Google Research

SimCLR: A Simple Framework for Contrastive Learning of Visual Representations

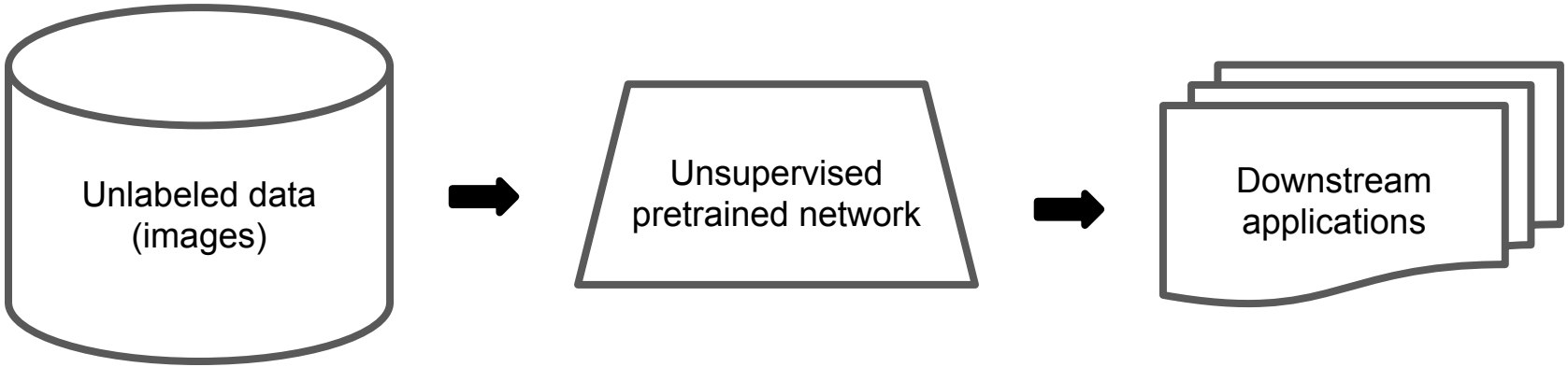
Ting Chen
Simon Kornblith
Mohammad Norouzi
Geoffrey Hinton

Google Research, Brain Team

Unsupervised representation learning

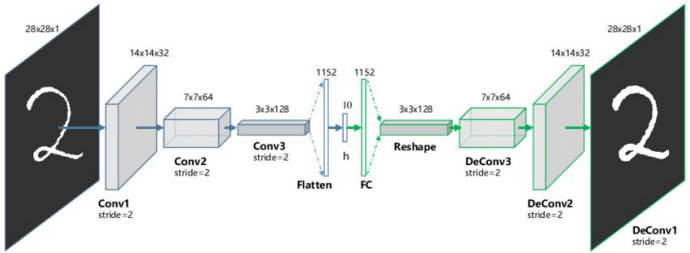
We tackle the problem of general visual representation learning from a set of unlabeled images.

After unsupervised learning, the learned model and image representations can be used for downstream applications.

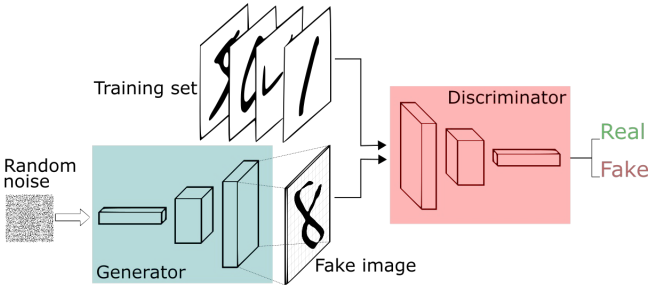


First category of unsupervised learning

- Generative modeling
  - Generate or otherwise model pixels in the input space
  - Pixel-level generation is computationally expensive
  - Generating images of high-fidelity may not be necessary for representation learning



