ECCV
TEL AVIV 2022
Tutorial

Contrastive Learning of Visual Representations

Ting Chen

Google Research, Brain Team

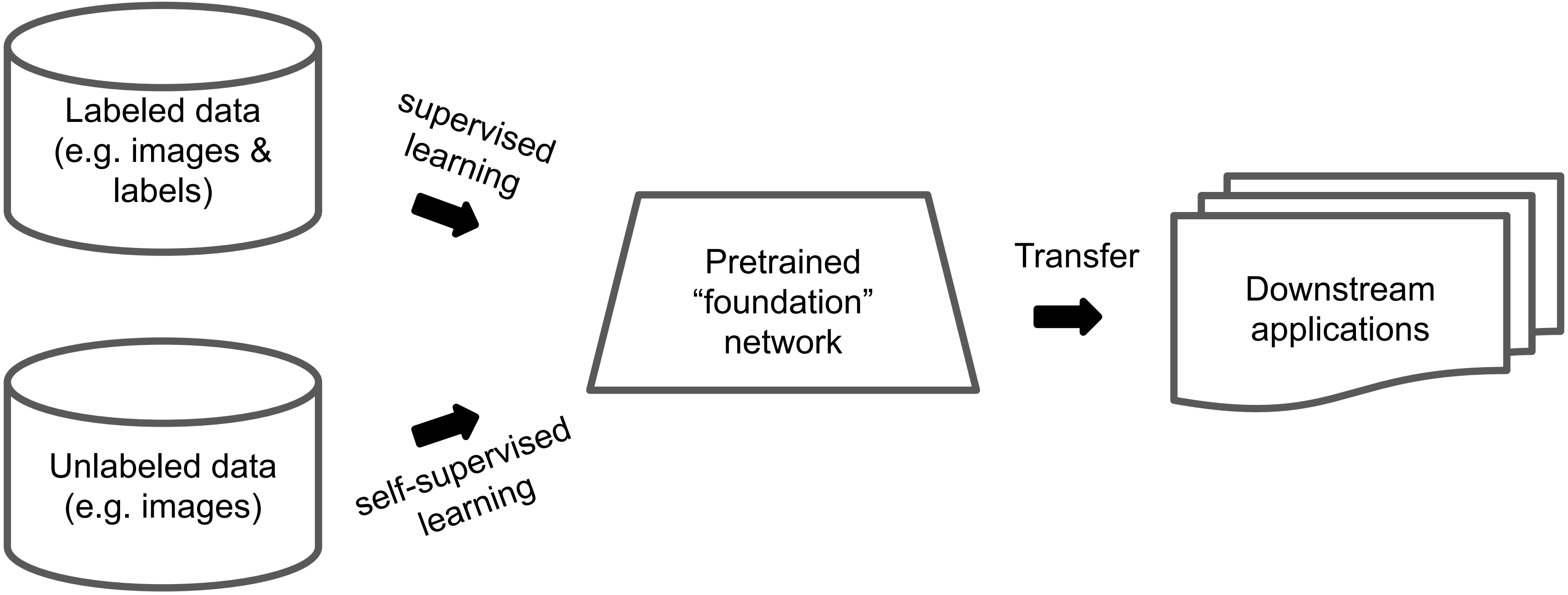
What’s covered in this talk?

- Motivation for contrastive learning
- Contrastive learning with negative examples
- Contrastive learning without negative examples
- Important design choices in contrastive learning
- Open challenges for contrastive learning

Motivation for contrastive learning

Google Research

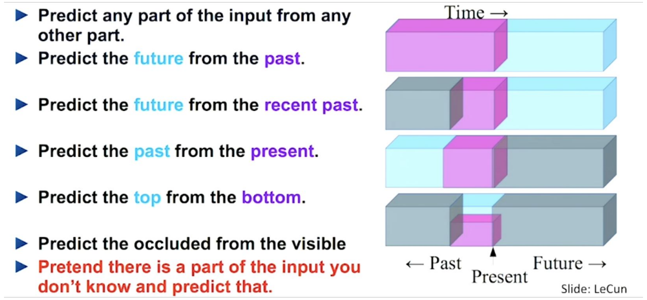
The paradigm of learning “foundation” models



“Self-supervised learning” is “supervised learning” without specific task annotations.

Learning by prediction in an abstract space

- One form of intelligence is the ability to predict
- →
- Predict abstracted states instead of raw inputs
- →
- Avoid *collapse* by a contrastive loss
  - Pull together positive states
  - Push away negative states
- →
- But what to predict?



- ▶ Predict any part of the input from any other part.
- ▶ Predict the future from the past.
- ▶ Predict the future from the recent past.
- ▶ Predict the past from the present.
- ▶ Predict the top from the bottom.
- ▶ Predict the occluded from the visible
- ▶ Pretend there is a part of the input you don't know and predict that.

Slide: LeCun

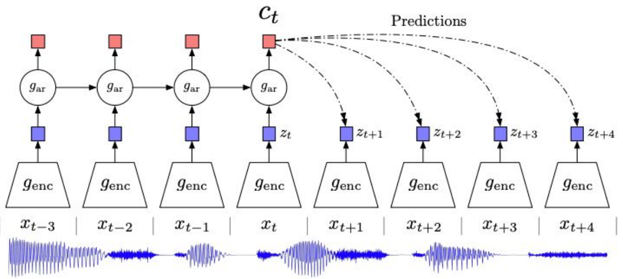


Figure 1: Overview of Contrastive Predictive Coding, the proposed representation learning approach. Although this figure shows audio as input, we use the same setup for images, text and reinforcement learning.

[Oord et al, Representation Learning with Contrastive Predictive Coding, 2018]

A multi-view agreement prediction task

- Predict instance identity (each instance as a class)

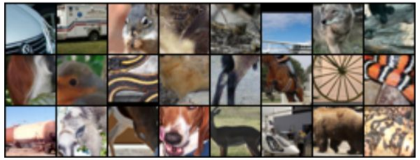


Figure 1: Exemplary patches sampled from

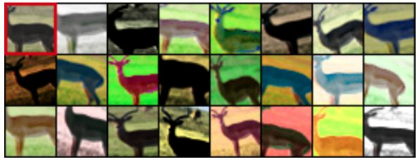
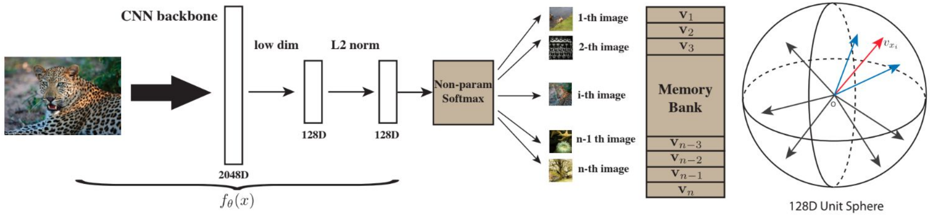


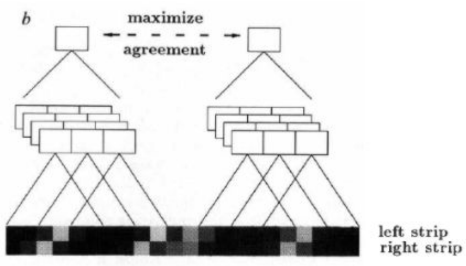
Figure 2: Several random transformations

[Dosovitskiy et al, NeurIPS'14]

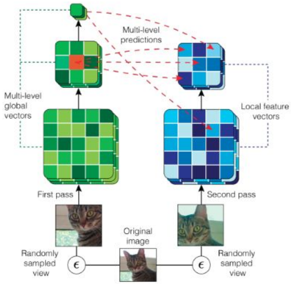


[Wu et al, CVPR'18]

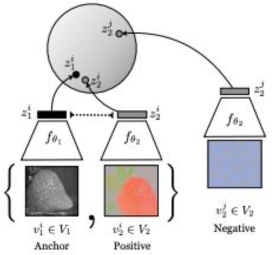
- Predict other views of the same example



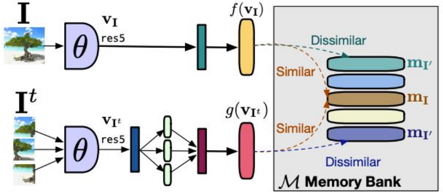
[Becker & Hinton, Nature'92]



[Bachman et al, NuerIPS'19]



[Tian et al, ECCV'20]



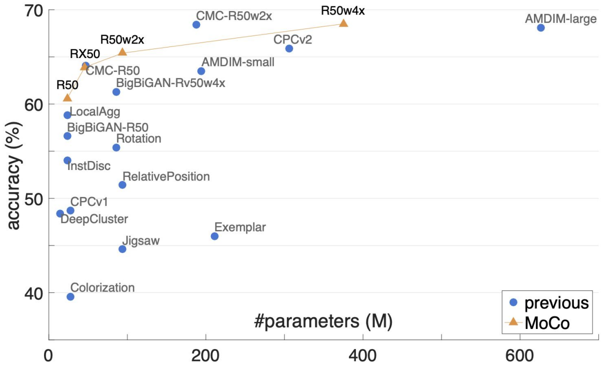
[Misra et al, CVPR'20]

(and many others…)

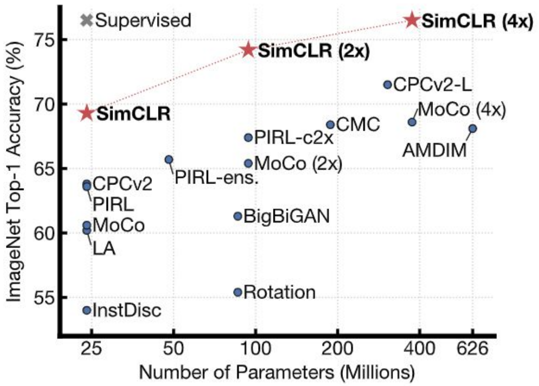
Many exciting results

Linear evaluation of representations

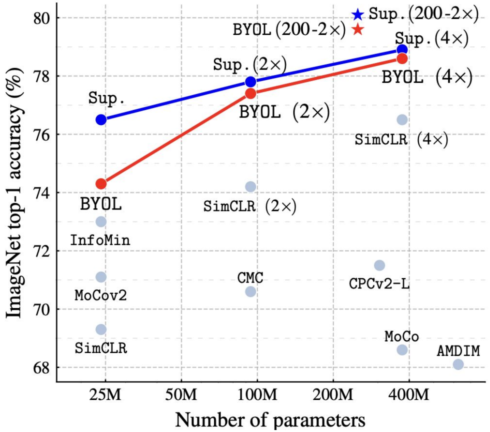
MoCo
CVPR'20



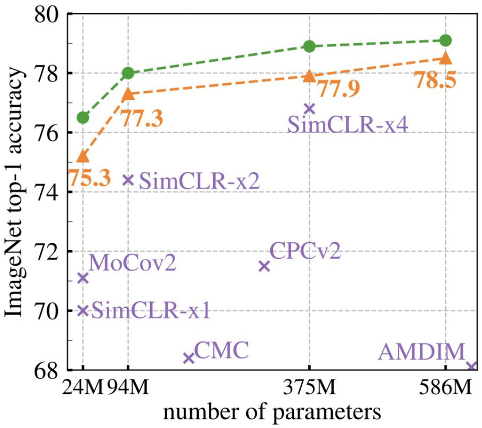
SimCLR
ICML'20



BYOL
NeurIPS'20

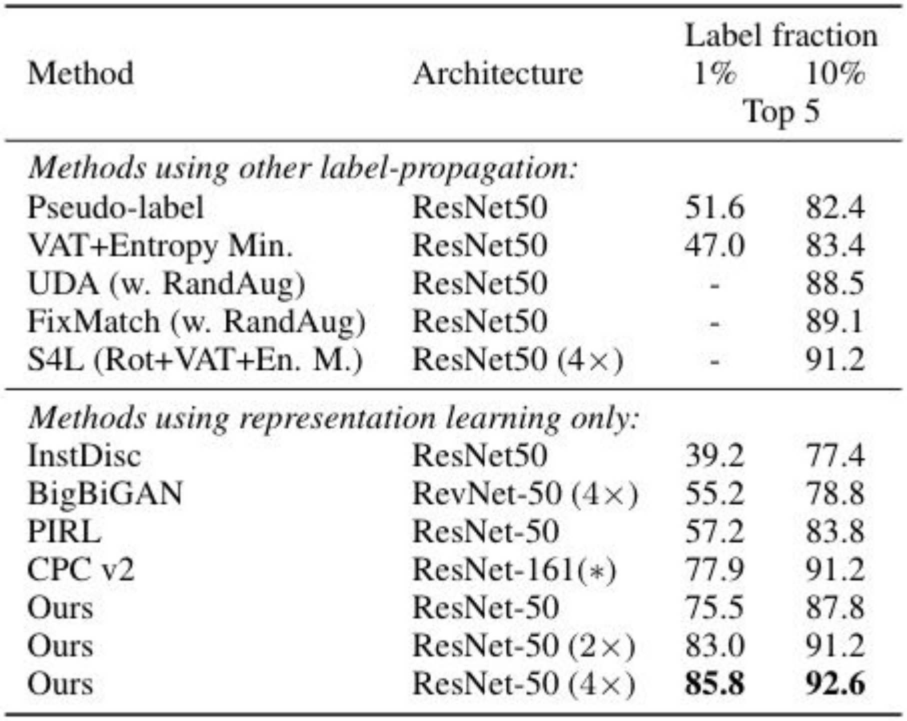


SwAV
NeurIPS'20



Semi-supervised learning

SimCLR as an example: strong semi-supervised learners, outperforms AlexNet with 100X fewer labels.

