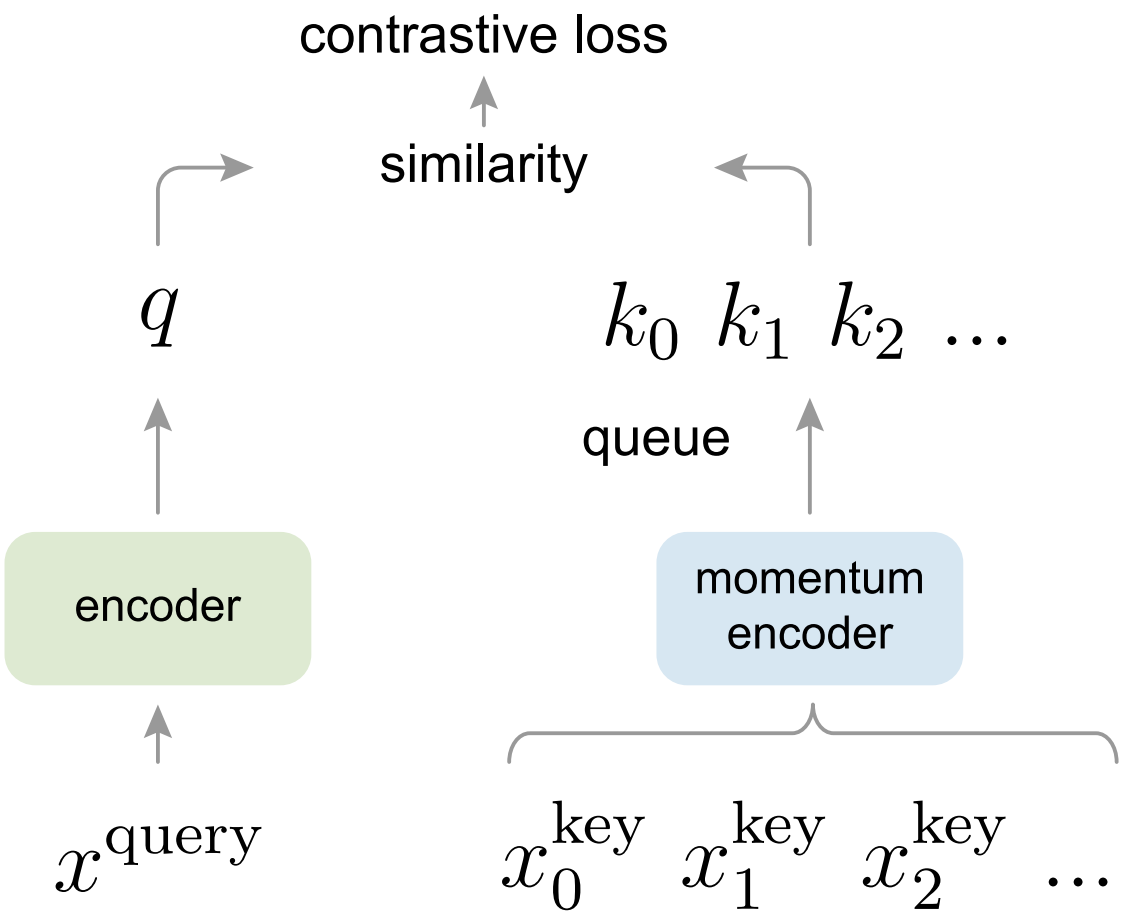
Momentum Contrast for Unsupervised Visual Representation Learning (He et al., 2020)

Asymmetric approach: not as limited by memory, better coverage of negative samples



MoCo

Momentum Contrast for Unsupervised Visual Representation Learning (He et al., 2020)



Another variation of contrastive loss:

$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i / \tau)}$$

Updates are delayed with momentum coefficient m:

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q$$

Negatives are sampled from a dictionary implemented as queue of previous batches

MoCo

MoCo design allows **rich** and **fresh** set of negatives

momentum coefficient favours slower updates:

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

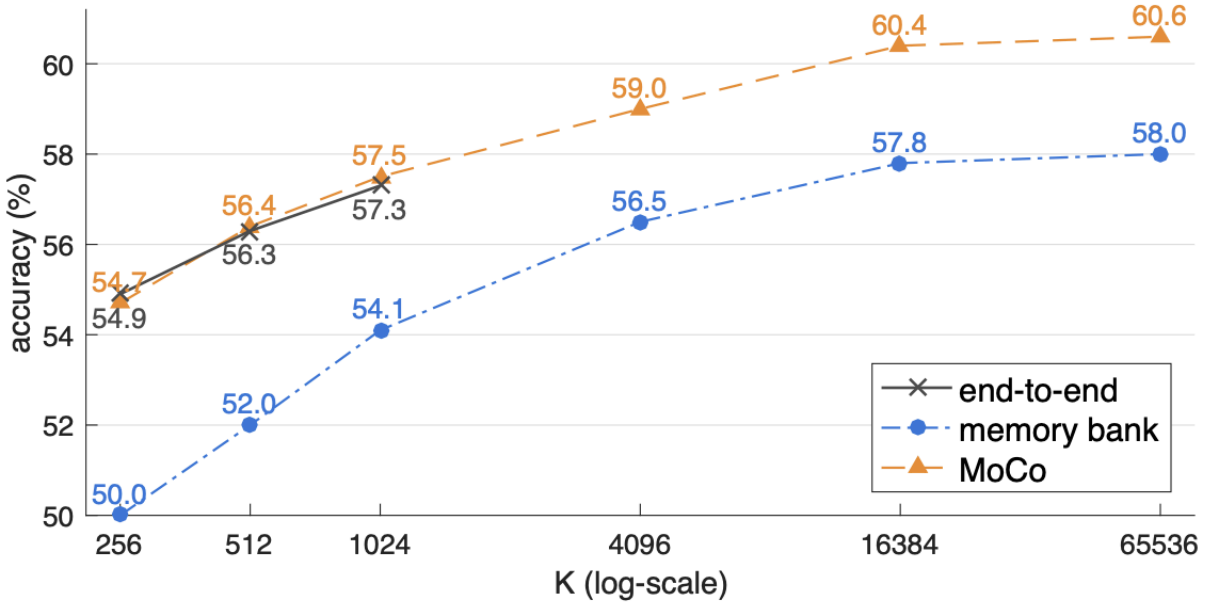
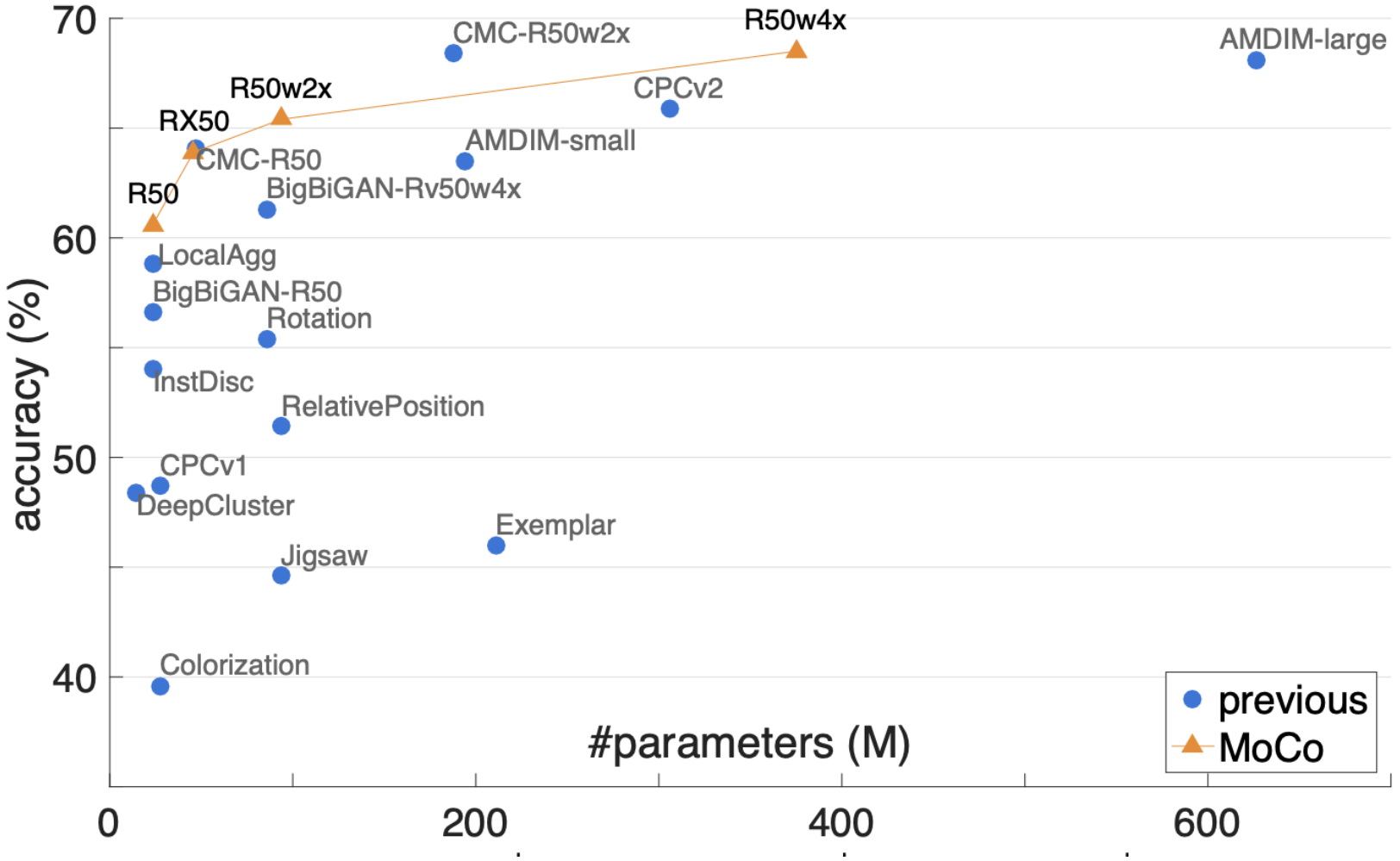


Figure 3. **Comparison of three contrastive loss mechanisms** under the ImageNet linear classification protocol. We adopt the same pretext task (Sec. 3.3) and only vary the contrastive loss mechanism (Figure 2). The number of negatives is K in memory bank and MoCo, and is $K-1$ in end-to-end (offset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

Linear Probing



Detection/Segmentation tasks

MoCo closed the gap with supervised on many vision tasks

pre-train	AP_{50}	AP	AP_{75}
random init.	64.4	37.9	38.6
super. IN-1M	81.4	54.0	59.1
MoCo IN-1M	81.1 (−0.3)	54.6 (**+0.6**)	59.9 (**+0.8**)
MoCo IG-1B	81.6 (+0.2)	55.5 (**+1.5**)	61.2 (**+2.1**)