Autoencoder

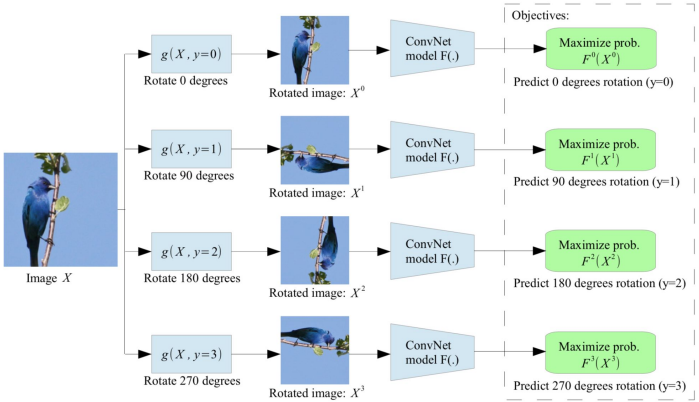


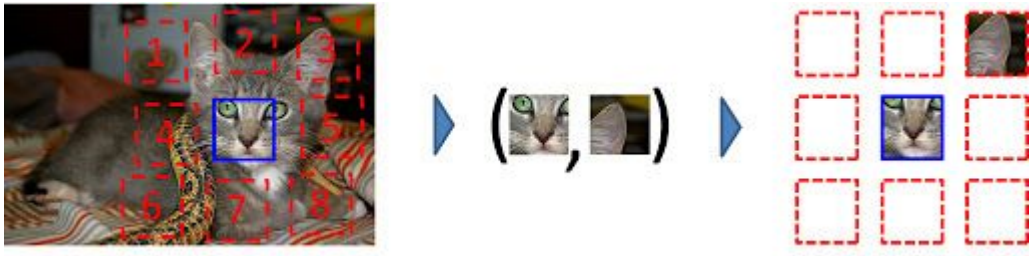
Generative Adversarial Nets

Image credit: Xifeng Guo, Thalles Silva.

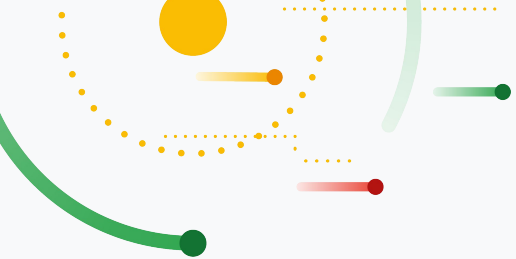
Second category of unsupervised learning

- Discriminative modeling
  - Train networks to perform *pretext tasks* where both the inputs and labels are derived from an unlabeled dataset.
  - Heuristic-based pretext tasks: rotation prediction, relative patch location prediction, colorization, solving jigsaw puzzle.
  - Many heuristics seem ad-hoc and may be limiting.





Images: [Gidaris et al 2018, Doersch et al 2015]



Google Research

Introducing SimCLR framework

The proposed SimCLR framework

A simple idea: maximizing the agreement of representations under data transformation, using a contrastive loss in the latent/feature space.

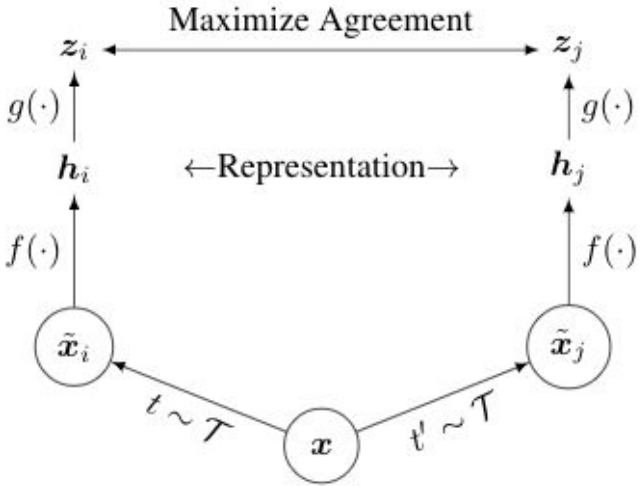
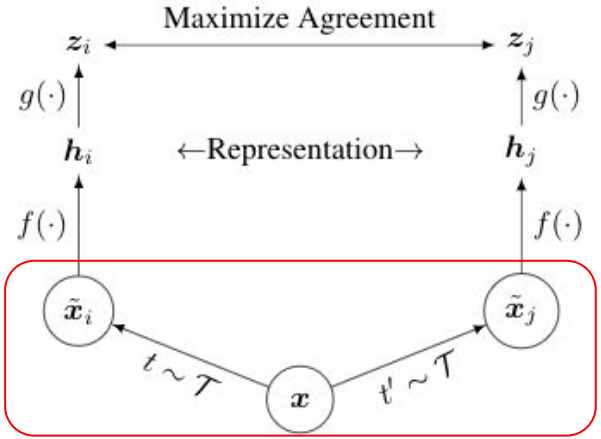