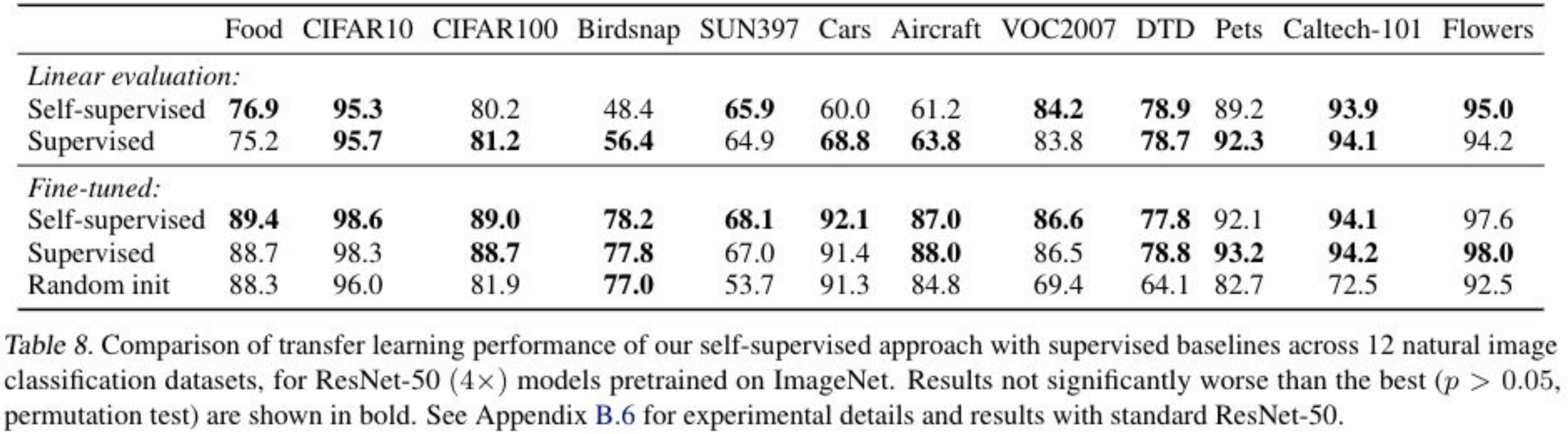


Method	Architecture	Label fraction	
		1%	10%
		Top 5	
Methods using other label-propagation:			
Pseudo-label	ResNet50	51.6	82.4
VAT+Entropy Min.	ResNet50	47.0	83.4
UDA (w. RandAug)	ResNet50	-	88.5
FixMatch (w. RandAug)	ResNet50	-	89.1
S4L (Rot+VAT+En. M.)	ResNet50 (4×)	-	91.2
Methods using representation learning only:			
InstDisc	ResNet50	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
Ours	ResNet-50	75.5	87.8
Ours	ResNet-50 (2×)	83.0	91.2
Ours	ResNet-50 (4×)	**85.8**	**92.6**

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

Transfer learning

SimCLR as an example: matches or surpasses supervised ImageNet pretraining when transferring to other classification tasks.



	Food	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Linear evaluation:												
Self-supervised	**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:												
Self-supervised	**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×) models pretrained on ImageNet. Results not significantly worse than the best ($p > 0.05$, permutation test) are shown in bold. See Appendix B.6 for experimental details and results with standard ResNet-50.

* The two datasets, where the supervised ImageNet pretrained model is better, are Pets and Flowers, which share a portion of labels with ImageNet.

Google Research

Contrastive learning with negative examples

(with SimCRL and MoCo as examples)

SimCLR

[Chen et al, A simple framework for contrastive learning of visual representations, ICML’20]

Maximizing the agreement of representations under data transformation, using a contrastive loss in the latent/feature space.

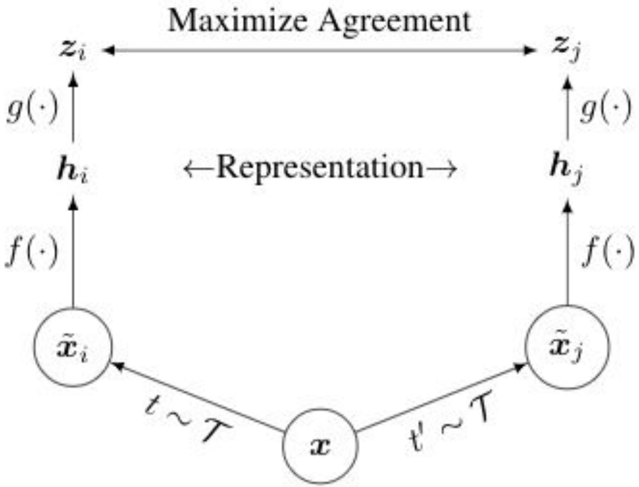
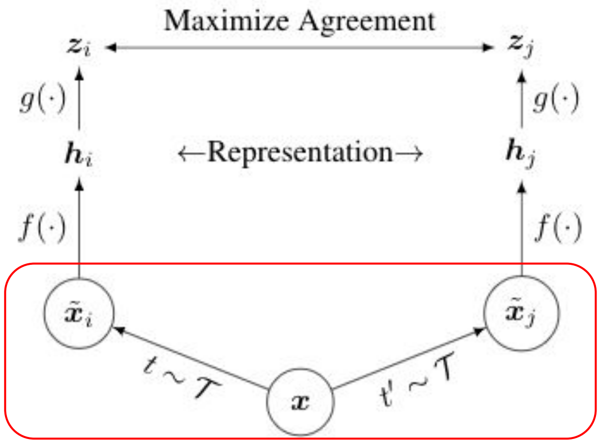


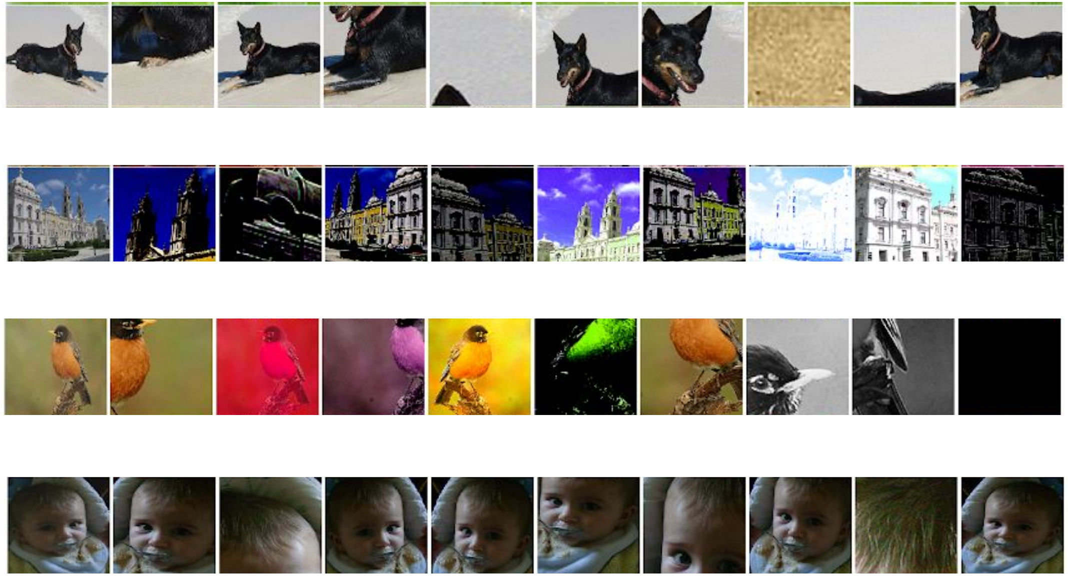
Figure 2. A framework for contrastive representation learning. Two separate stochastic data augmentations $t, t' \sim \mathcal{T}$ are applied to each example to obtain two correlated views. A base encoder network $f(\cdot)$ with a projection head $g(\cdot)$ is trained to maximize agreement in *latent representations* via a contrastive loss.

SimCLR component: data augmentation



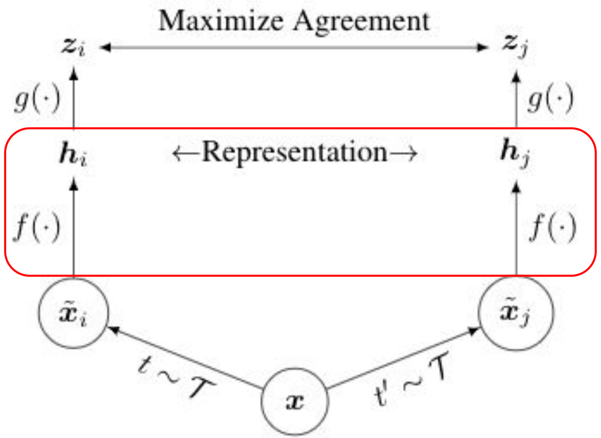
We use random crop and color distortion for augmentation.

Examples of augmentation applied to the left most images:



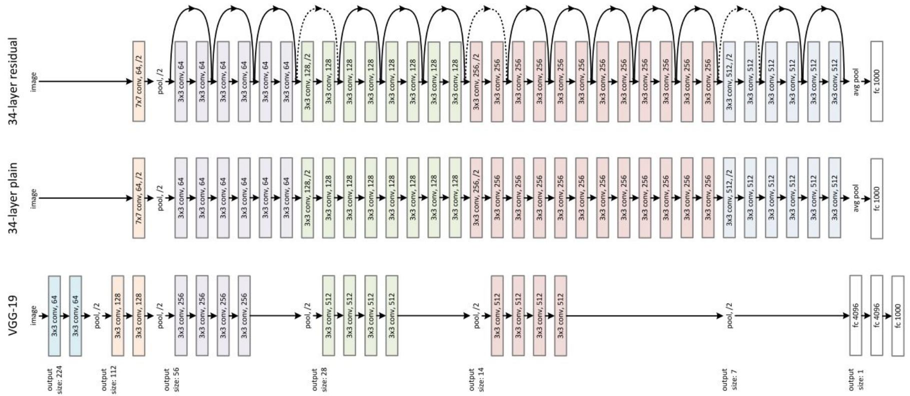
Google Research

SimCLR component: encoder

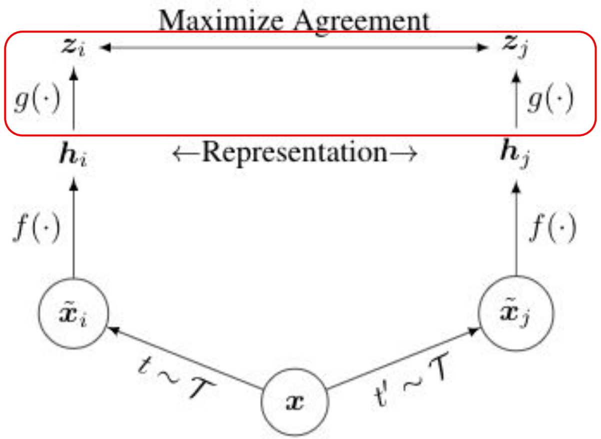


f(x) is the base network that computes internal representation.

We use (unconstrained) ResNet in this work. However, it can be other networks.

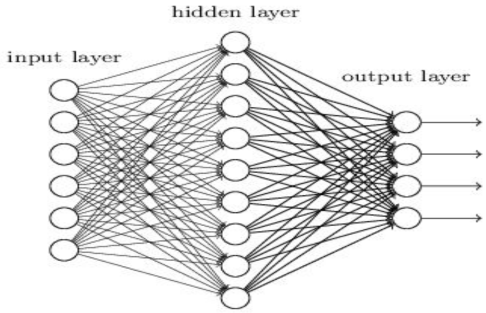


SimCLR component: projection head

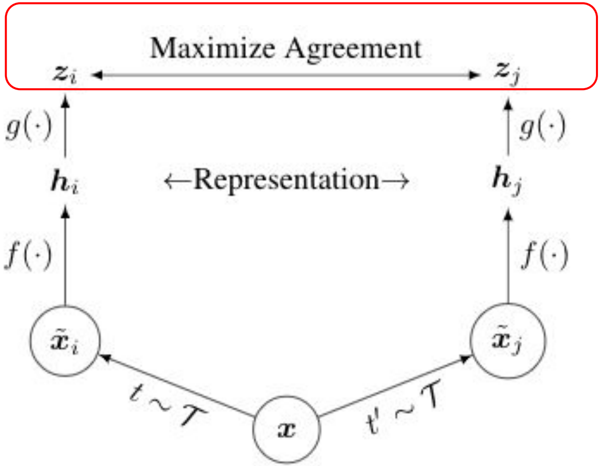


g(h) is a projection network that project representation to a latent space.

We use a MLP (with non-linearity).