(a) Faster R-CNN, R50-**dilated-C5**

pre-train	AP_{50}	AP	AP_{75}
random init.	60.2	33.8	33.1
super. IN-1M	81.3	53.5	58.8
MoCo IN-1M	81.5 (+0.2)	55.9 (**+2.4**)	62.6 (**+3.8**)
MoCo IG-1B	82.2 (**+0.9**)	57.2 (**+3.7**)	63.7 (**+4.9**)

(b) Faster R-CNN, R50-**C4**

Table 2. **Object detection fine-tuned on PASCAL VOC** `trainval07+12`. Evaluation is on `test2007`: AP_{50} (default VOC metric), AP (COCO-style), and AP_{75}, averaged over 5 trials. All are fine-tuned for 24k iterations (~23 epochs). In the brackets are the gaps to the ImageNet supervised pre-training counterpart. In green are the gaps of at least **+0.5** point.

pre-train	***COCO keypoint detection*** AP^{kp}	AP^{kp}_{50}	AP^{kp}_{75}
random init.	65.9	86.5	71.7
super. IN-1M	65.8	86.9	71.9
MoCo IN-1M	66.8 (**+1.0**)	87.4 (**+0.5**)	72.5 (**+0.6**)
MoCo IG-1B	66.9 (**+1.1**)	87.8 (**+0.9**)	73.0 (**+1.1**)

pre-train	***COCO dense pose estimation*** AP^{dp}	AP^{dp}_{50}	AP^{dp}_{75}
random init.	39.4	78.5	35.1
super. IN-1M	48.3	85.6	50.6
MoCo IN-1M	50.1 (**+1.8**)	86.8 (**+1.2**)	53.9 (**+3.3**)
MoCo IG-1B	50.6 (**+2.3**)	87.0 (**+1.4**)	54.3 (**+3.7**)

pre-train	***LVIS v0.5 instance segmentation*** AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
random init.	22.5	34.8	23.8
super. IN-1M†	24.4	37.8	25.8
MoCo IN-1M	24.1 (−0.3)	37.4 (−0.4)	25.5 (−0.3)
MoCo IG-1B	24.9 (**+0.5**)	38.2 (+0.4)	26.4 (**+0.6**)

pre-train	***Cityscapes instance seg.*** AP^{mk}	AP^{mk}_{50}	***Semantic seg.*** (mIoU) **Cityscapes**	**VOC**
random init.	25.4	51.1	65.3	39.5
super. IN-1M	32.9	59.6	74.6	74.4
MoCo IN-1M	32.3 (−0.6)	59.3 (−0.3)	75.3 (**+0.7**)	72.5 (−1.9)
MoCo IG-1B	32.9 (0.0)	60.3 (**+0.7**)	75.5 (**+0.9**)	73.6 (−0.8)

Table 6. **MoCo *vs*. ImageNet supervised pre-training, fine-tuned on various tasks**. For each task, the same architecture and schedule are used for all entries (see appendix). In the brackets are the gaps to the ImageNet supervised pre-training counterpart. In green are the gaps of at least **+0.5** point.

03

Self-Distillation

Bootstrap your own latent

BYOL (Grill et al., 2020) needs no negative examples to avoid trivial solution

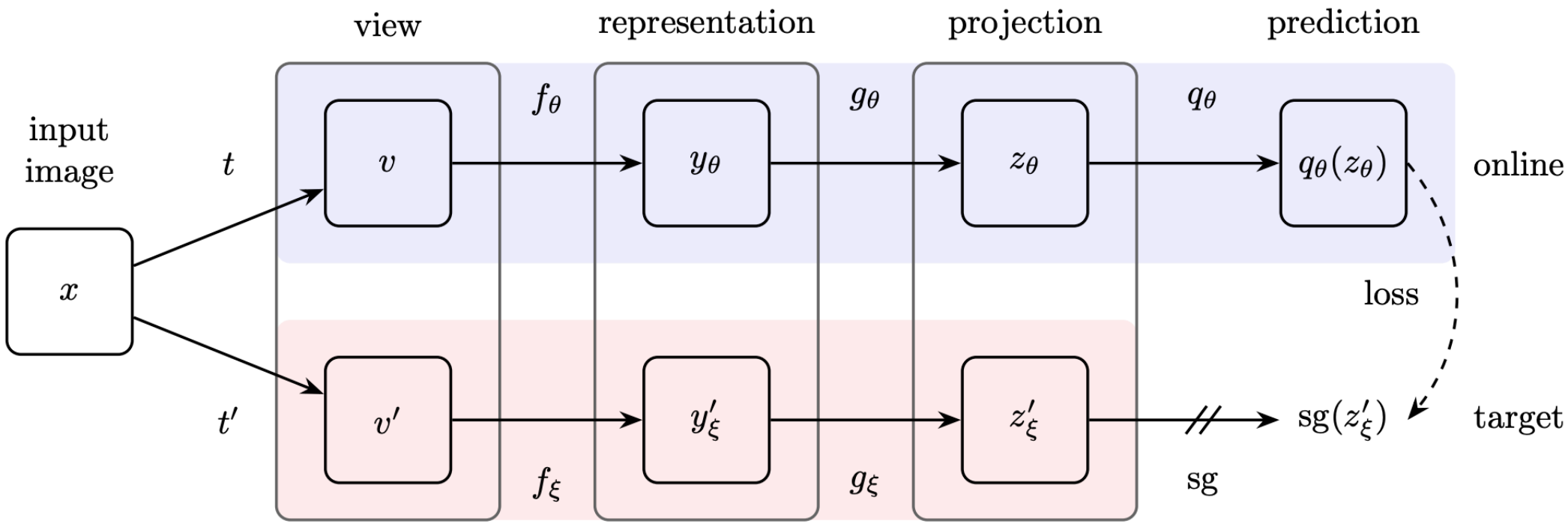
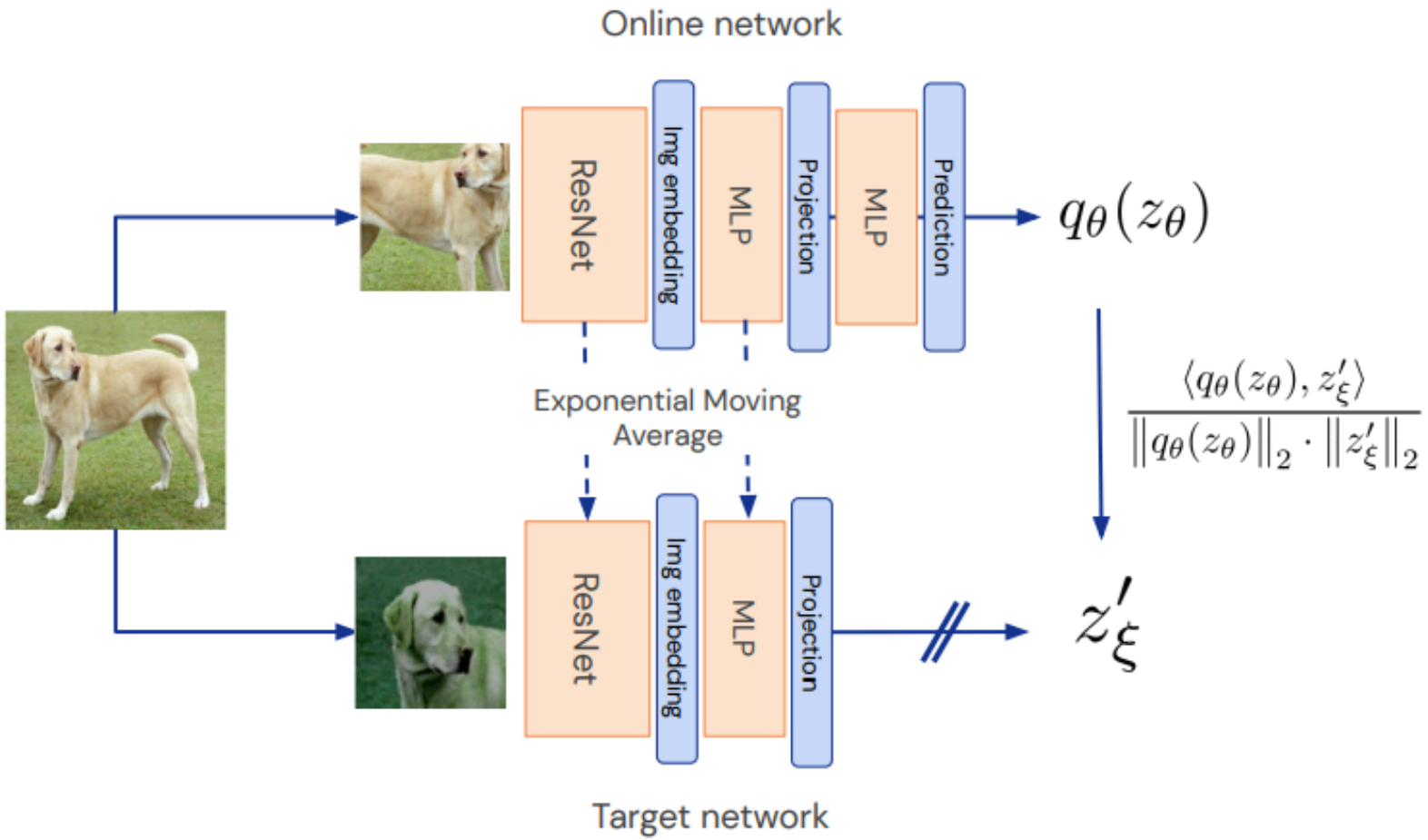


Figure 2: BYOL's architecture. BYOL minimizes a similarity loss between $q_\theta(z_\theta)$ and $\mathrm{sg}(z'_\xi)$, where θ are the trained weights, ξ are an exponential moving average of θ and sg means stop-gradient. At the end of training, everything but f_θ is discarded, and y_θ is used as the image representation.