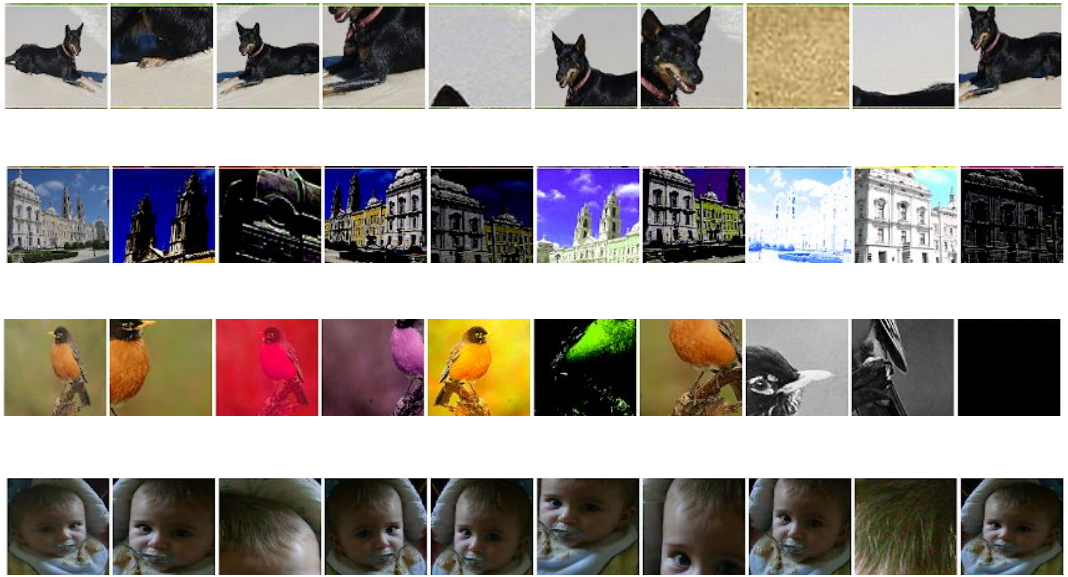
Figure 2. A framework for contrastive representation learning. Two separate stochastic data augmentations $t, t' \sim \mathcal{T}$ are applied to each example to obtain two correlated views. A base encoder network $f(\cdot)$ with a projection head $g(\cdot)$ is trained to maximize agreement in *latent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

The proposed SimCLR framework

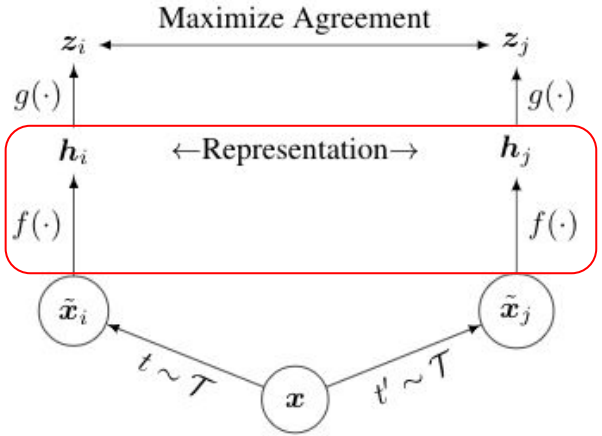


We use random crop and color distortion for augmentation.

Examples of augmentation applied to the left most images:

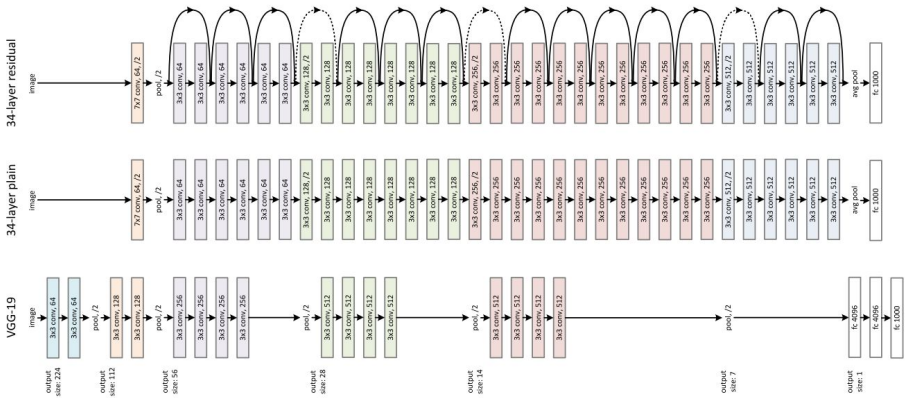


The proposed SimCLR framework

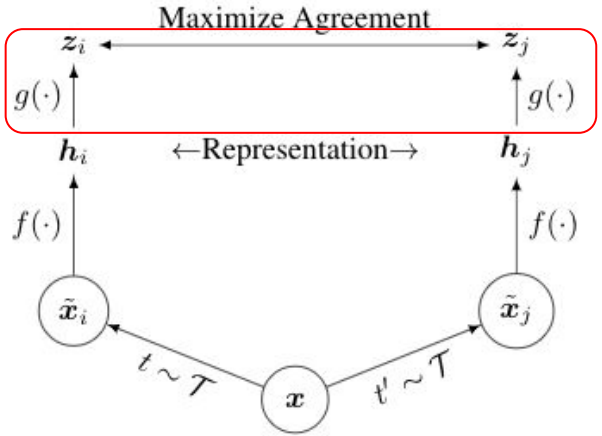


f(x) is the base network that computes internal representation.

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

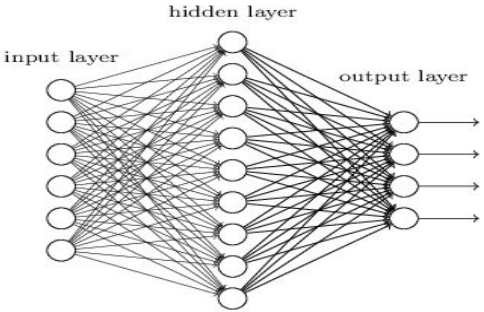


The proposed SimCLR framework

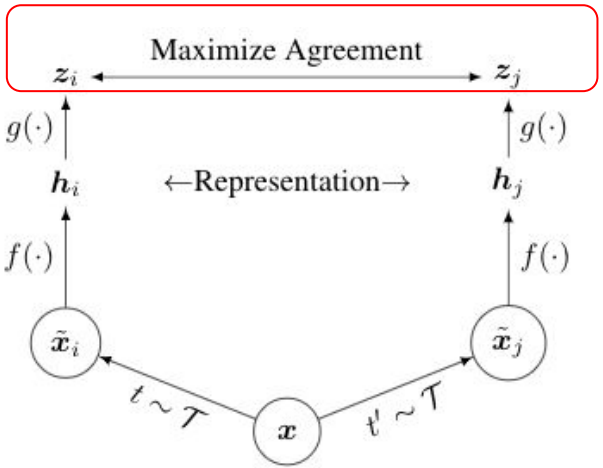


g(h) is a projection network that project representation to a latent space.

We use a 2-layer non-linear MLP (fully connected net).



The proposed SimCLR framework



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:

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

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

Google Research

SimCLR pseudo code and illustration

Algorithm 1 SimCLR's main learning algorithm.

input: batch size N, temperature τ, form of f, g, $\mathcal{T}$.
for sampled mini-batch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ **do**
  for all $k \in \{1, \dots, N\}$ **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 $i \in \{1, \dots, 2N\}$ and $j \in \{1, \dots, 2N\}$ **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

Important implementation details

- We trained the model with varied batch sizes (256-8192).
  - No memory bank, as a batch size of 8K gives us 16K negatives per positive pair.
  - Typically, an intermediate batch size (e.g. 1k, 2k) could work well.
- To stabilize training for large bsz, we use LARS optimizer.
  - Scale learning rate dynamically according to grad norm.
- To avoid shortcut, we use global BN.
  - Compute BN statistics over all cores.