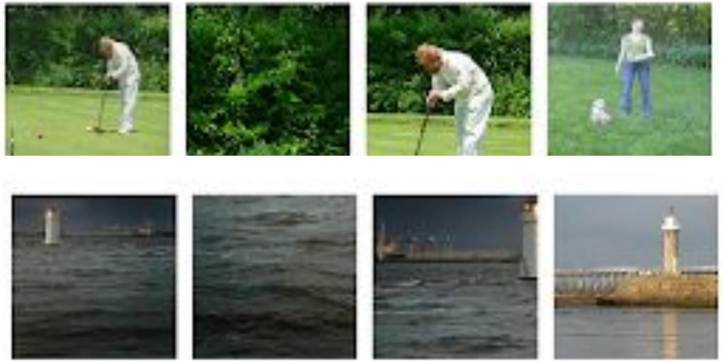


SimCLR component: contrastive loss



Maximize agreement using a contrastive task:

Given {x_k} where two different examples x_i and x_j are a positive pair, identify x_j in {x_k}_{k!=i} for x_i.



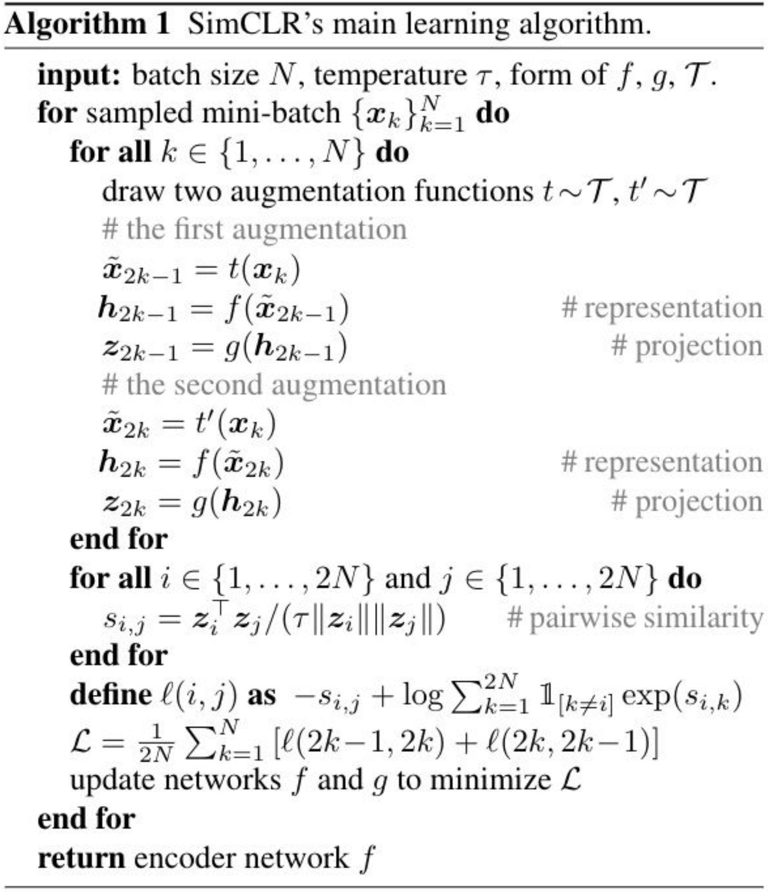
Original image crop 1 crop 2 contrastive image

Loss function:

$$\text{Let } \operatorname{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top \boldsymbol{v} / \|\boldsymbol{u}\| \|\boldsymbol{v}\|$$

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

SimCLR pseudo code and illustration



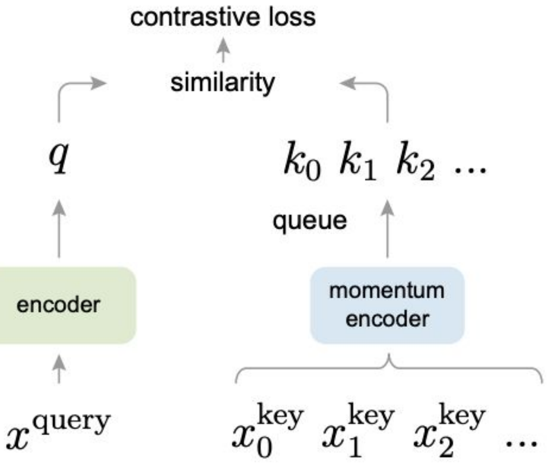
Algorithm 1 SimCLR's main learning algorithm.

```
input: batch size $N$, temperature $\tau$, form of $f$, $g$, $\mathcal{T}$.
for sampled mini-batch $\{\boldsymbol{x}_k\}_{k=1}^N$ do
    for all $k \in \{1, \ldots, N\}$ do
        draw two augmentation functions $t \sim \mathcal{T}$, $t' \sim \mathcal{T}$
        # the first augmentation
        $\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)$
        $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$                # representation
        $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$                        # projection
        # the second augmentation
        $\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)$
        $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$                    # representation
        $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$                            # projection
    end for
    for all $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ do
        $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\tau \|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$      # pairwise similarity
    end for
    define $\ell(i,j)$ as $-s_{i,j} + \log \sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k})$
    $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
    update networks $f$ and $g$ to minimize $\mathcal{L}$
end for
return encoder network $f$
```

GIF credit: Tom Small

More negative examples: MoCo

- SimCLR use images in the same mini-batch as negative examples, so batch size and negatives are tied
- MoCo decouples batch size and negatives by introducing a **momentum encoder**, and a **queue of activations**.



Momentum encoder update with:

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

(instead of backpropagation)

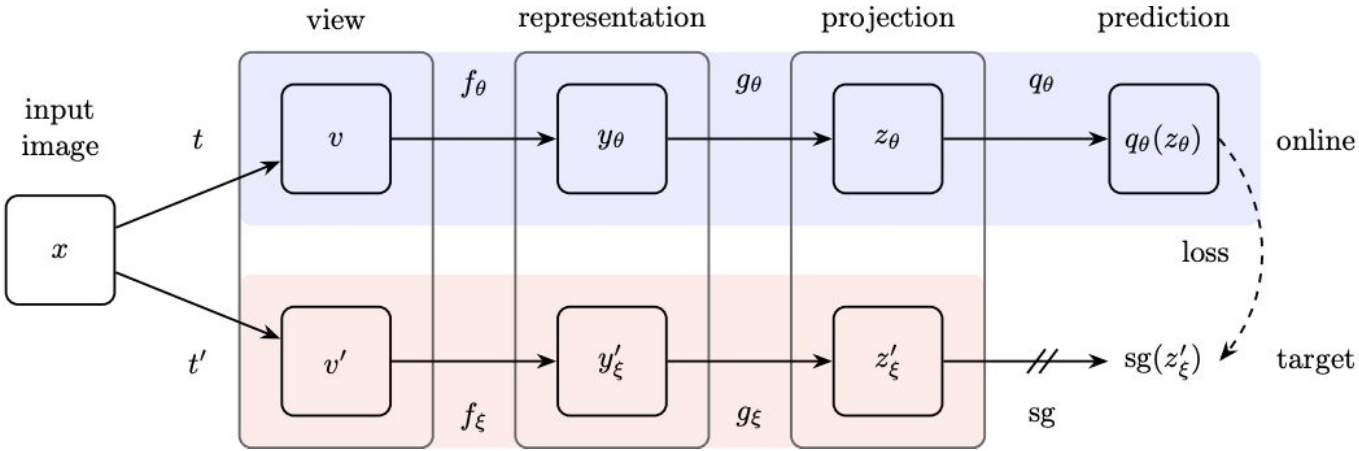
[He et al, Momentum Contrast for Unsupervised Visual Representation Learning, CVPR'20]

Contrastive learning without negative examples

(Using BYOL and SimSiam as main examples)

BYOL

- With momentum encoder and additional predictor network on top of the projection head, the model doesn't collapse even without negative examples.

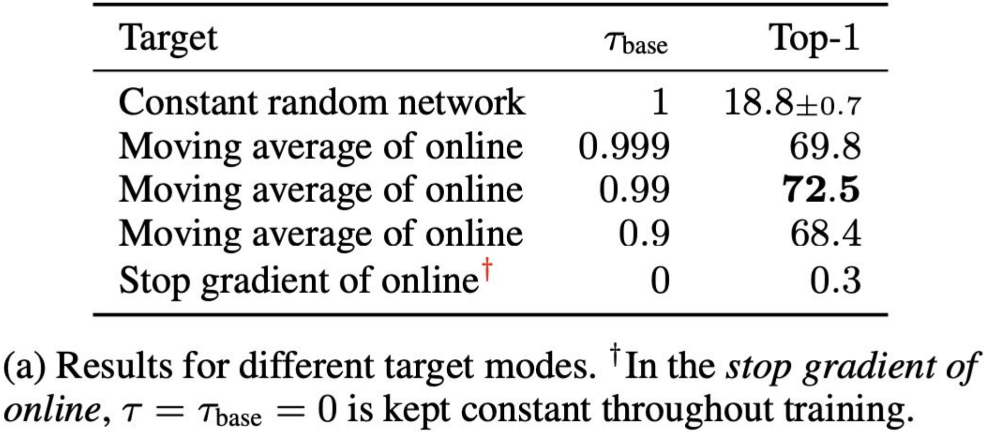


$$\mathcal{L}_{\theta,\xi} \triangleq \left\|\overline{q_\theta}(z_\theta) - \overline{z}'_\xi\right\|_2^2 = 2 - 2\cdot\frac{\langle q_\theta(z_\theta), z'_\xi\rangle}{\left\|q_\theta(z_\theta)\right\|_2\cdot\left\|z'_\xi\right\|_2}.$$

[Grill et al, Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning, NeurIPS'20]

BYOL

- Both **predictor** and **momentum encoder** play an important role in preventing collapse



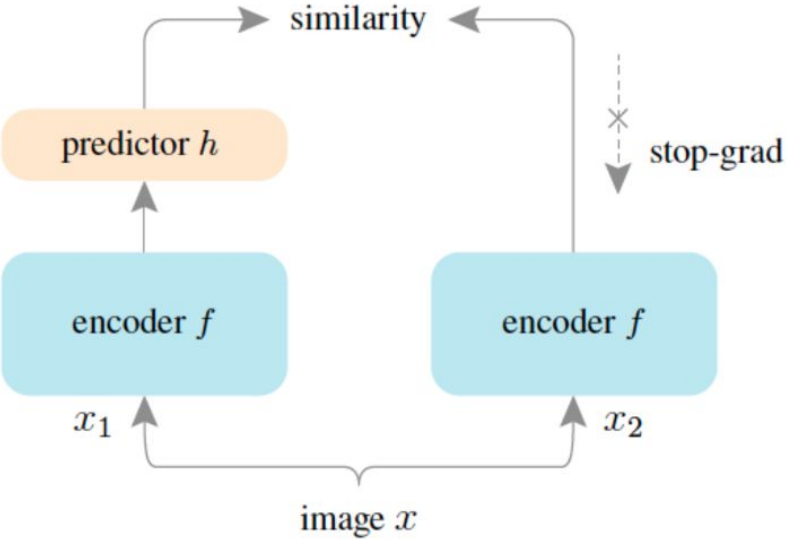
Target	τ_{base}	Top-1
Constant random network	1	$18.8{\pm}0.7$
Moving average of online	0.999	69.8
Moving average of online	0.99	**72.5**
Moving average of online	0.9	68.4
Stop gradient of online†	0	0.3

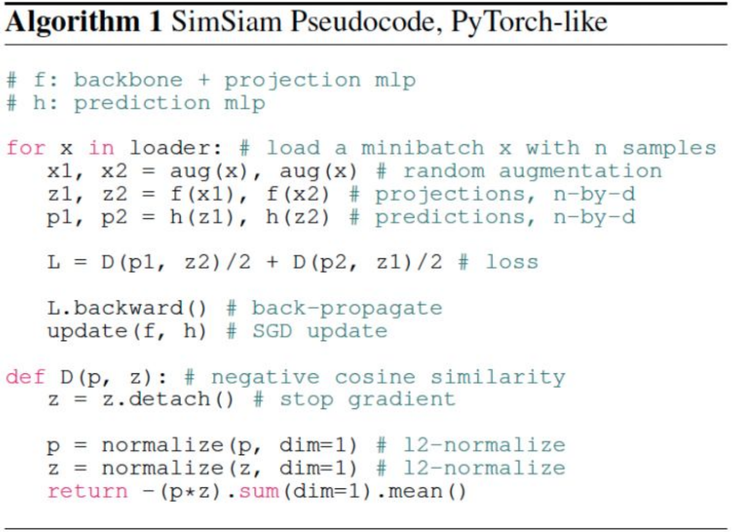
(a) Results for different target modes. †In the *stop gradient of online*, $\tau = \tau_{\text{base}} = 0$ is kept constant throughout training.

[Grill et al, Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning, NeurIPS’20]

SimSiam

- Further simplifies the framework by removing the momentum encoder.