BYOL


Online network
ResNet
Img embedding
MLP
Projection
MLP
Prediction
$q_\theta(z_\theta)$
Exponential Moving Average
$\frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\|q_\theta(z_\theta)\|_2 \cdot \|z'_\xi\|_2}$
ResNet
Img embedding
MLP
Projection
z'_ξ
Target network

BYOL

Loss is MSE between normalized predictions $\bar{q}_{\theta(z_\theta)}$ and target projections $\bar{z}'_\xi$

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

$\mathcal{L}_{\theta,\xi}^{\mathrm{BYOL}} = \mathcal{L}_{\theta,\xi} + \tilde{\mathcal{L}}_{\theta,\xi}$, where $\tilde{\mathcal{L}}_{\theta,\xi}$ has swapped input

BYOL

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

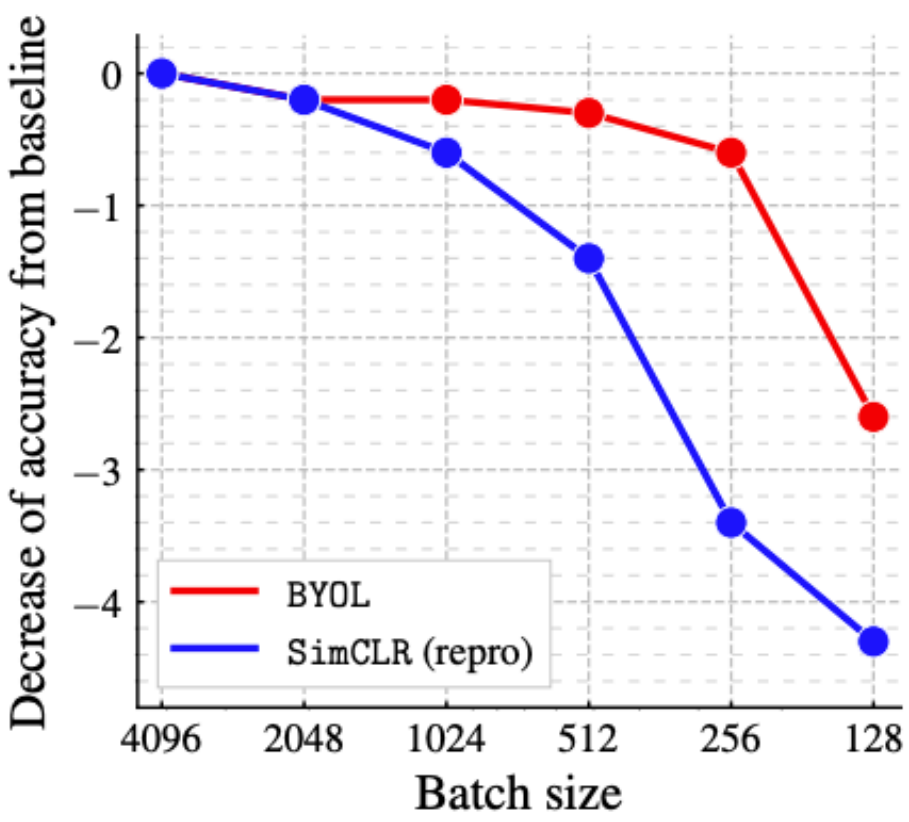
Update online network θ at each training step, use EMA updates for target network ξ:

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

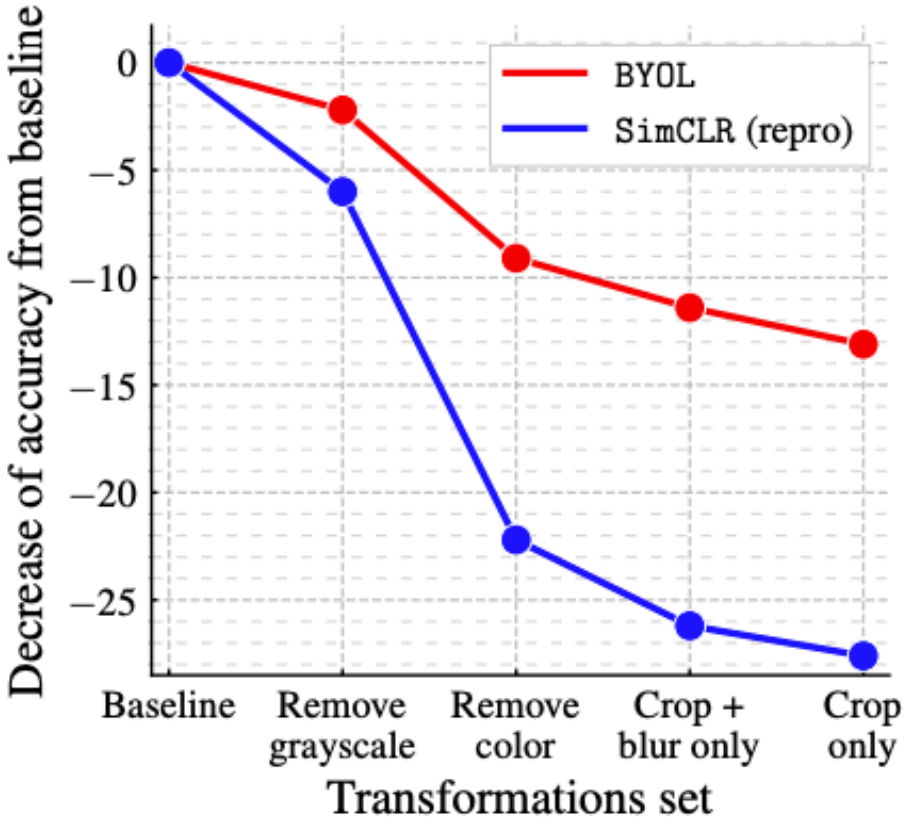
Intuitions behind absense of collapse:

- ξ updates are **not** in the direction of $\nabla_\xi \mathcal{L}^{\text{BYOL}}_{\theta,\xi}$
- Collapsed constant solutions are unstable due to variance induced by asymmetric design / training dynamics

BYOL



(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.

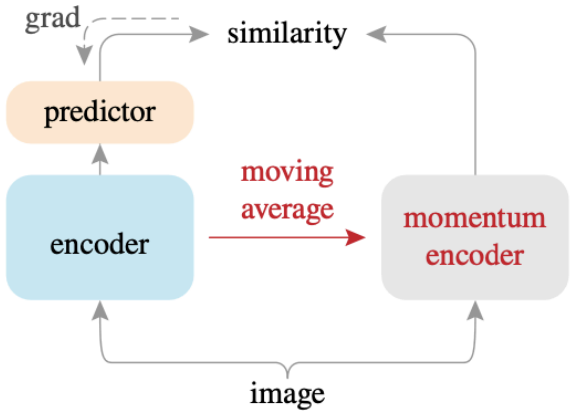
Simple Siamese Representation Learning

SimSiam (Chen & He, 2021):
no negatives, no EMA — no problem!

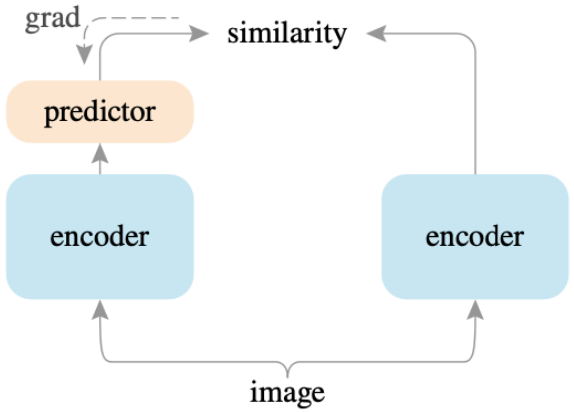
$$\mathcal{D}(p_1, z_2) = -\frac{p_1}{\|p_1\|_2} \frac{z_1}{\|z_2\|_2}$$

equivalent to MSE

symmetrized similar to BYOL

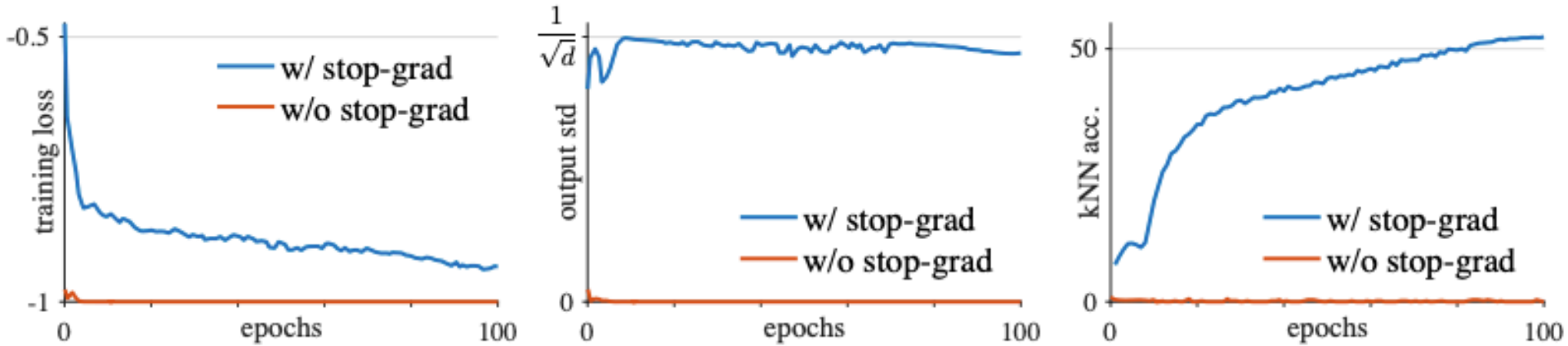


BYOL



SimSiam

SimSiam



	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).

SimSiam

	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	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).

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

SimSiam

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^{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 [32] 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.

SimSiam is secretly EM?

$\mathcal{F}$ — network with parameters θ, τ — augmentation, x — image

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

Similar to k-means clustering: θ — "centroids", η_x — assignment vector for x

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

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

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

SimSiam is secretly EM?

$\mathcal{F}$ — network with parameters θ, τ — augmentation, x — image

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

Similar to k-means clustering: θ — "centroids", η_x — assignment vector for x

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

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

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

SimSiam is secretly EM?

$\mathcal{F}$ — network with parameters θ, τ — augmentation, x — image

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

Similar to k-means clustering: θ — "centroids", η_x — assignment vector for x

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

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

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

Instead of $\eta^t \leftarrow \mathbb{E}_\tau(\mathcal{F}_{\theta^t}(\tau(x)))$

One step: $\eta^t \leftarrow \mathcal{F}_{\theta^t}(\tau'(x))$

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

SimSiam is secretly EM?