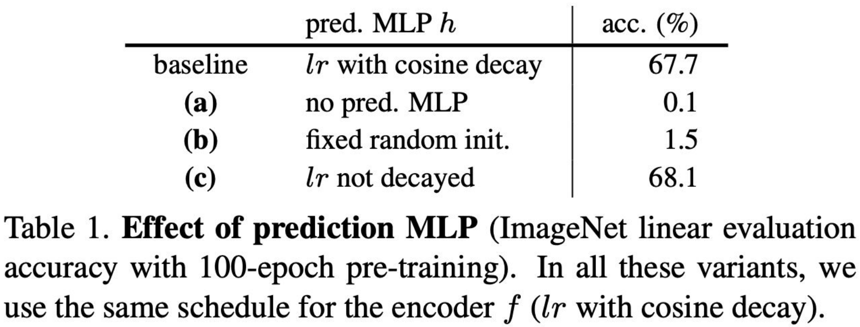
Algorithm 1 SimSiam Pseudocode, PyTorch-like

```
# f: backbone + projection mlp
# h: prediction mlp

for x in loader: # load a minibatch x with n samples
   x1, x2 = aug(x), aug(x) # random augmentation
   z1, z2 = f(x1), f(x2) # projections, n-by-d
   p1, p2 = h(z1), h(z2) # predictions, n-by-d

   L = D(p1, z2)/2 + D(p2, z1)/2 # loss

   L.backward() # back-propagate
   update(f, h) # SGD update

def D(p, z): # negative cosine similarity
   z = z.detach() # stop gradient

   p = normalize(p, dim=1) # l2-normalize
   z = normalize(z, dim=1) # l2-normalize
   return -(p*z).sum(dim=1).mean()
```

[Chen and He, Exploring Simple Siamese Representation Learning, CVPR 2021]

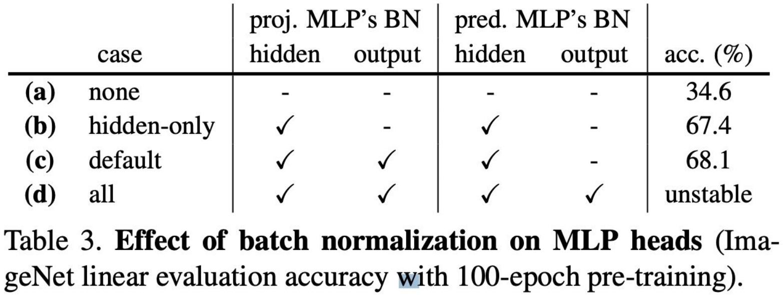
SimSiam

- Key ingredients:
  - No momentum encoder but still has stop-gradient
    - equal to setting ema factor to 0.
  - Careful design of predictor:
    - Batch Norm & bottleneck structure & no lr decay



	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).



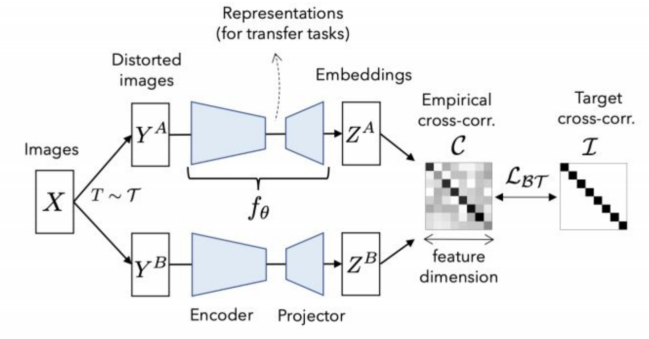
		proj. MLP's BN		pred. MLP's BN		
	case	hidden	output	hidden	output	acc. (%)
(a)	none	-	-	-	-	34.6
(b)	hidden-only	✓	-	✓	-	67.4
(c)	default	✓	✓	✓	-	68.1
(d)	all	✓	✓	✓	✓	unstable

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

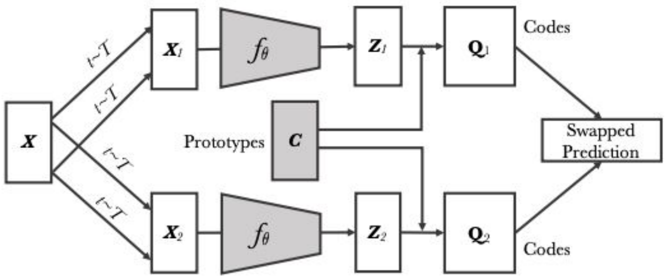
[Chen and He, Exploring Simple Siamese Representation Learning, CVPR 2021]

Avoid collapse by using other batch statistics

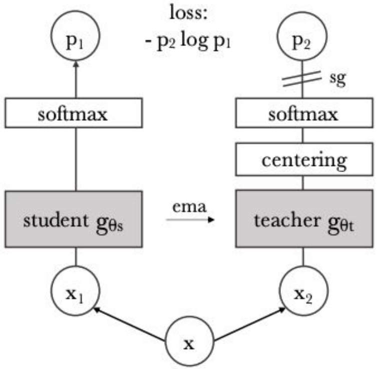
- We can avoid representation collapse using neither negative examples nor predictor: other batch statistics can work.



Barlow Twins
[Zbontar et al, ICML'21]



SWAV
[Caron et al, NeurIPS'20]



DINO
[Caron et al, CVPR'21]

Also: SWD distribution loss, [chen et al, intriguing properties of contrastive losses, 2021]

Some important design choices in contrastive learning

(using SimCLR as main example)

Evaluation setup

Main dataset for self-supervised pretraining:

- ImageNet (without labels)

Two evaluation protocols for the remaining slides

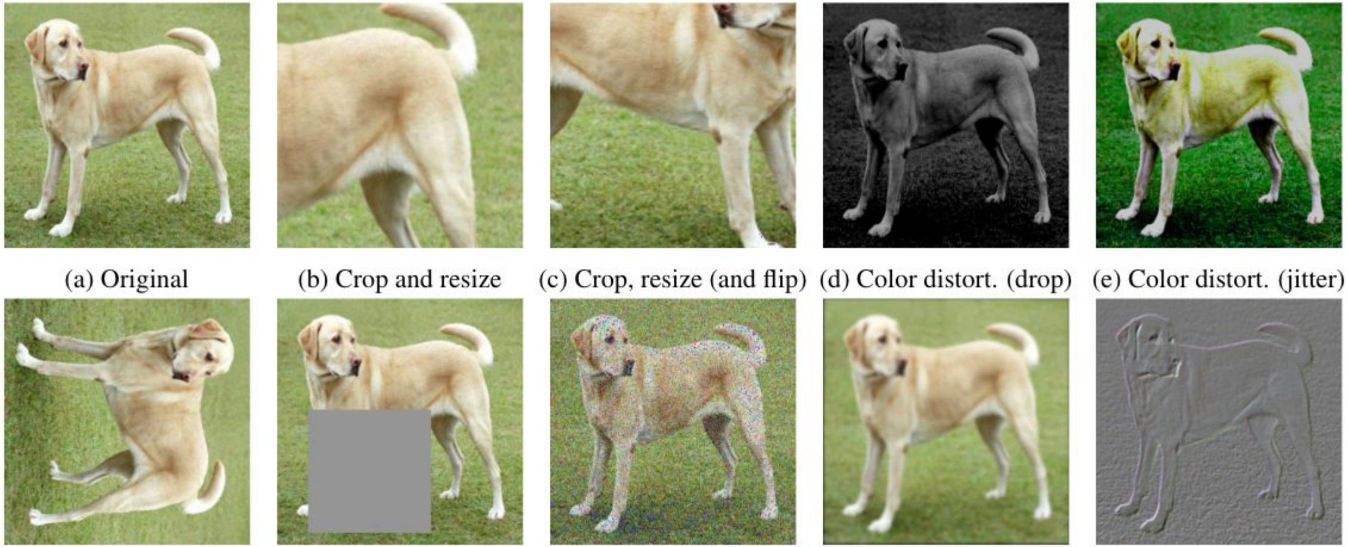
- Linear classifier trained on learned features
- Fine-tune the model (with few labels)

Important design choice in Contrastive Learning:

1. Data Augmentation is critical

A set of transformations studied in SimCLR

Systematically study a set of augmentation



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

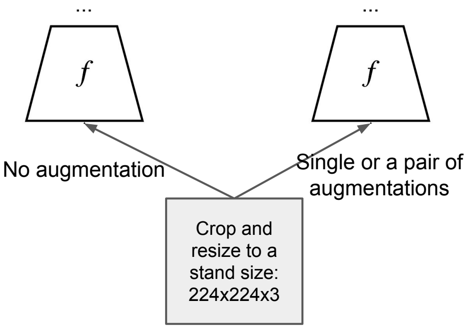
(f) Rotate {90°, 180°, 270°} (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

* Note that we only test these for ablation, the augmentation policy used to train our models only involves random crop (with flip and resize) + color distortion + Gaussian blur.

[Figures from SimCLR paper]

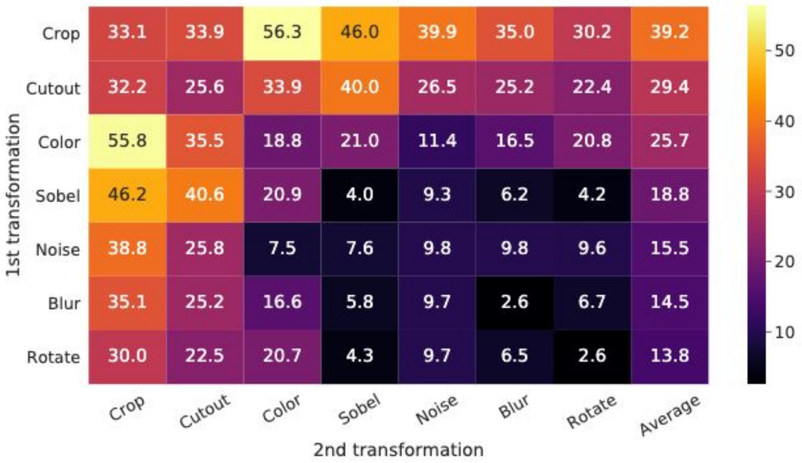
Studying single or a pair of augmentations

- ImageNet images are of different resolutions, so random crops are typically applied.
- To remove co-founding
  - First random crop an image and resize to a standard resolution.
  - Then apply a single or a pair of augmentations on one branch, while keeping the other as identity mapping.
  - This is suboptimal than applying augmentations to both branches, but sufficient for ablation.


...
...
f
f
No augmentation
Single or a pair of augmentations
Crop and resize to a stand size: 224x224x3

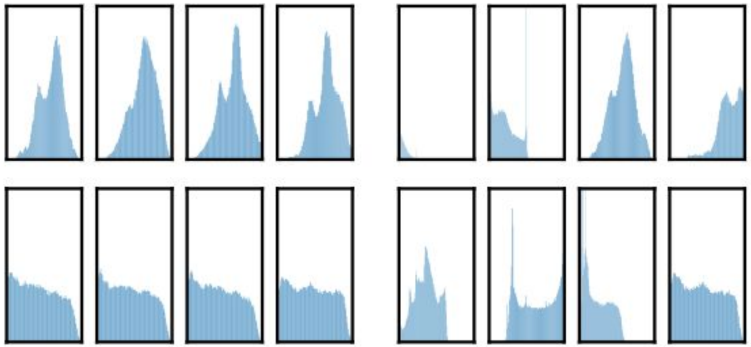
Composition of augmentations are crucial

Composition of crop and color stands out!



	Crop	Cutout	Color	Sobel	Noise	Blur	Rotate	Average
Crop	33.1	33.9	56.3	46.0	39.9	35.0	30.2	39.2
Cutout	32.2	25.6	33.9	40.0	26.5	25.2	22.4	29.4
Color	55.8	35.5	18.8	21.0	11.4	16.5	20.8	25.7
Sobel	46.2	40.6	20.9	4.0	9.3	6.2	4.2	18.8
Noise	38.8	25.8	7.5	7.6	9.8	9.8	9.6	15.5
Blur	35.1	25.2	16.6	5.8	9.7	2.6	6.7	14.5
Rotate	30.0	22.5	20.7	4.3	9.7	6.5	2.6	13.8

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.



(a) Without color distortion.

(b) With color distortion.

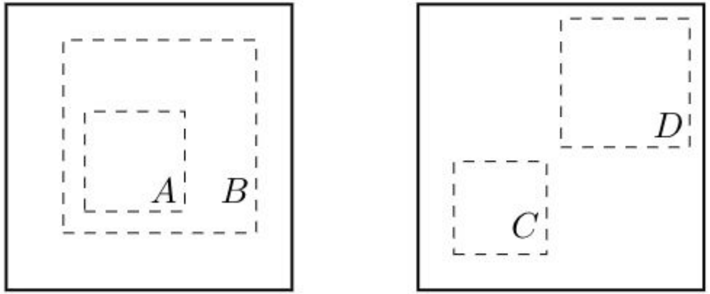
Figure 6. Histograms of pixel intensities (over all channels) for different crops of two different images (i.e. two rows). The image for the first row is from Figure 4. All axes have the same range.

[Figures from SimCLR paper]

Random cropping gives the major learning signal

Simply via Random Crop (with resize to standard size), we can mimic (1) global to local view prediction, and (2) neighboring view prediction.

This simple transformation defines a family of predictive tasks.



(a) Global and local views.

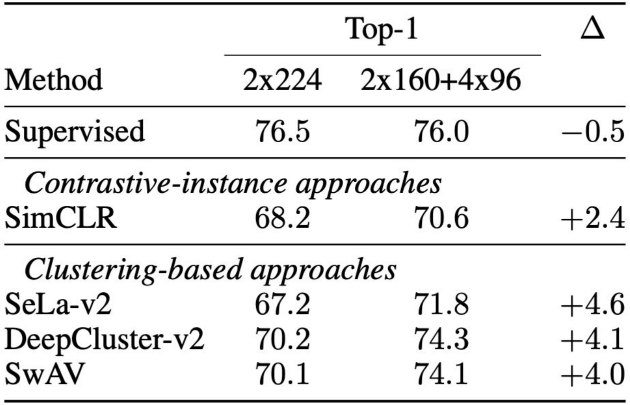
(b) Adjacent views.