$\mathcal{F}$ — network with parameters θ, τ — augmentation, x — image

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

Similar to k-means clustering: θ — "centroids", η_x — assignment vector for x

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

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

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

Instead of $\eta^t \leftarrow \mathbb{E}_\tau(\mathcal{F}_{\theta^t}(\tau(x)))$

One step: $\eta^t \leftarrow \mathcal{F}_{\theta^t}(\tau'(x))$

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

Predictor $h^*(z_1) = \mathbb{E}_{z(z_2)} = \mathbb{E}_\tau[f(\tau(x))]$

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

SimSiam – EM connection (concise)

Setup:

encoder f_θ, predictior q_φ

embeddings $z_i := \mathrm{norm}(f_\theta(x_i)) \in S^{d-1}$, predicted means $\mu_i := \mathrm{norm}\Big(q_\varphi(f_\theta(x_i))\Big) \in S^{d-1}$

probabilistic model $p_{\theta,\varphi}(y_i \mid \mu_i, \kappa) = \mathrm{vMF}(y_i \mid \mu_i, \kappa)$

likelihood $\log p_{\theta,\varphi}(y_i \mid \mu_i, \kappa) = \kappa y_i^\top \mu_i - \log Z(\kappa)$

$$\kappa(y_1^\top \mu_1 + y_2^\top \mu_2) - 2\log Z(\kappa)$$

E-step: compute targets $y_1 = \mathrm{sg}(z_2)$, $y_2 = \mathrm{sg}(z_1)$

M-step: $\max_{\theta,\varphi} \cos\Big(q_\varphi(f_\theta(x_1)), y_1\Big) + \cos\Big(q_\varphi(f_\theta(x_2)), y_2\Big)$

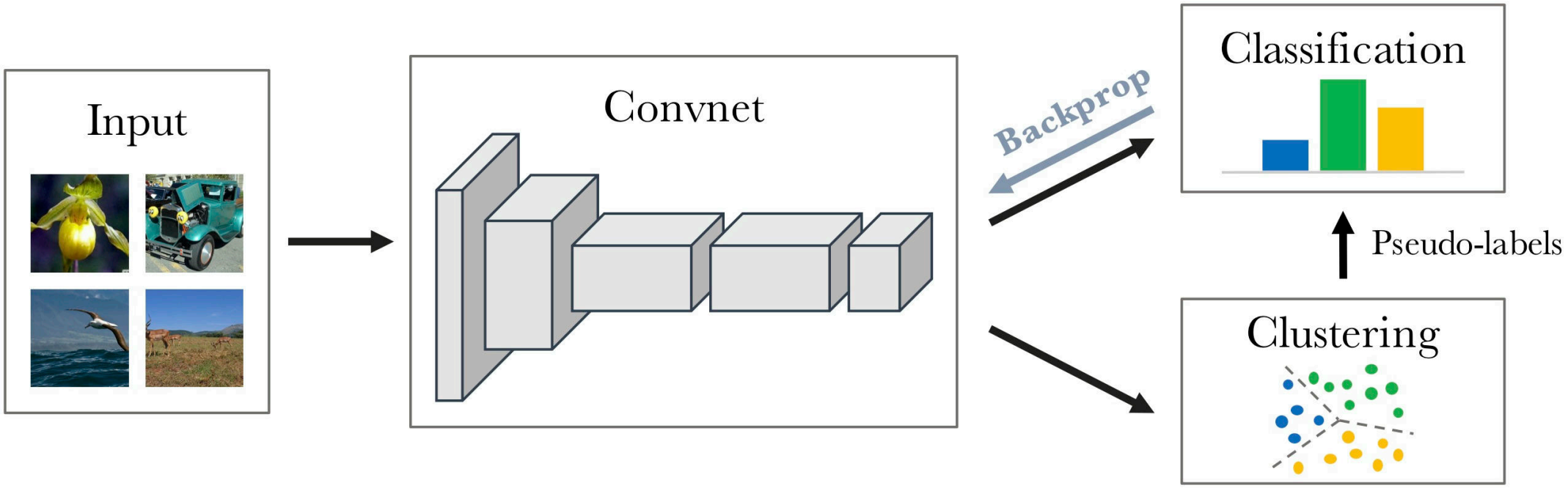
04

Clustering

Deep Cluster (Recap)

Just iteratively cluster features to get pseudo-labels for classification:

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



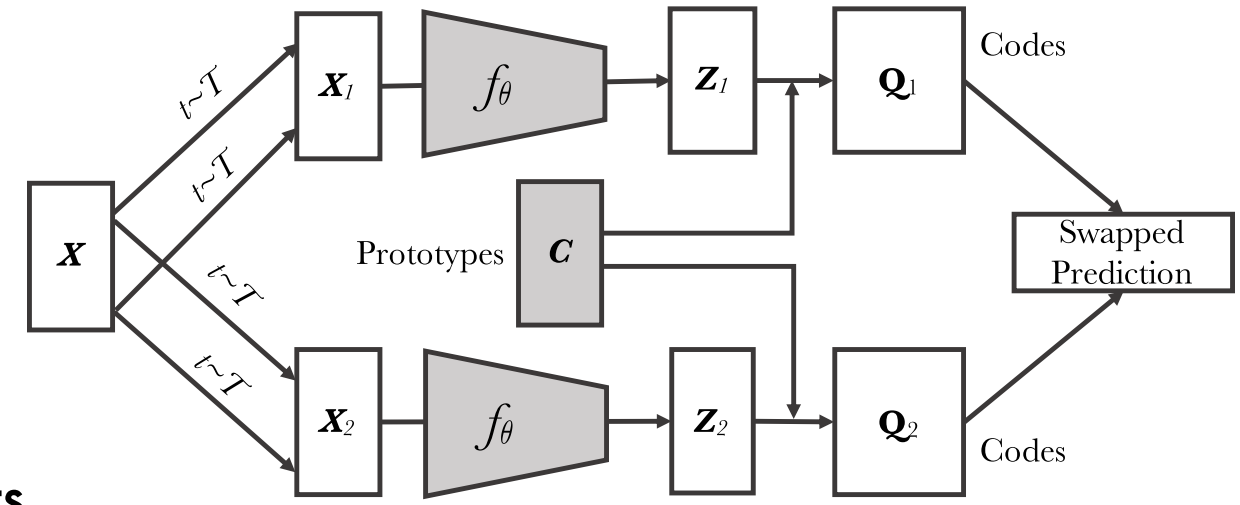
(Caron et al., 2018)

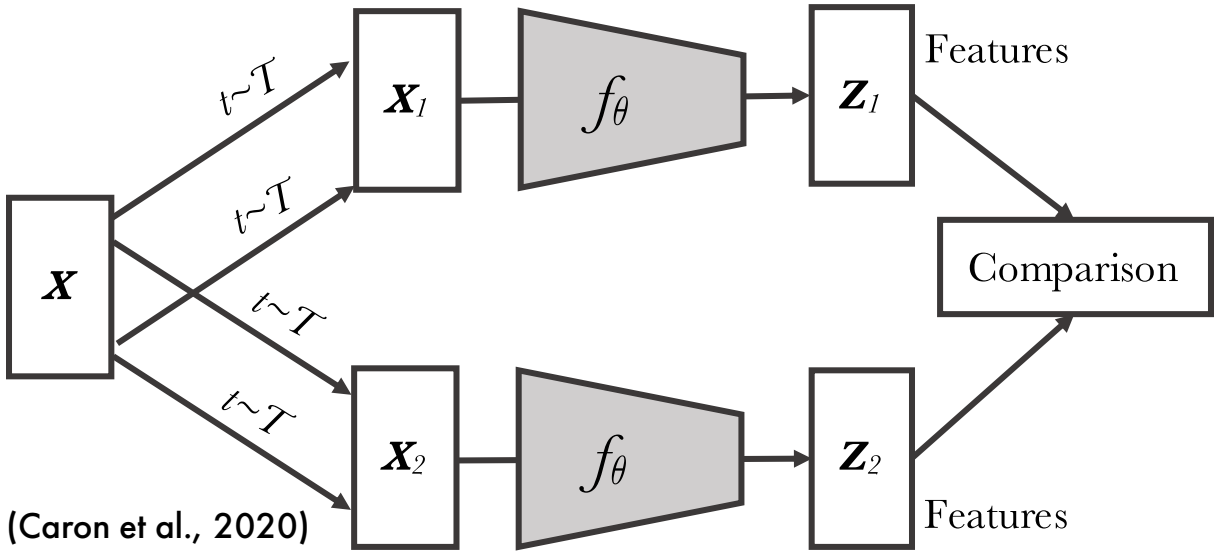
Clustering takes **third** of epoch time!

Swapping Assignments between Views

SwAV — Contrastive "DeepCluster"

- contrastive signal via swapping assignments
- learnable prototypes
- online clustering





(Caron et al., 2020)

Online Clustering

Map $Z = [z_1, ..., z_B]$ to $C = [c_1, ..., c_K]$ via codes matrix $Q = [q_1, ..., q_B]$

Similarity between clusters and representations $C^T Z$

Learn to equally partition codes in batch with $H(Q) = -\sum_{ij} Q_{ij} \log Q_{ij}$:

$$\max_{Q \in \mathcal{Q}} \operatorname{Tr}\, Q^\top C^T Z + \varepsilon H(Q)$$

Doubly stochastic matrices with positive entries

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

Enforce each prototypes picked to $\frac{B}{K}$ times on average

Sinkhorn-Knopp algorithm (iteratively normalize rows/columns):

$Q^* = \operatorname{Diag}(u) \exp\left(\frac{C^T Z}{\varepsilon}\right) \operatorname{Diag}(v)$, where u, v - renormalization vectors

SwAV

Once **soft** Cluster Assignment is done, we have codes Q

Now contrastive loss for image positive pair x_t, x_s:

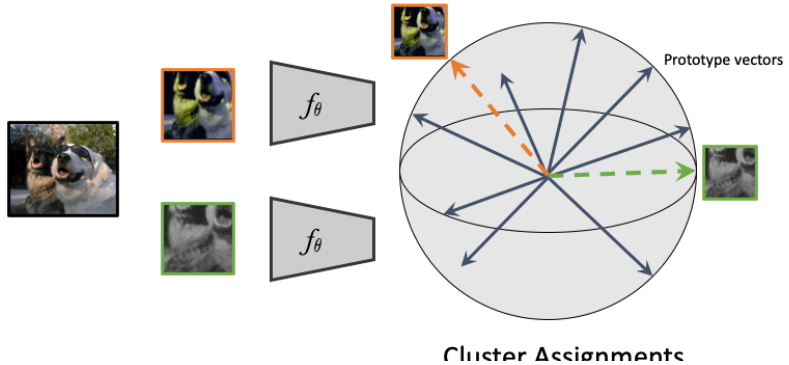
$$l(z_t, q_s) = -\sum_k q_s^{(k)} \log p_t^{(k)}$$