Figure 3. By randomly cropping and resizing images (solid rectangles) to a standard size, we sample contrastive prediction tasks that mimic global to local view ($B \rightarrow A$) or neighbouring view ($D \rightarrow C$) prediction.

[Figures from SimCLR paper]

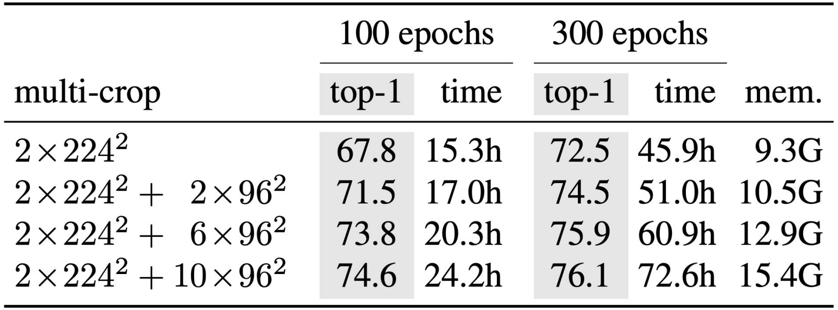
An enhancement of random cropping

Instead of taking two crops of the same size, one may take multiple crops of different sizes.



	Top-1		Δ
Method	2x224	2x160+4x96	
Supervised	76.5	76.0	−0.5
Contrastive-instance approaches			
SimCLR	68.2	70.6	+2.4
Clustering-based approaches			
SeLa-v2	67.2	71.8	+4.6
DeepCluster-v2	70.2	74.3	+4.1
SwAV	70.1	74.1	+4.0

SwAV [Caron et al, NeurIPS 2020]

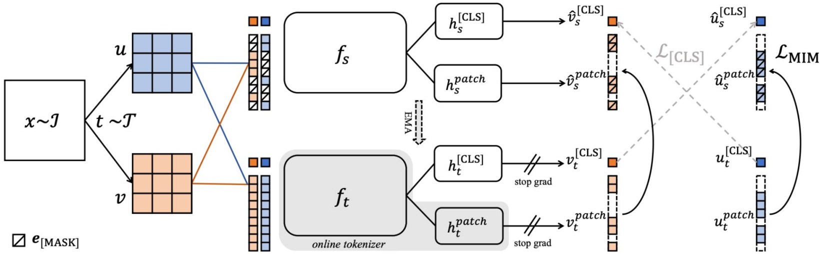


	100 epochs		300 epochs		
multi-crop	top-1	time	top-1	time	mem.
2×224^2	67.8	15.3h	72.5	45.9h	9.3G
$2\times224^2 + 2\times96^2$	71.5	17.0h	74.5	51.0h	10.5G
$2\times224^2 + 6\times96^2$	73.8	20.3h	75.9	60.9h	12.9G
$2\times224^2 + 10\times96^2$	74.6	24.2h	76.1	72.6h	15.4G

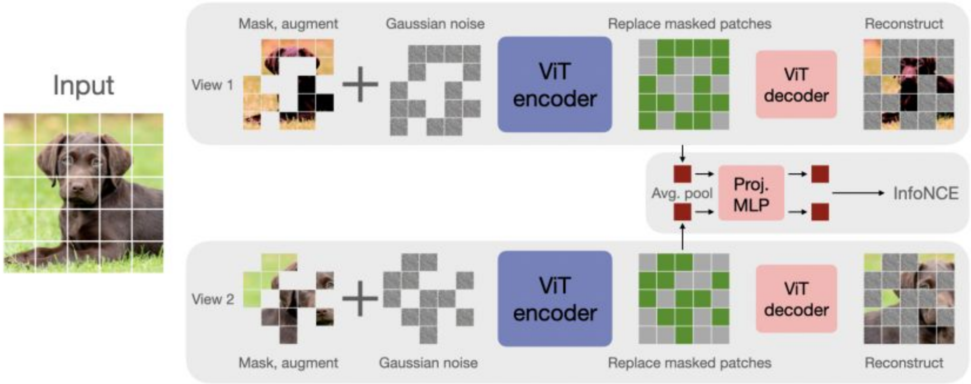
DINO [Caron et al, CVPR 2021]

Patch masking as additional augmentation

Recently, there are methods leveraging image patch masking as additional augmentation. Some examples:



[Zhou et al, iBOT, ICLR'22]



[Anonymous authors, CAN, ICLR'23 submission]

Important design choice in Contrastive Learning:

2. Projection head is important

A nonlinear projection head improves the representation quality of the layer before it

Compare projection heads (after average pooling of ResNet) in SimCLR:

- Identity mapping
- Linear projection
- Nonlinear projection with one additional hidden layer (and ReLU activation)

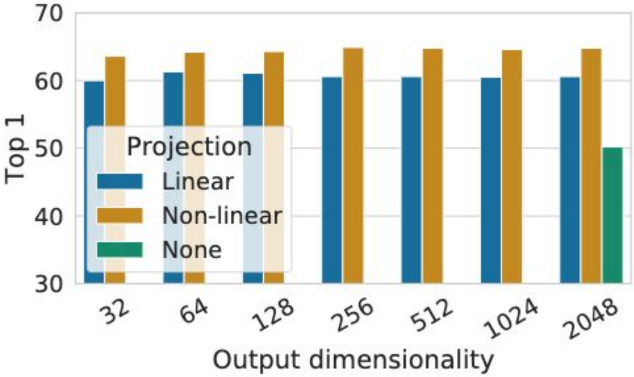


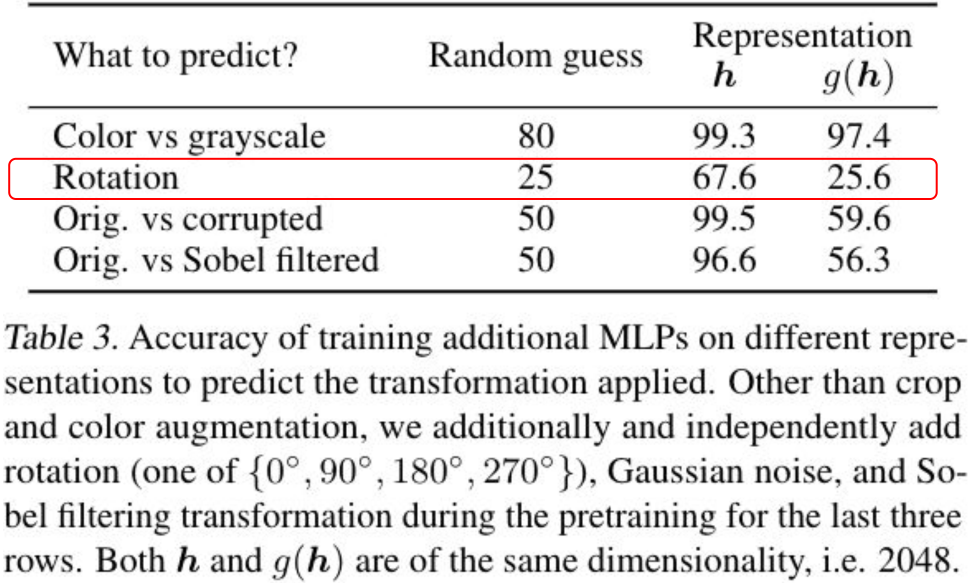
Figure 8. Linear evaluation of pretraining with different projection heads. The dimension of h (before projection) is 2048.

Even when non-linear projection is used, the layer before the projection head,h,is still much better (>10%) than the layer after,z=g(h).

[Figures from SimCLR paper]

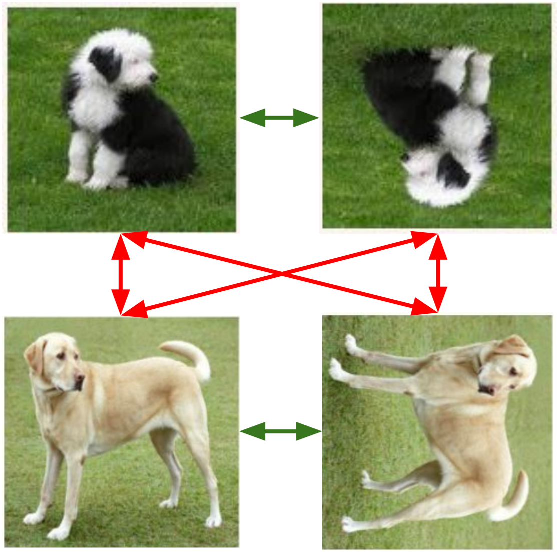
A nonlinear projection head improves the representation quality of the layer before it

To understand why this happens, we measure information in ***h*** and ***z=g(h)***



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.



Contrastive loss can remove/damping rotation information in the last layer when the model is asked to identify rotated variant of an image.

[Figures from SimCLR paper]

Google Research

Important design choice in Contrastive Learning:

3. Model size

Unsupervised contrastive learning benefits (more) from bigger models

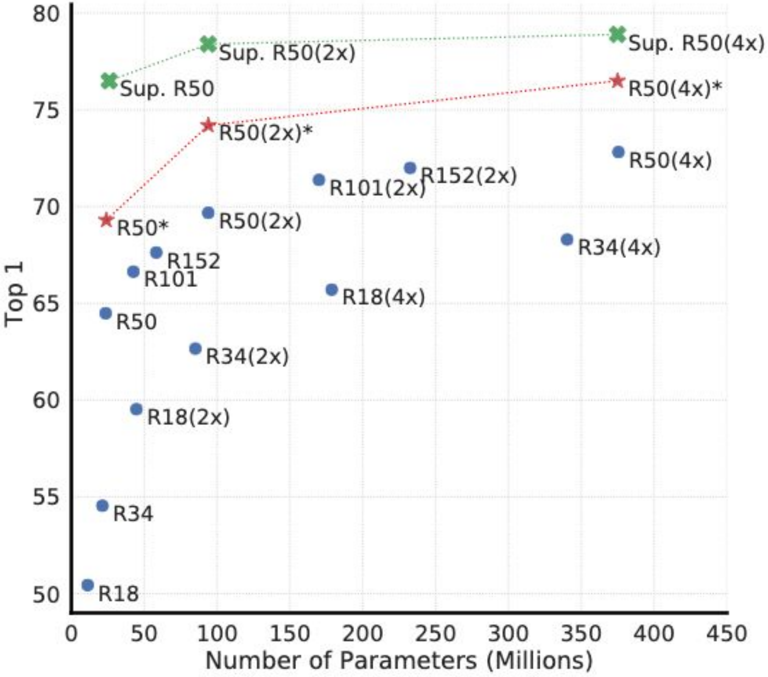


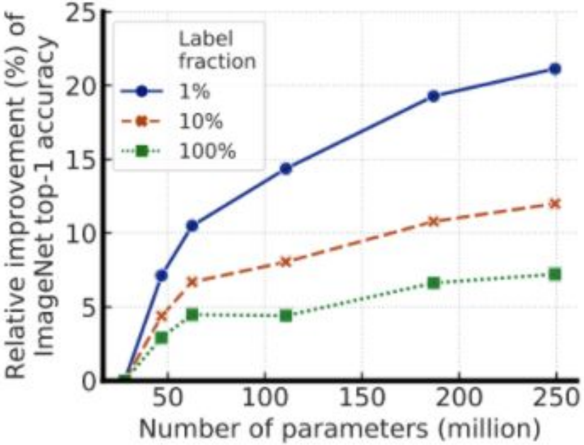
Figure 7. Linear evaluation of models with varied depth and width. Models in blue dots are ours trained for 100 epochs, models in red stars are ours trained for 1000 epochs, and models in green crosses are supervised ResNets (He et al., 2016).[7]

[Figures from SimCLR paper]

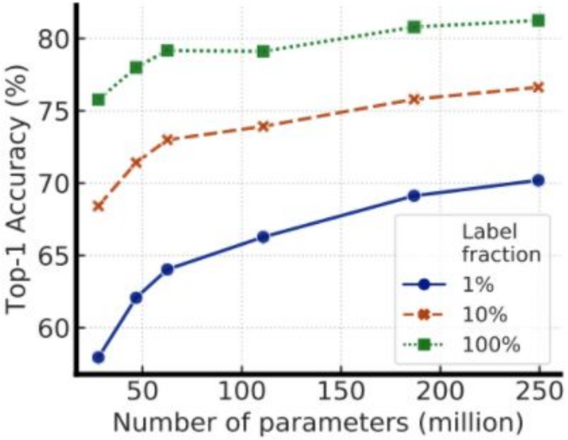
Bigger models are more label-efficient

- Using pre-training + fine-tuning, “the fewer the labels, the bigger the model”
- Increasing the size of model size by 10X, it reduces required labels to achieve certain accuracy by 10X.