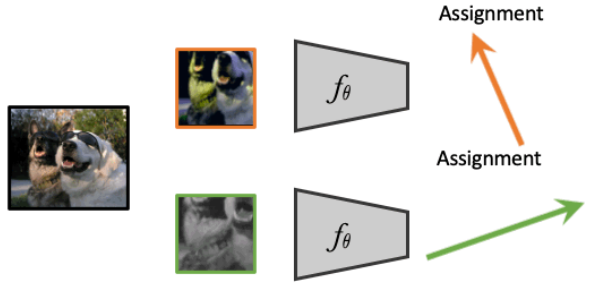
$$p_t^{(k)} = \frac{\exp\left(\frac{1}{\tau} z_t^\top c_k\right)}{\sum_{k'} \exp\left(\frac{1}{\tau} z_t^\top c_{k'}\right)}$$

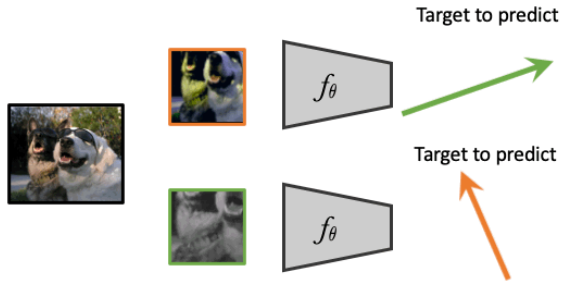
Symmetric loss $L(z_t, z_s) = l(z_t, q_s) + l(z_s, q_t)$

NB SwAV allows multi-crop

$$L(z_{t_1}, z_{t_2}, \ldots, z_{t_{V+2}} = \sum_{(i \in \{1,2\})} \sum_{v=1}^{V+2} \mathbf{1}_{v \neq i} l\left(z_{t_v}, q_{t_i}\right)$$







SwAV

Table 3: **Training in small batch setting.** Top-1 accuracy on ImageNet with a linear classifier trained on top of frozen features from a ResNet-50. All methods are trained with a batch size of 256. We also report the number of stored features, the type of cropping used and the number of epochs.

Method	Mom. Encoder	Stored Features	`multi-crop`	epoch	batch	Top-1
SimCLR		0	2×224	200	256	61.9
MoCov2	✓	65,536	2×224	200	256	67.5
MoCov2	✓	65,536	2×224	800	256	71.1
SwAV		3,840	$2\times160 + 4\times96$	200	256	72.0
SwAV		3,840	$2\times224 + 6\times96$	200	256	72.7
SwAV		3,840	$2\times224 + 6\times96$	400	256	**74.3**

SwAV

Method	Arch.	Param.	Top1
Supervised	R50	24	76.5
Colorization [65]	R50	24	39.6
Jigsaw [46]	R50	24	45.7
NPID [58]	R50	24	54.0
BigBiGAN [15]	R50	24	56.6
LA [68]	R50	24	58.8
NPID++ [44]	R50	24	59.0
MoCo [24]	R50	24	60.6
SeLa [2]	R50	24	61.5
PIRL [44]	R50	24	63.6
CPC v2 [28]	R50	24	63.8
PCL [37]	R50	24	65.9
SimCLR [10]	R50	24	70.0
MoCov2 [11]	R50	24	71.1
SwAV	R50	24	**75.3**

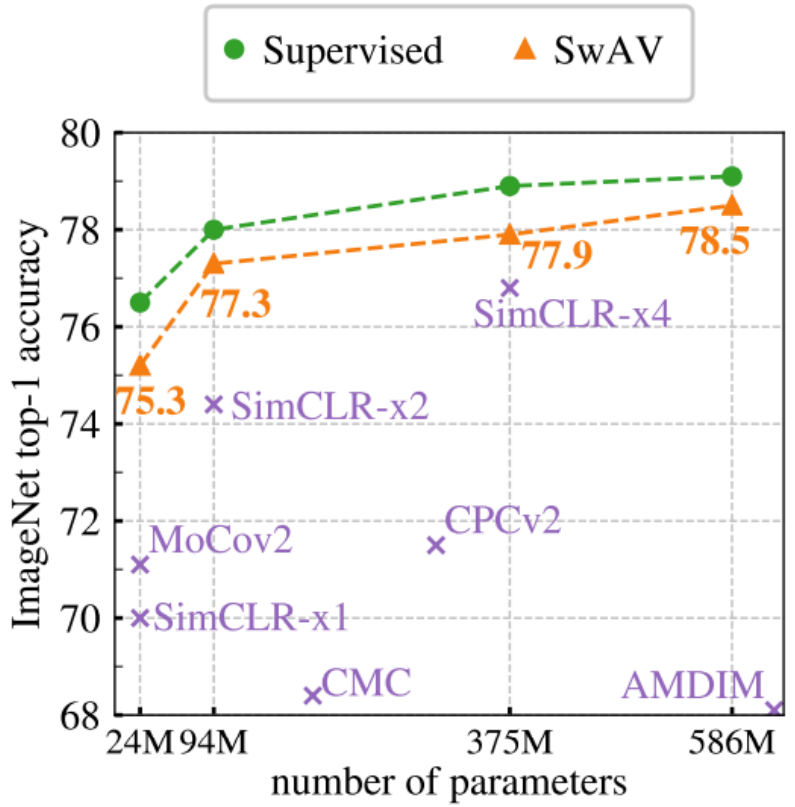
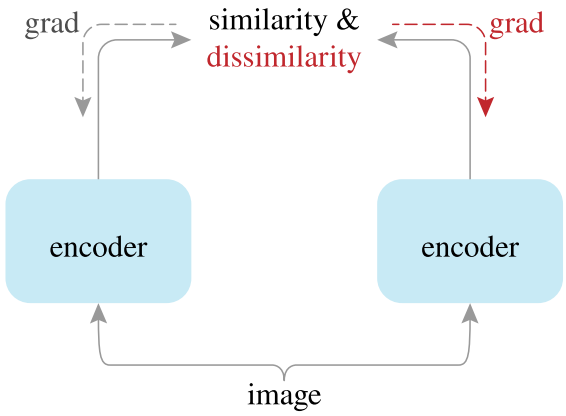


Figure 2: **Linear classification on ImageNet.** Top-1 accuracy for linear models trained on frozen features from different self-supervised methods. **(left)** Performance with a standard ResNet-50. **(right)** Performance as we multiply the width of a ResNet-50 by a factor $\times 2$, $\times 4$, and $\times 5$.

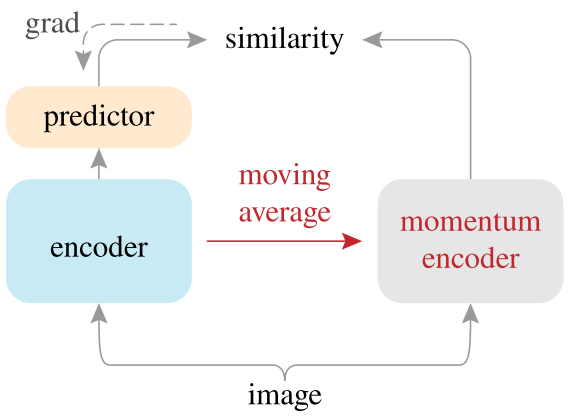
SwAV

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

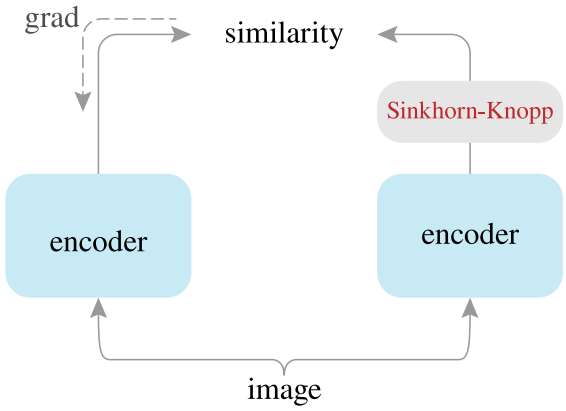
Recap



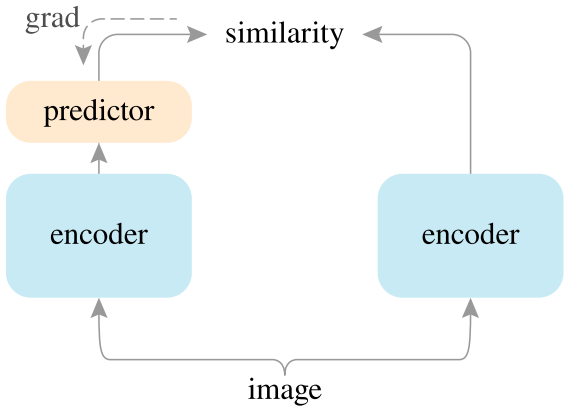
SimCLR



BYOL



SwAV



SimSiam

Collapse in Contrastive Learning

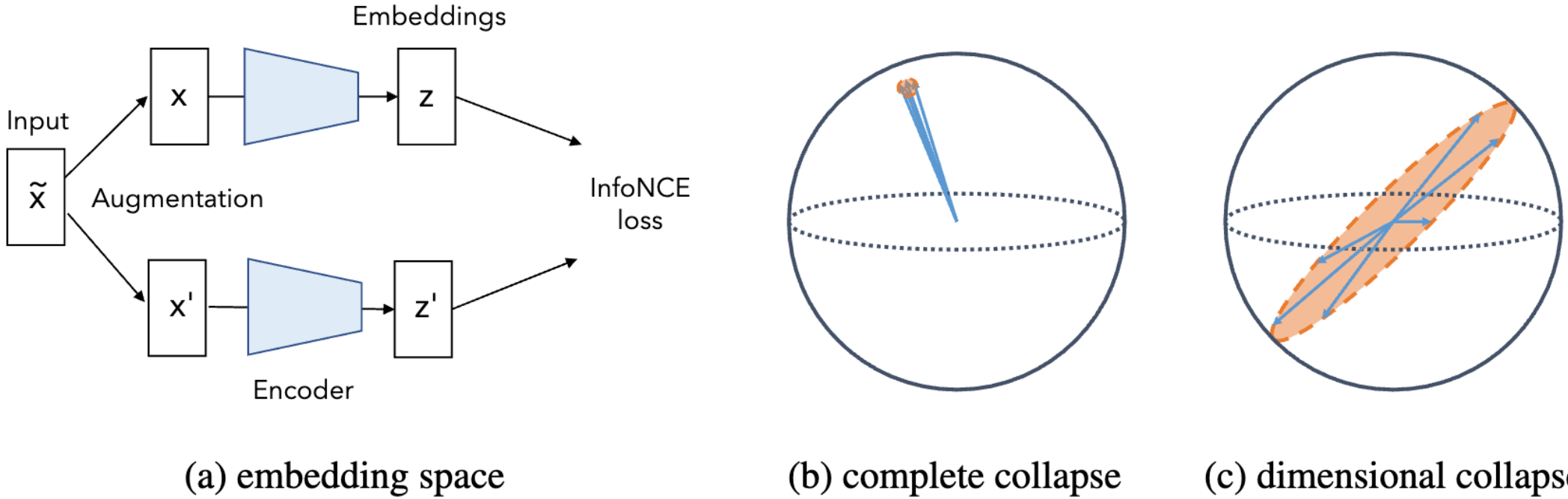


Figure 1: Illustration of the collapsing problem. For complete collapse, the embedding vectors collapse to same point. For dimensional collapse, the embedding vectors only span a lower dimensional space.

(Jing et al., 2021)

Dimensional Collapse

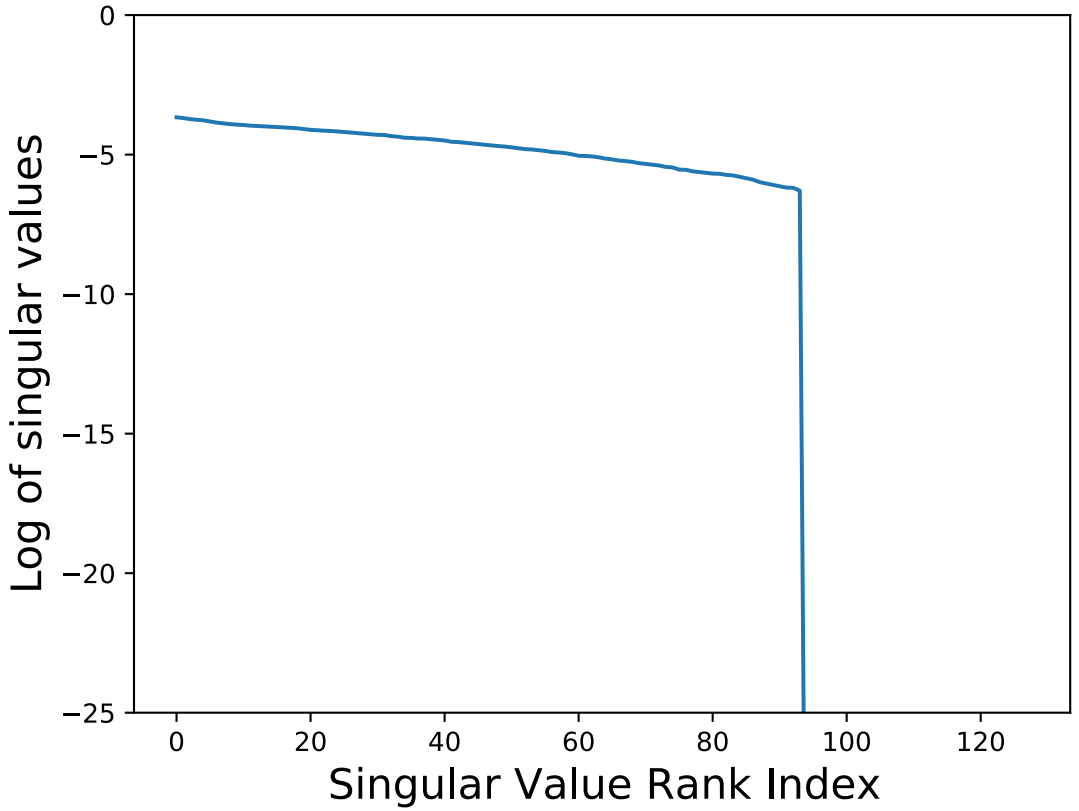
Covariance matrix of SimCLR embeddings

$$C = \frac{1}{N}\sum_{i=1}^{N}(z_i - \bar{z})(z_i - \bar{z})^T,$$

where $\bar{z} = \frac{1}{N}\sum_{i=1}^{N} z_i$

$C = U\Sigma V^\top$

>20 singular values drop to zero

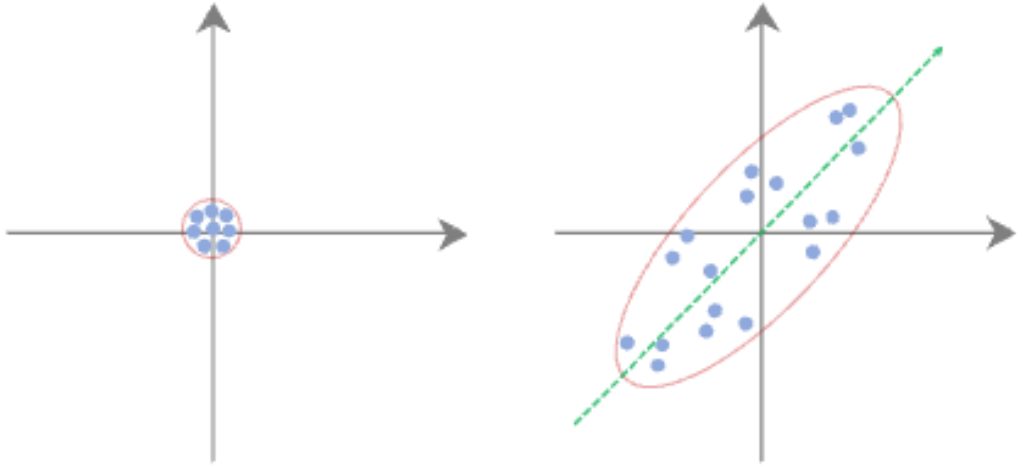


(Jing et al., 2021)

05

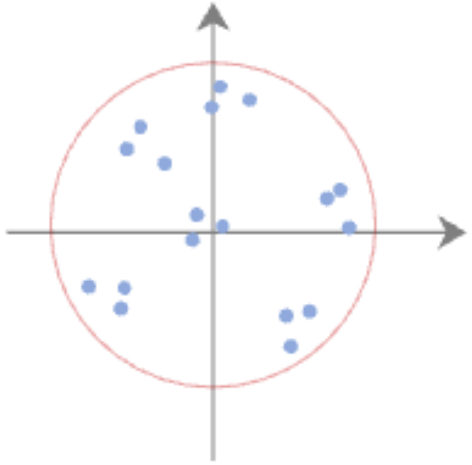
Decorrelation / Whitening

Feature Decorrelation

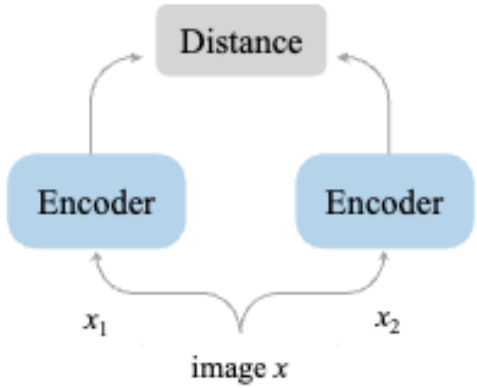


(a) complete collapse

(b) dimensional collapse



(c) decorrelated



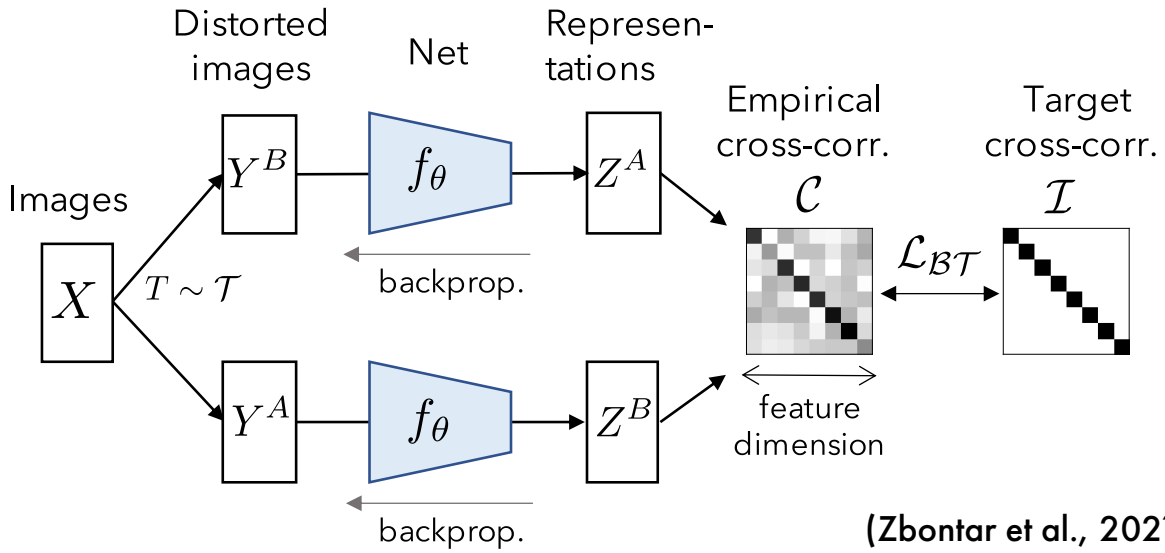
(d) the concise framework

Barlow Twins

Enforce statistically independent components

Cross-correlation matrix of twin embeddings

$$C_{ij} \triangleq \frac{\sum_b Z^A_{bi} Z^B_{bj}}{\sqrt{\sum_b \left(Z^A_{bi}\right)^2} \sqrt{\sum_b \left(Z^B_{bj}\right)^2}},$$



(Zbontar et al., 2021)

Z^A, Z^B — mean centered embedding matrices

for two data views A and B, i, j - index features, b - index of sample in a batch

invariance and redundancy-reduction terms: $\mathcal{L}_{\mathrm{BT}} = \sum_i (1 - C_{ii})^2 + \lambda \sum_i \sum_{j \neq i} C_{ij}^2$

Decorrelating every pair of features maximizes information content preventing collapse