Relative improvement (%)

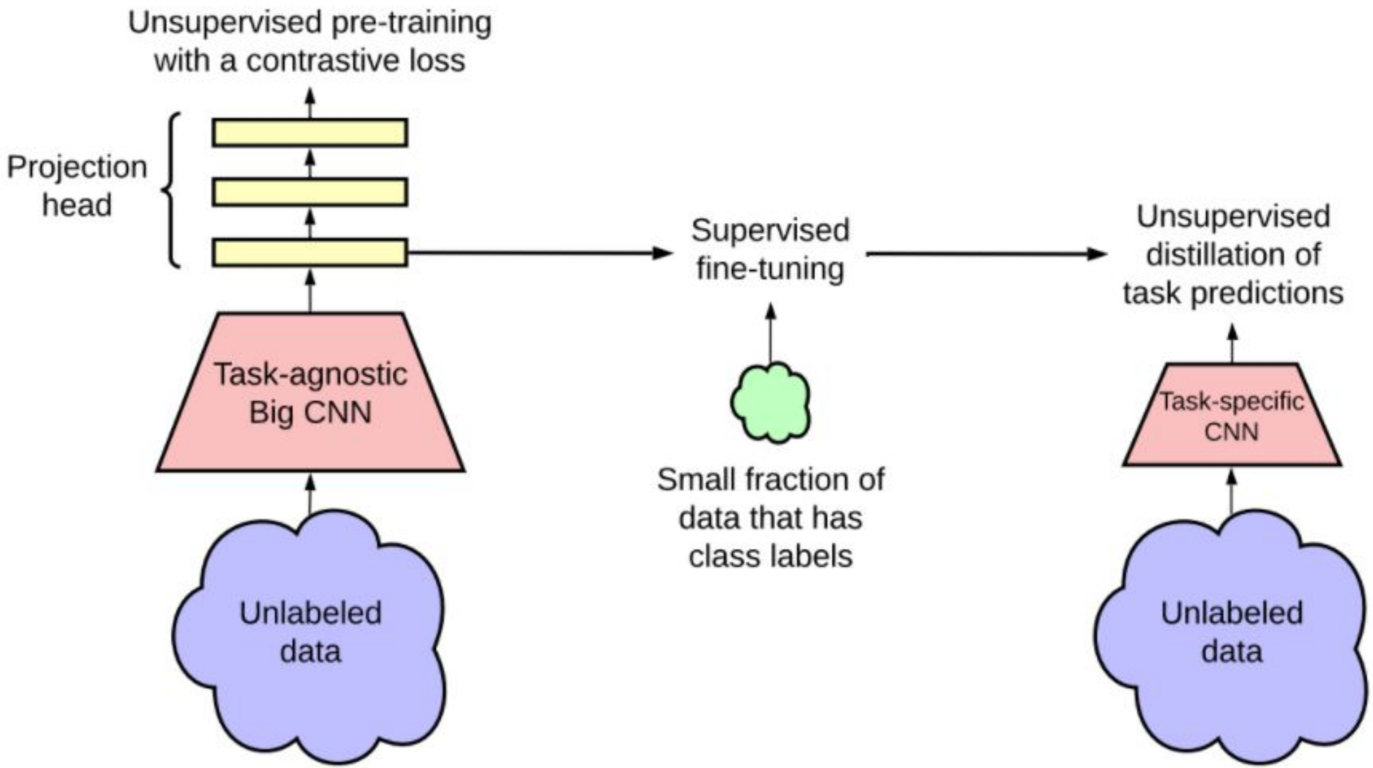


ImageNet top-1 (%)



[Figures from SimCLRv2 paper]

Distill/Self-train with unlabeled data to reduce the model size


Unsupervised pre-training with a contrastive loss
Projection head
Task-agnostic Big CNN
Unlabeled data
Supervised fine-tuning
Small fraction of data that has class labels
Unsupervised distillation of task predictions
Task-specific CNN
Unlabeled data

[Figures from SimCLRv2 paper]

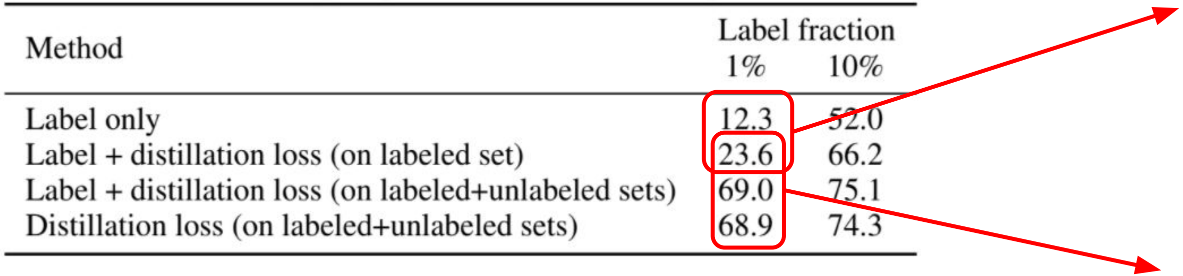
Distillation / self-training with unlabeled data

- To distill the task specific knowledge, we use the teacher model to provide task-specific labels on unlabeled examples, based on which train a student model:

$$\mathcal{L}^{\text{distill}} = -\sum_{\boldsymbol{x}_i \in \mathcal{D}} \left[\sum_{y} P^T(y|\boldsymbol{x}_i;\tau) \log P^S(y|\boldsymbol{x}_i;\tau) \right]$$

- Evaluation on ImageNet with only 1%/10% labels (all images).

Distillation on labeled dataset alone is not sufficient.



Method	Label fraction	
	1%	10%
Label only	12.3	52.0
Label + distillation loss (on labeled set)	23.6	66.2
Label + distillation loss (on labeled+unlabeled sets)	69.0	75.1
Distillation loss (on labeled+unlabeled sets)	68.9	74.3

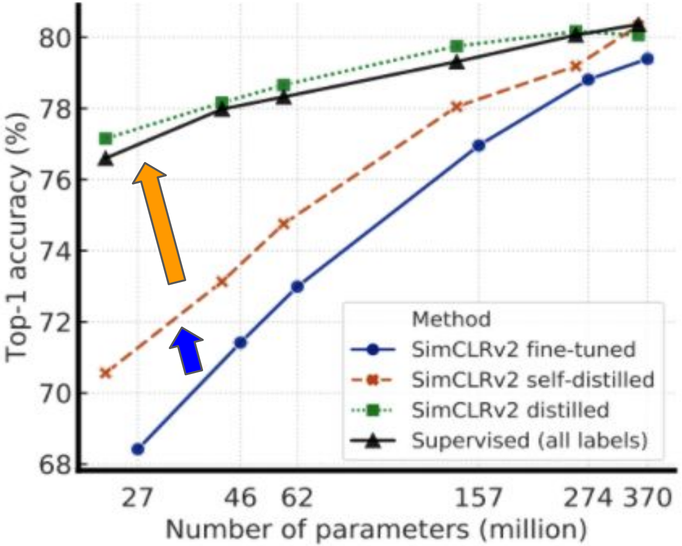
Using unlabeled examples largely improve distillation.

[Figures from SimCLRv2 paper]

Distillation with unlabeled data improves all model sizes

- Both self-distillation and big-model-to-small-model distillation help.
- With 10% of labels, SimCLRv2 can achieve better performance than standard supervised training with 100% of labels.

ImageNet (label fraction: 10%)



↑ Self-distillation / self-training

↑ Distillation from the biggest self-distilled model

[Figures from SimCLRv2 paper]

Google Research

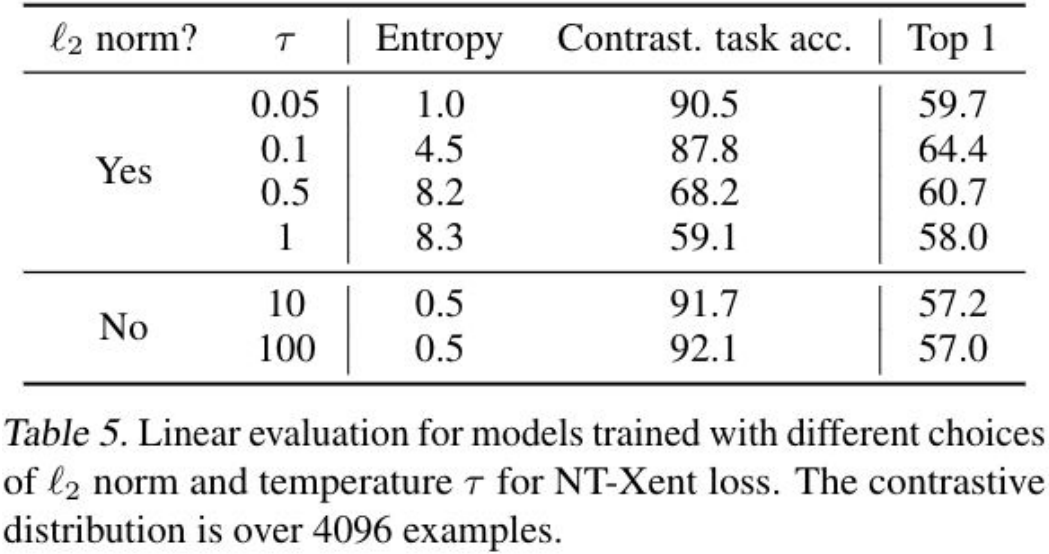
Important design choice in Contrastive Learning:

4. Some hyper-parameters (e.g., in contrastive loss)

Tune normalization and temperature

Compare variants of contrastive (NT-Xent) loss in SimCLR

- L2 normalization with temperature scaling makes a better loss.
- Contrastive accuracy is not correlated with linear evaluation when l2 norm and/or temperature are changed.



ℓ_2 norm?	τ	Entropy	Contrast. task acc.	Top 1
Yes	0.05	1.0	90.5	59.7
	0.1	4.5	87.8	64.4
	0.5	8.2	68.2	60.7
	1	8.3	59.1	58.0
No	10	0.5	91.7	57.2
	100	0.5	92.1	57.0

Table 5. Linear evaluation for models trained with different choices of ℓ_2 norm and temperature τ for NT-Xent loss. The contrastive distribution is over 4096 examples.

[Figures from SimCLR paper]

Contrastive learning benefits from longer training

Compare epochs & batch size in SimCLR (hyper-parameter tuned with batch size of 4096)

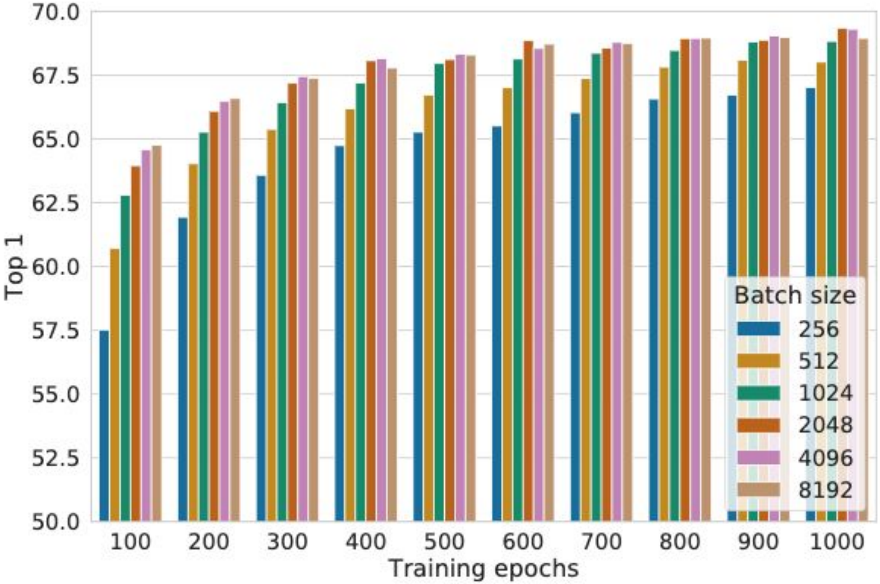


Figure 9. Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch.

[Figures from SimCLR paper]

Small batch sizes work well too with good hparams tuning

Original SimCLR was developed with large batch size, so the hyper-params were not optimized for smaller ones in the above batch size study.

With proper tuning on learning rate, temperature, and deeper projection head, the difference in batch sizes becomes smaller.

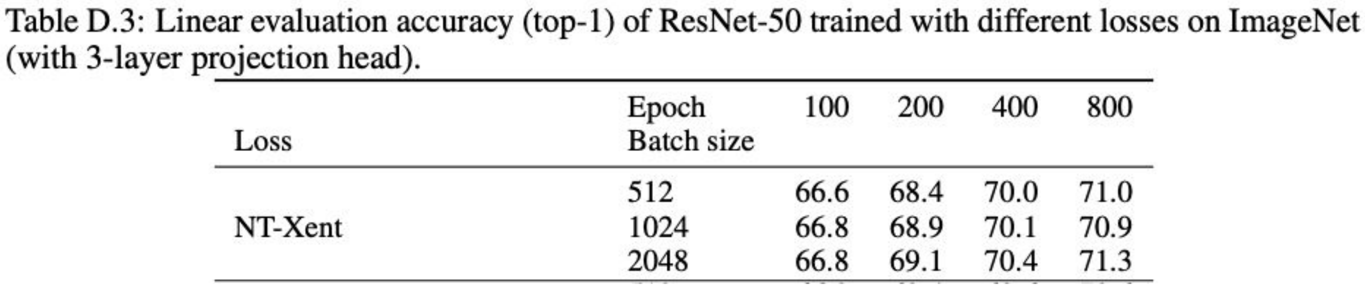


Table D.3: Linear evaluation accuracy (top-1) of ResNet-50 trained with different losses on ImageNet (with 3-layer projection head).