Information Bottleneck Principle

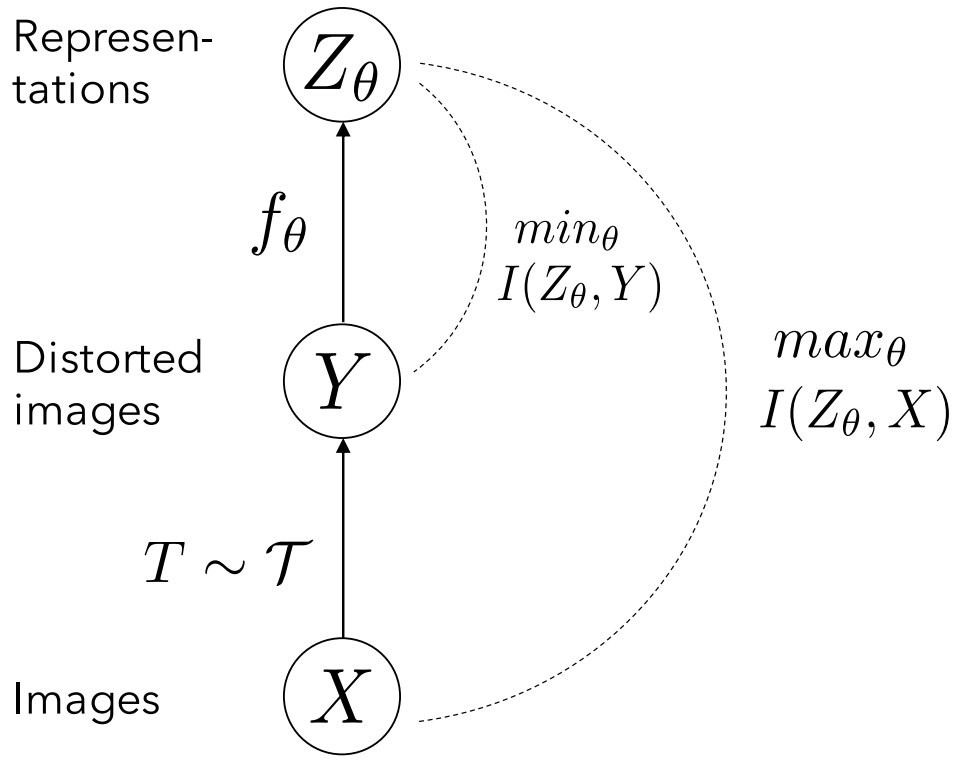
Information Bottleneck Principle applied to SSL

$$\mathrm{IB}_\theta \triangleq I(Z_\theta, Y) - \beta I(Z_\theta, X)$$

- representations informative about input
- representations invariant to distortions

$$\mathrm{IB}_\theta = [H(Z_\theta) - \cancel{H(Z_\theta|Y)}] - \beta[H(Z_\theta) - H(Z_\theta|X)]$$

$$= H(Z_\theta|X) + \frac{1-\beta}{\beta} H(Z_\theta)$$

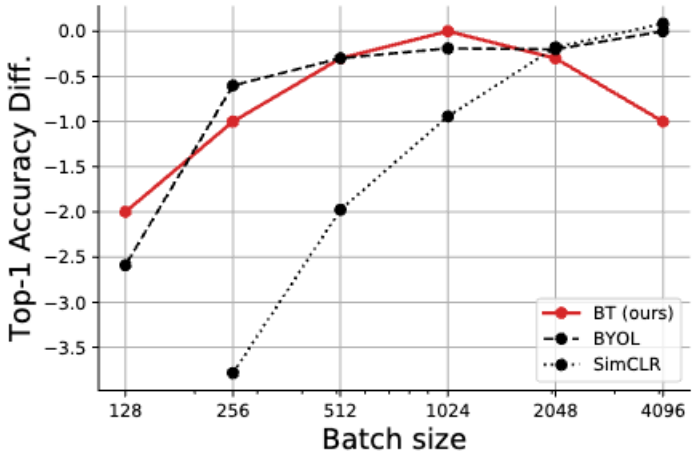


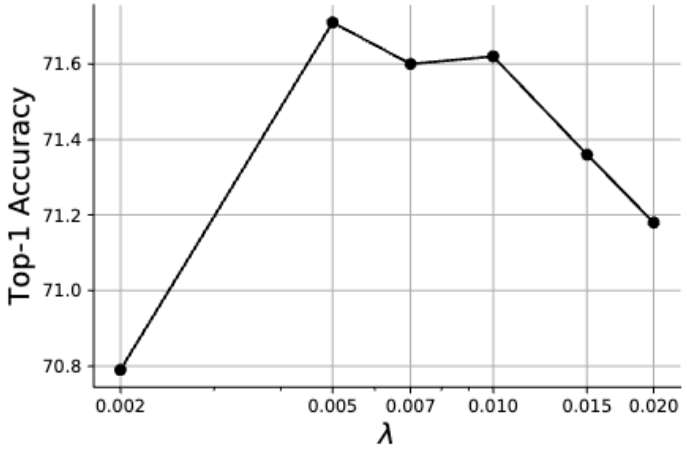
(Zbontar et al., 2021)

Assume Z is Gaussian $\Rightarrow \mathrm{IB}_\theta = \mathbb{E}_X \log|C_{Z_\theta|X}| + \frac{1-\beta}{\beta} \log|C_{Z_\theta}|$

$\beta > 1$, replace covariance with cross-correlation $\Rightarrow \mathcal{L}_{\mathrm{BT}}$

BT Ablations





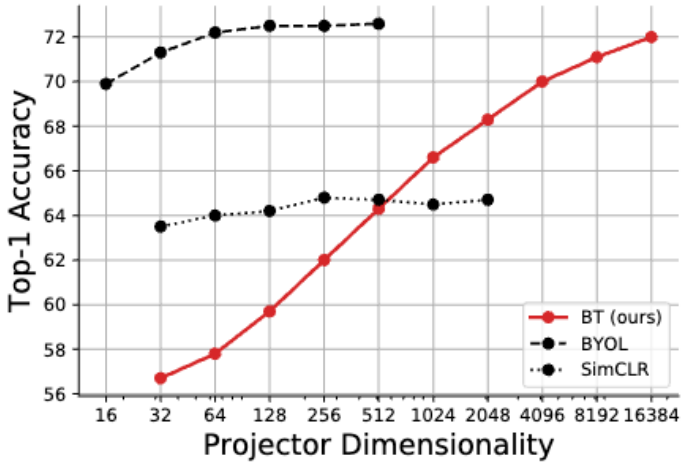


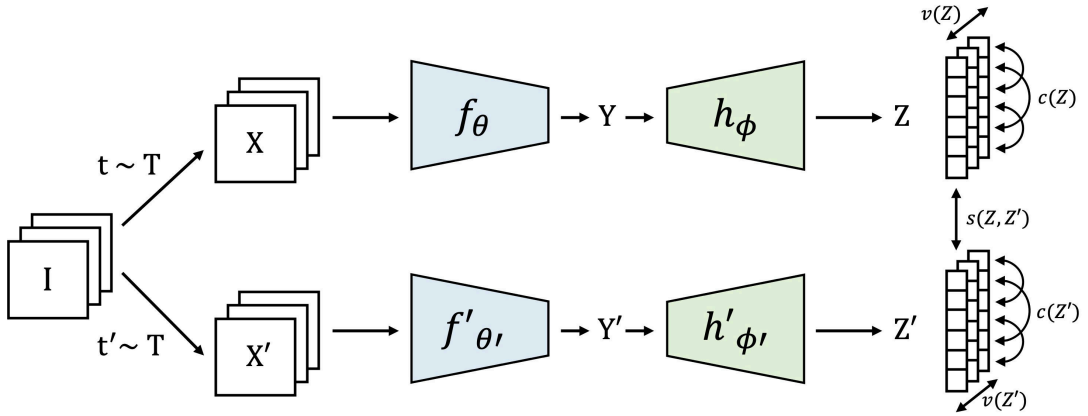
Table 3. **Transfer learning: image classification.** We benchmark learned representations on the image classification task by training linear classifiers on fixed features. We report top-1 accuracy on Places-205 and iNat18 datasets, and classification mAP on VOC07. Top-3 best self-supervised methods are underlined.

Method	Places-205	VOC07	iNat18
Supervised	53.2	87.5	46.7
SimCLR	52.5	85.5	37.2
MoCo-v2	51.8	86.4	38.6
SwAV (w/o multi-crop)	52.8	86.4	39.5
SwAV	56.7	88.9	48.6
BYOL	54.0	86.6	47.6
BARLOW TWINS (ours)	54.1	86.2	46.5

Variance-Invariance-Covariance

VICReg (Bardes et al., 2021):

- Variance: $v(Z) = \frac{1}{d}\sum_j^d \max(0, \gamma - S(z^j, \varepsilon))$
  $S(x, \varepsilon) = \sqrt{\mathrm{Var}(x) + \varepsilon}$
- Covariance: $c(Z) = \frac{1}{d}\sum_{i \neq j} [C(Z)]_{i,j}^2$
- Invariance: $s(Z) = \frac{1}{n}\sum_i \|z_i - z_i'\|_2^2$



invariance to different views + collapse prevention + information content maximization:

$$l(Z, Z') = \lambda s(Z, Z') + \mu[v(Z) + v(Z')] + \nu[c(Z) + c(Z')]$$

Not much difference with other methods, what's up?

Regularization

Table 4: **Effect of incorporating variance and covariance regularization in different methods.** Top-1 ImageNet accuracy with the linear evaluation protocol after 100 pretraining epochs. For all methods, pretraining follows the architecture, the optimization and the data augmentation protocol of the original method using our reimplementation. ME: Momentum Encoder. SG: stop-gradient. PR: predictor. BN: Batch normalization layers after input and inner linear layers in the expander. No Reg: No additional regularization. Var Reg: Variance regularization. Var/Cov Reg: Variance and Covariance regularization. Unmodified original setups are marked by a †.

Method	ME	SG	PR	BN	No Reg	Var Reg	Var/Cov Reg
BYOL	✓	✓	✓	✓	69.3†	70.2	69.5
SimSiam		✓	✓	✓	67.9†	68.1	67.6
SimSiam		✓	✓		35.1	67.3	67.1
SimSiam		✓			collapse	56.8	66.1
VICReg			✓		collapse	56.2	67.3
VICReg			✓	✓	collapse	57.1	68.7
VICReg				✓	collapse	57.5	68.6†
VICReg					collapse	56.5	67.4

Weights and Architecture

Table 5: **Impact of sharing weights or not between branches.** Top-1 accuracy on linear classification with 100 pretraining epochs. The encoder and expander of both branches can share the same architecture and share their weights (SW), share the same architecture with different weights (DW), or have different architectures (DA). The encoders can be ResNet-50, ResNet-101 or ViT-S.

	SW R50	DW R50	DA R50/R101	DA R50/ViT-S
BYOL	69.3	✗	✗	✗
SimCLR	64.4	63.1	63.9	63.5
Barlow Twins	68.7	64.2	65.3	63.9
VICReg	68.6	66.5	68.1	66.2

Preliminaries

Whitening converts $x = (x_1, ..., x_d)^\top$, $\mathbb{E}(x) = \mu = (\mu_1, ..., \mu_d)^\top$, $\mathrm{var}(x) = \Sigma$ into

$$z = (z_1, ..., z_d)^\top = Wx$$

that has unit diagonal "white" covariance $\mathrm{var}(z) = I$, and W — whitening matrix

How to choose W? $\quad W\Sigma W^\top = \mathrm{var}(z) = I \rightarrow W^\top W = \Sigma^{-1}$ W is not uniquely defined!

How to select optimal W? (Kessy et al., 2018)

We consider

- Soft-whitening (Barlow Twins, VICreg)
- Cholesky: $\Sigma = LL^T \rightarrow W_{\mathrm{Chol}} = L^{-1}$
- ZCA: $W^{\mathrm{ZCA}} = \Sigma^{-\frac{1}{2}}$

Table 1: Five natural whitening transformations and their properties.

	Sphering matrix W	Cross-covariance Φ	Cross-correlation Ψ	Rotation matrix Q_1	Rotation matrix Q_2
ZCA	$\Sigma^{-1/2}$	$\Sigma^{1/2}$	$\Sigma^{1/2}V^{-1/2}$	I	A^T
PCA	$\Lambda^{-1/2}U^T$	$\Lambda^{1/2}U^T$	$\Lambda^{1/2}U^TV^{-1/2}$	U^T	U^TA^T
Cholesky	L^T	$L^T\Sigma$	$L^T\Sigma V^{-1/2}$	$L^T\Sigma^{1/2}$	$L^TV^{1/2}P^{1/2}$
ZCA-cor	$P^{-1/2}V^{-1/2}$	$P^{1/2}V^{1/2}$	$P^{1/2}$	A	I
PCA-cor	$\Theta^{-1/2}G^TV^{-1/2}$	$\Theta^{1/2}G^TV^{1/2}$	$\Theta^{1/2}G^T$	G^TA	G^T

Whitening for Self-Supervised Learning

W-MSE (Ermolov et al., 2021): Cholesky decomposition

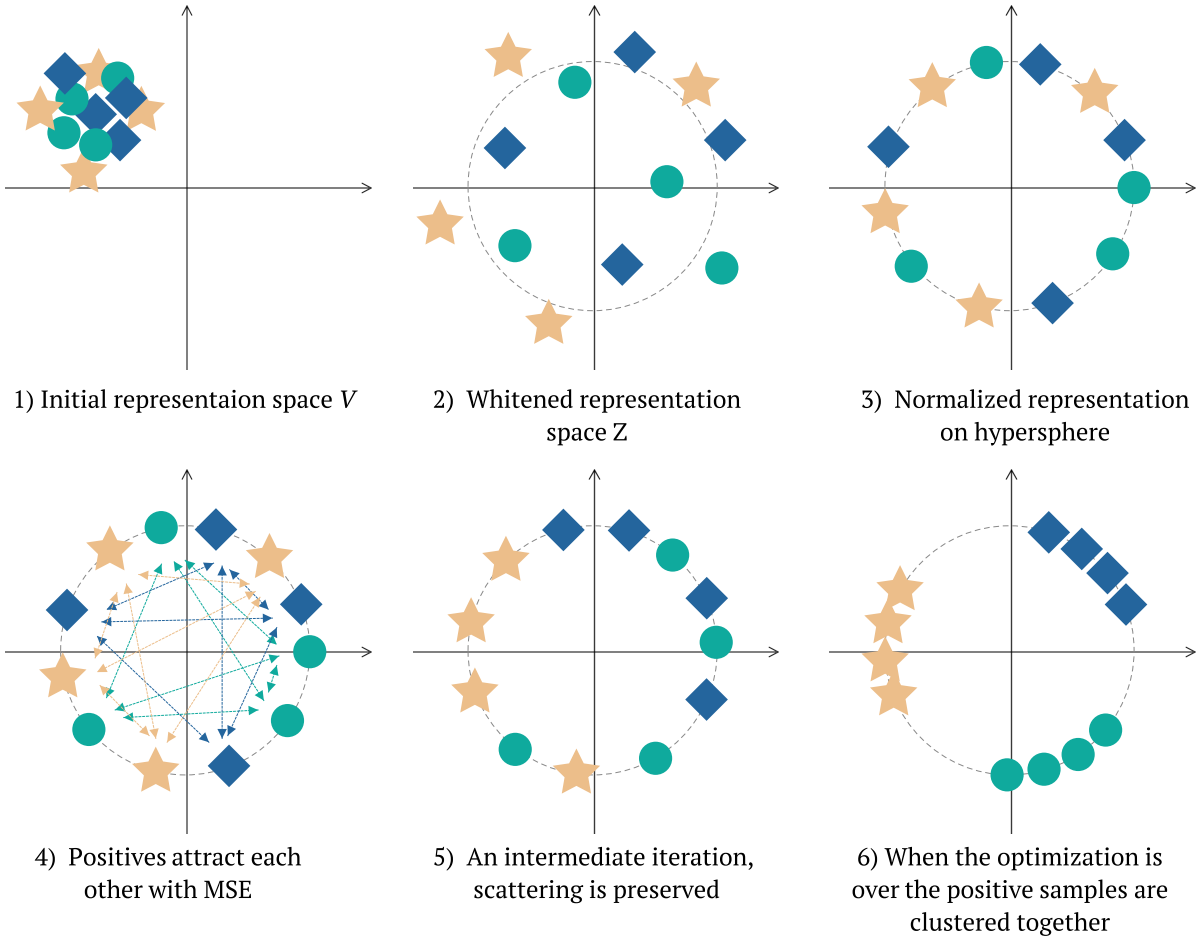
(x_i, x_j) — positive pairs

z_i, z_j — embeddings of positive pair

$$\min_{\theta} \mathbb{E}\left[\text{dist}\left(z_i, z_j\right)\right]$$

$$s.t.\ \text{cov}(z_i, z_i) = \text{cov}\left(z_j, z_j\right) = I$$

dist — cosine similarity



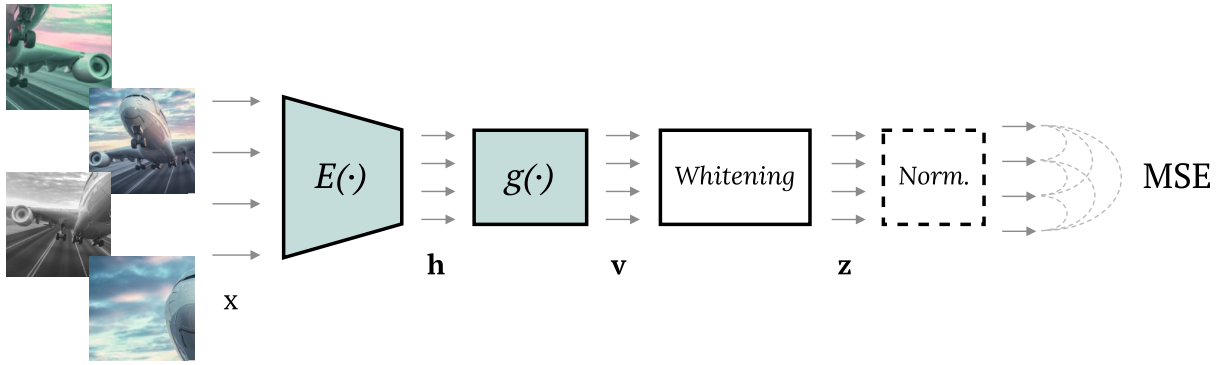
W-MSE

N unique images, d — augmentations,

$K = Nd$ — total batch size

$V = \{v_1, ..., v_K\}$ — embeddings of batch

$\Sigma_V = \frac{1}{K-1} \sum_k (v_k - \mu_V)(v_k - \mu_V)^\top$



W-MSE loss uses reparameterization of v to whitened z:

$$L_{\text{W-MSE}}(V) = \frac{2}{Nd(d-1)} \sum_{\text{pos}} \text{dist}\big(z_i, z_j\big),$$

$z = \text{Whitening}(v) = W_v(v - \mu_v)$ with $W_V^\top W_V = \Sigma_V^{-1}$

Compute Cholesky decomposition $\Sigma_V = LL^T$, take $W_V = L^{-1}$ on sub-batches of V

Complexity $O(k^3 + Mk^2)$ with k embedding dim, M slice size — comparable to forward pass

Decorrelated Batch Normalization

Consise framework to study collapse:

$$\mathcal{L}(\theta) = \mathbb{E}_{x \sim D, t_1, t_2 \sim T} \ \| f_\theta(x_1) - f_\theta(x_2) \|_2^2$$

Collapse patterns (Hua et al., 2021):

Batch Normalization

$X = (x_1, ..., x_B) \in \mathbb{R}^{D \times B}$ — input
$Y = (x_1, ..., x_B) \in \mathbb{R}^{D \times B}$ — output

Batch Normalization: $y_{b,d} = \frac{x_{b,d} - \mu_d}{\sqrt{\sigma_d^2 + \varepsilon}} \gamma_d + \beta_d$

- removes complete collapse

Decorrelated Batch Normalization (DBN): $Y^{[h]} = \mathrm{ZCA}\left(X^{[h]}\right)$ with $\mathrm{ZCA} : Y = Q \Lambda^{-\frac{1}{2}} Q^\top \hat{X}$

- decorrelates covariance of feature groups

06

Bibliography

Bardes, A., Ponce, J., & LeCun, Y. (2021). Vicreg: Variance-invariance-covariance regularization for self-supervised learning. Arxiv Preprint Arxiv:2105.04906.

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. Proceedings of the European Conference on Computer Vision (ECCV), 132–149.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. Advances in Neural Information Processing Systems, 33, 9912–9924.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. International Conference on Machine Learning, 1597–1607.

Chen, X., & He, K. (2021). Exploring simple siamese representation learning. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 15750–15758.

Ermolov, A., Siarohin, A., Sangineto, E., & Sebe, N. (2021). Whitening for self-supervised representation learning. International Conference on Machine Learning, 3015–3024.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., & others. (2020). Bootstrap your own latent-a new approach to self-supervised learning. Advances in Neural Information Processing Systems, 33, 21271–21284.

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9729–9738.

Hua, T., Wang, W., Xue, Z., Ren, S., Wang, Y., & Zhao, H. (2021). On feature decorrelation in self-supervised learning. Proceedings of the IEEE/CVF International Conference on Computer Vision, 9598–9608.

Jing, L., Vincent, P., LeCun, Y., & Tian, Y. (2021). Understanding dimensional collapse in contrastive self-supervised learning. Arxiv Preprint Arxiv:2110.09348.

Kessy, A., Lewin, A., & Strimmer, K. (2018). Optimal whitening and decorrelation. The American Statistician, 72(4), 309–314.

Oord, A. v. d., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. Arxiv Preprint Arxiv:1807.03748.

Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., & Tucker, G. (2019). On variational bounds of mutual information. International Conference on Machine Learning, 5171–5180.

Sridharan, K., & Kakade, S. M. (2008). An information theoretic framework for multi-view learning. COLT, 114, 403–414.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., & Deny, S. (2021). Barlow twins: Self-supervised learning via redundancy reduction. International Conference on Machine Learning, 12310–12320.

Thank you!