Loss	Epoch Batch size	100	200	400	800
NT-Xent	512	66.6	68.4	70.0	71.0
	1024	66.8	68.9	70.1	70.9
	2048	66.8	69.1	70.4	71.3

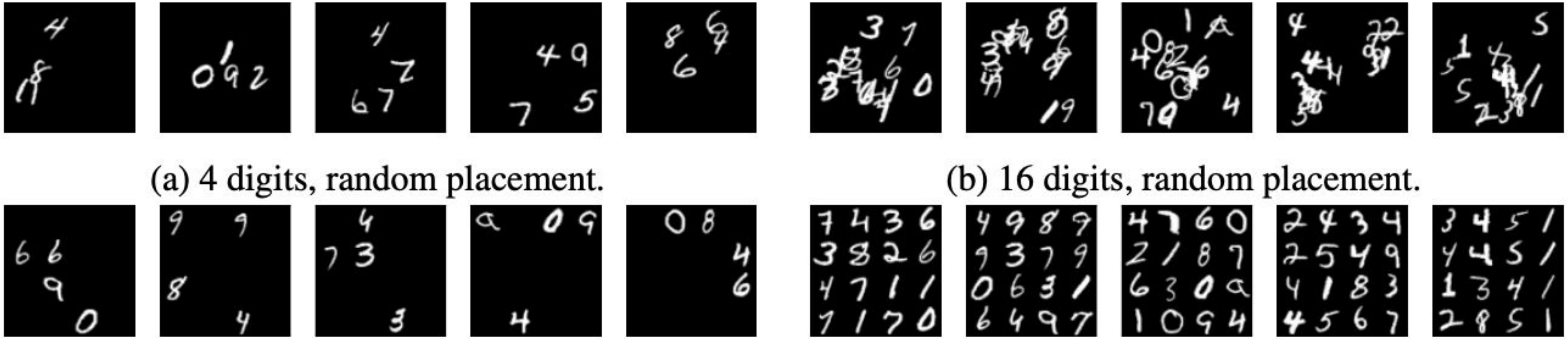
Table from “Intriguing Properties of Contrastive Losses” (Chen et al, 2020)

Some open challenges for contrastive learning

Based on "Intriguing Properties of Contrastive Losses" (https://arxiv.org/abs/2011.02803)

Single object vs Multi objects

- It has been conjectured that many existing contrastive learning is taking advantage of **dataset bias** (e.g. in ImageNet): there's a single/dominant object in the center, and random crops typically share object identity.
- So we construct a dataset of multiple mnist digits



(a) 4 digits, random placement.

(b) 16 digits, random placement.

(c) 4 digits, in-grid placement.

(d) 16 digits, in-grid placement.

[Figures from https://arxiv.org/abs/2011.02803]

Single object vs Multi objects

- The results below show: **SimCLR is able to learn just fine even with multiple mnist digits**

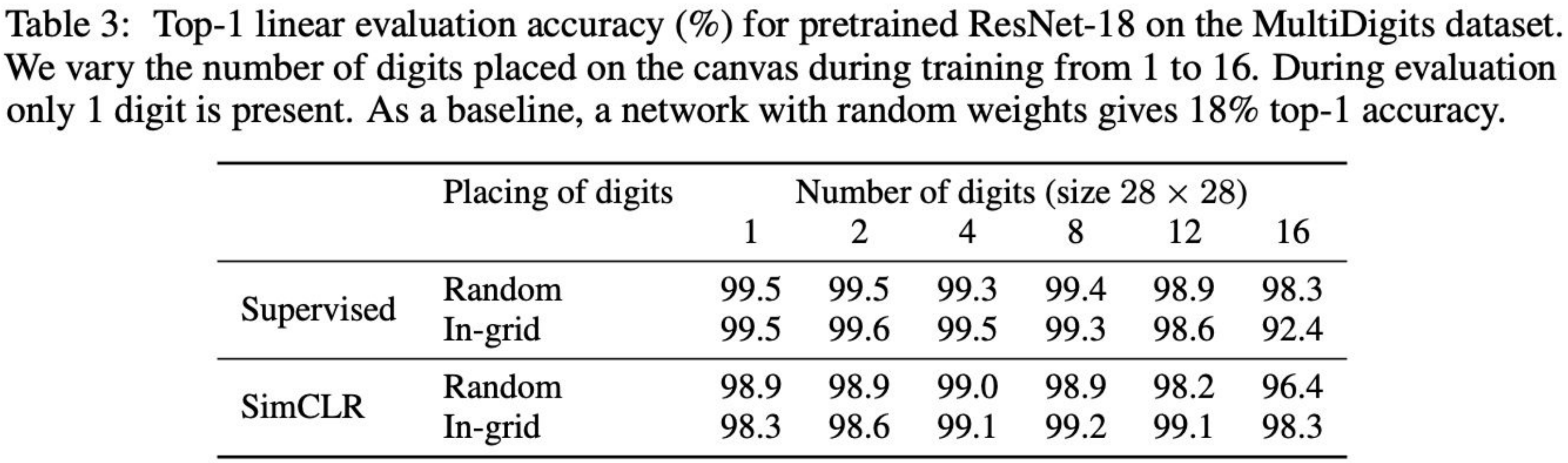


Table 3: Top-1 linear evaluation accuracy (%) for pretrained ResNet-18 on the MultiDigits dataset. We vary the number of digits placed on the canvas during training from 1 to 16. During evaluation only 1 digit is present. As a baseline, a network with random weights gives 18% top-1 accuracy.

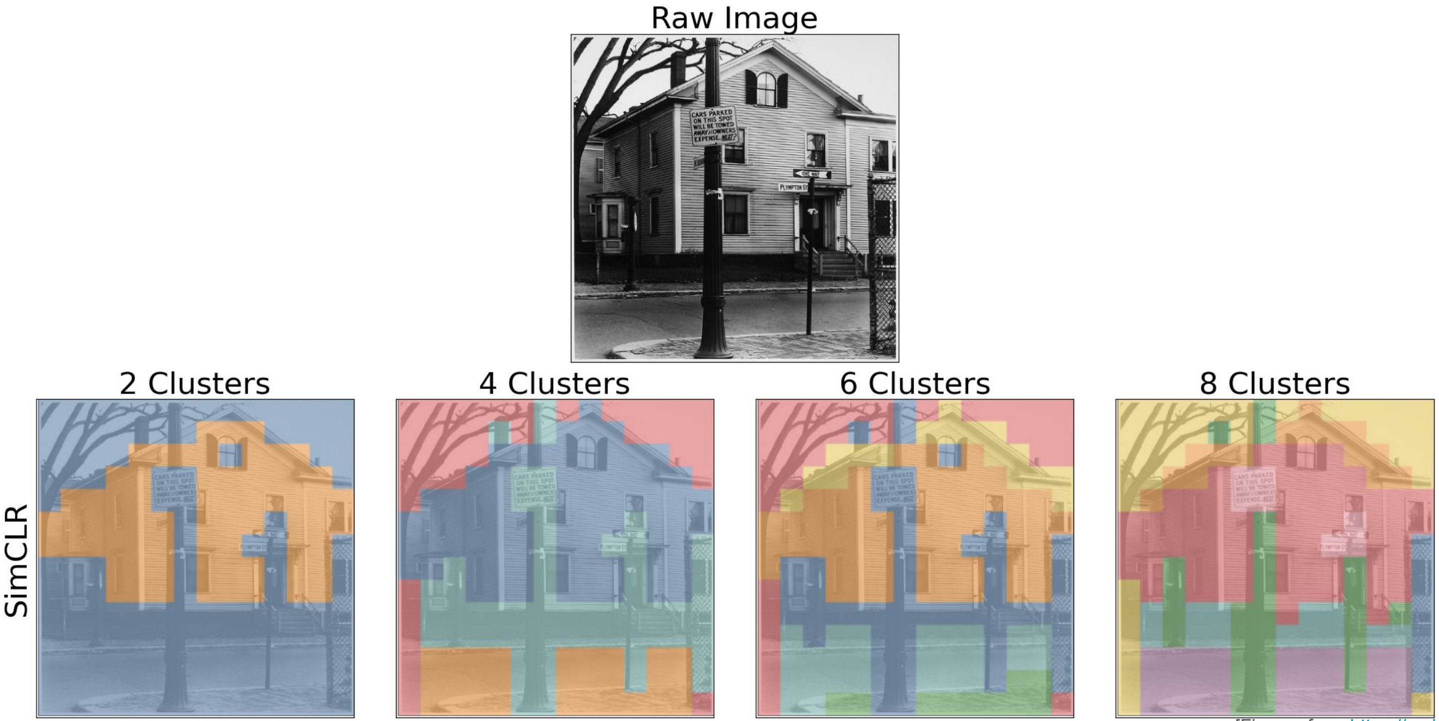
	Placing of digits	Number of digits (size 28 × 28)					
		1	2	4	8	12	16
Supervised	Random	99.5	99.5	99.3	99.4	98.9	98.3
	In-grid	99.5	99.6	99.5	99.3	98.6	92.4
SimCLR	Random	98.9	98.9	99.0	98.9	98.2	96.4
	In-grid	98.3	98.6	99.1	99.2	99.1	98.3

[Figures from https://arxiv.org/abs/2011.02803]

Google Research

Global feature vs local features

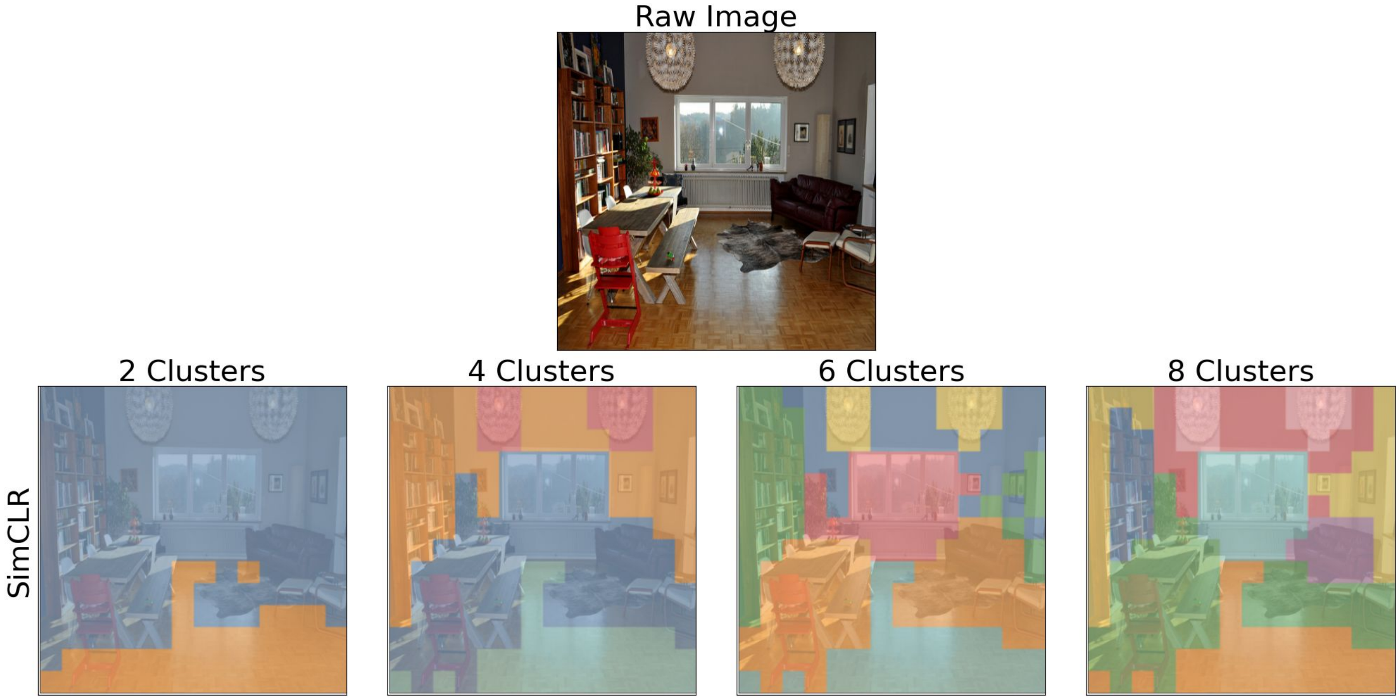
- Is instance-based contrastive learning able to learn local features?
    - Take a middle layer of SimCLR learned ResNet, do clustering



[Figures from https://arxiv.org/abs/2011.02803]

Global feature vs local features

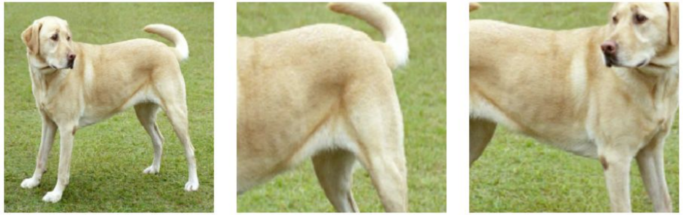
- SimCLR, despite trained with image-level loss, learns local features.
    - (although, local contrastive learning can still help)



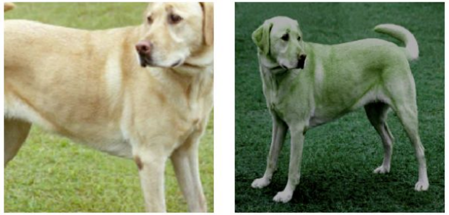
[Figures from https://arxiv.org/abs/2011.02803]

Feature suppression limits contrastive learning

- **Competing features** are different features shared between augmented views:



In common: dog class, color distribution, ..

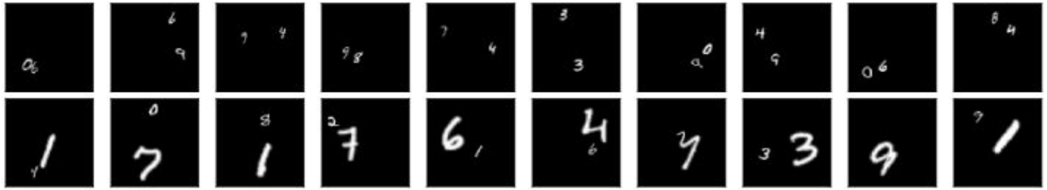


In common: dog class, ..

- Some features (e.g. color distribution) may suppress the learning of other set of features (e.g. object class)
- Can we **quantitatively** study the impact (suppression effect) of competing features?

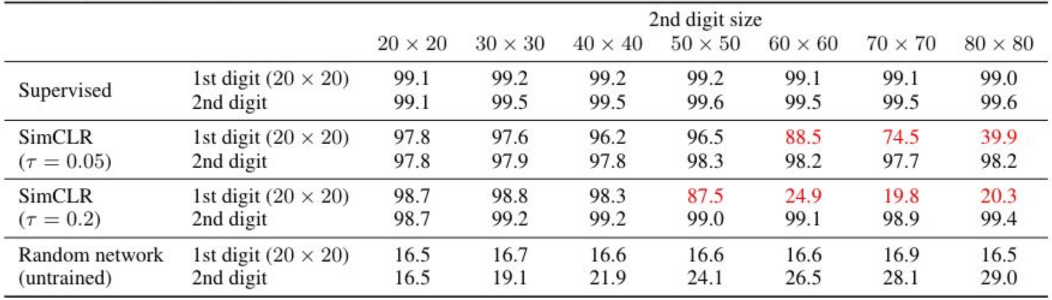
Larger objects suppress the learning of smaller objects

We place two MNIST digits of randomly on a canvas, and increase the size of one digit while keeping the other fixed.



(b) Two MNIST digits randomly placed on a shared canvas (of size 112 × 112). The two digits can have the same size (upper row) or different sizes (lower row), and digits of different sizes can be considered as competing features. We fix the size of one digit and vary the other.

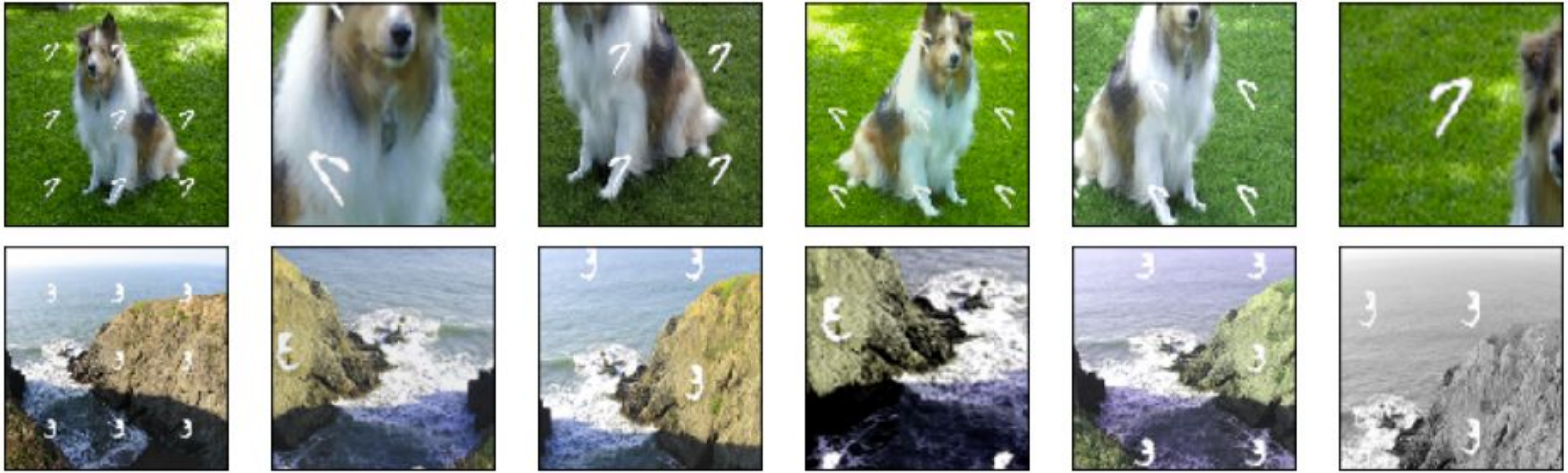
Table 2. Top-1 linear evaluation accuracy (%) for pretrained ResNet-18 on the MultiDigits dataset. We fix the size of 1st digit while increasing the size of the 2nd digit. For SimCLR, results are presented for two temperatures. Accuracies suffered from a significant drop when increasing 2nd digit size are red colored.



		2nd digit size						
		20 × 20	30 × 30	40 × 40	50 × 50	60 × 60	70 × 70	80 × 80
Supervised	1st digit (20 × 20)	99.1	99.2	99.2	99.2	99.1	99.1	99.0
	2nd digit	99.1	99.5	99.5	99.6	99.5	99.5	99.6
SimCLR ($\tau = 0.05$)	1st digit (20 × 20)	97.8	97.6	96.2	96.5	88.5	74.5	39.9
	2nd digit	97.8	97.9	97.8	98.3	98.2	97.7	98.2
SimCLR ($\tau = 0.2$)	1st digit (20 × 20)	98.7	98.8	98.3	87.5	24.9	19.8	20.3
	2nd digit	98.7	99.2	99.2	99.0	99.1	98.9	99.4
Random network (untrained)	1st digit (20 × 20)	16.5	16.7	16.6	16.6	16.6	16.9	16.5
	2nd digit	16.5	19.1	21.9	24.1	26.5	28.1	29.0

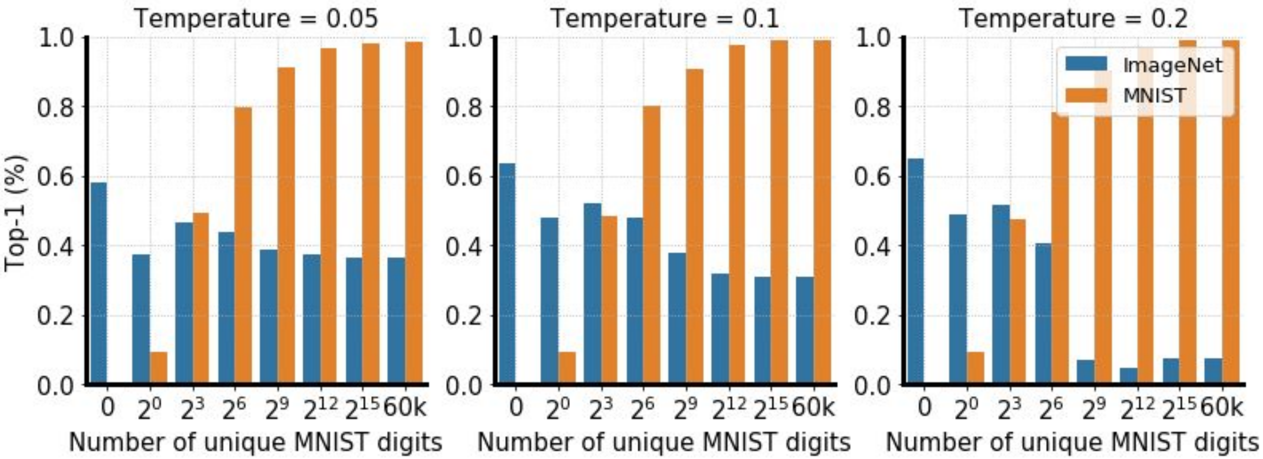
Digit features vs Object features

Adding competing features using channel addition: overlay a controlled number of unique MNIST digits on ImageNet images.

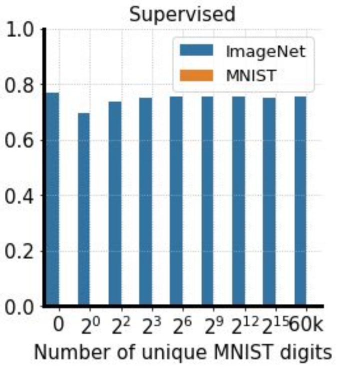


Easy-to-learn features suppress other features

Standard SimCLR couldn’t learn features that are good for linear evaluation on both MNIST digits and ImageNet classes.

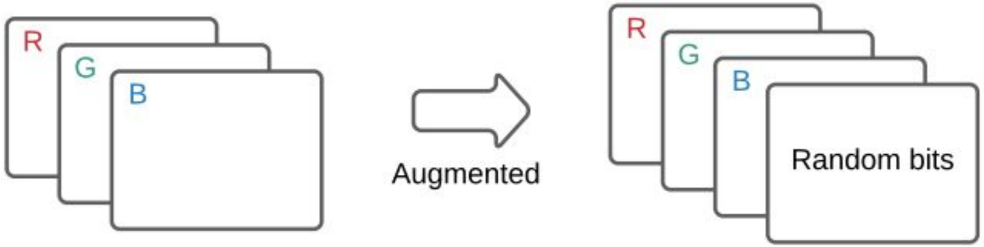


However, supervised learning of ImageNet classes is fine ⟶



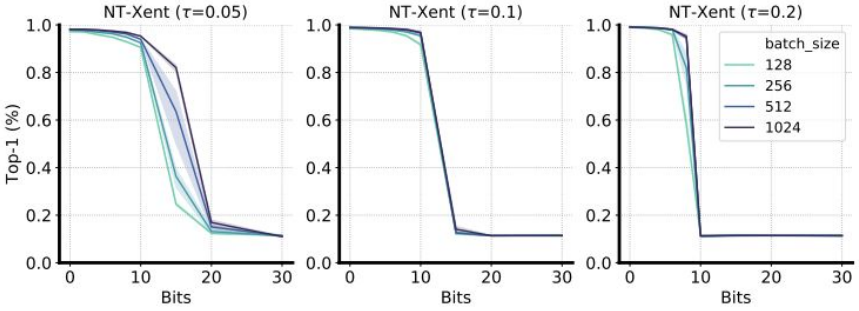
RGB features vs random bits

Adding competing features using channel concatenation: extra channels are controllable random bits that are shared between views.

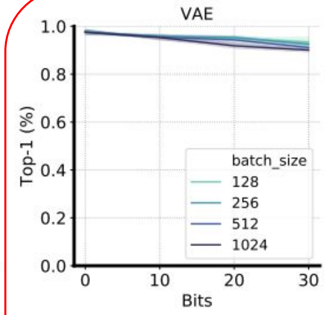


A few random bits suppress features in RGB

MNIST:

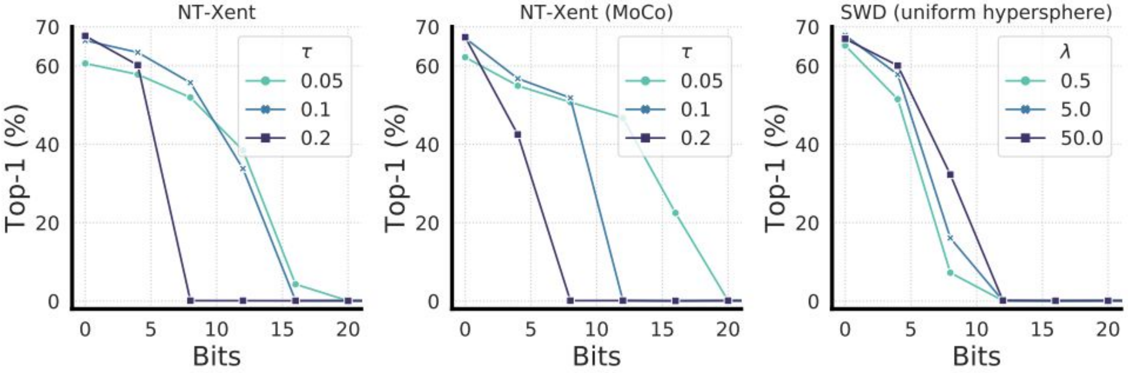


(a) MNIST.



(b) MNIST using VAE.

ImageNet:



(c) ImageNet

Final remarks

- Contrastive learning is a family of effective self-supervised learning methods, which can learn representations on par or better than supervised learning.
- Some key ideas in contrastive learning:
  - Define loss (e.g., max agreement) in learned abstract/latent space.
  - Augmentations as ways to define the prediction task.
  - Contrastive loss with negative examples to prevent collapse.
  - Other mechanism (e.g., momentum encoder, stop-gradient) to prevent collapse.
- Some open challenges for existing contrastive learning techniques
  - Feature suppression
  - Others, e.g., selection of data augmentations

Thank You!