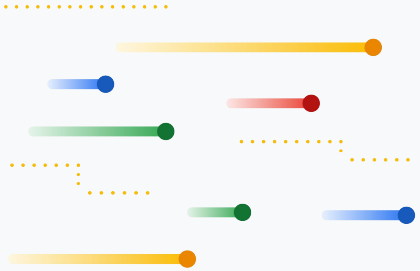
Understand the learned representations & essentials

Main dataset:

- ImageNet
- (Also works on CIFAR-10 & MNIST)

Three evaluation protocols

- Linear classifier trained on learned features
  - What we used for ablations
- Fine-tune the model on few labels
- Transfer learning by fine-tuning on other datasets



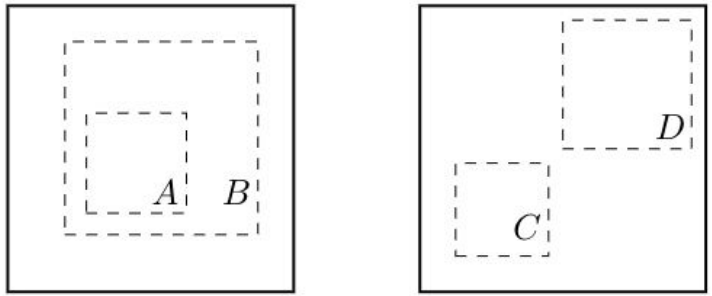
Google Research

Data Augmentation for Contrastive Representation Learning

Data augmentation defines predictive tasks

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.

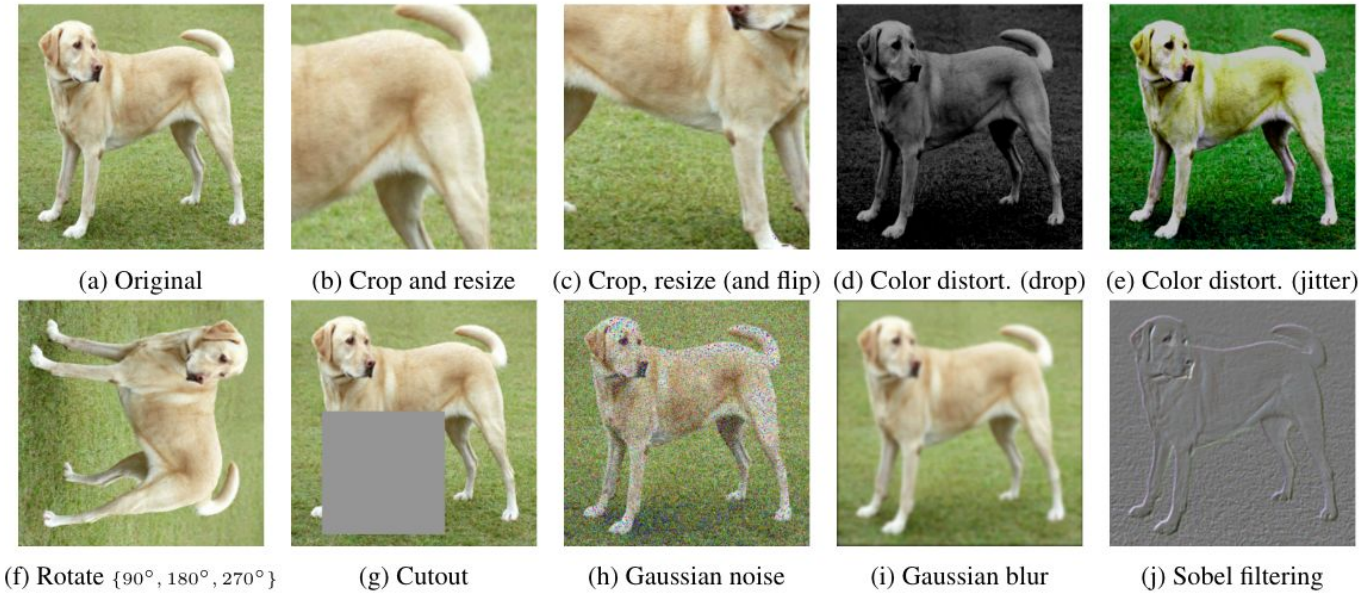


(a) Global and local views. (b) Adjacent views.

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.

We study a set of transformations...

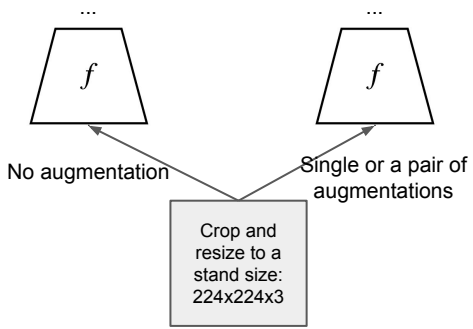
Systematically study a set of augmentation



* 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.

Studying single or a pair of augmentations

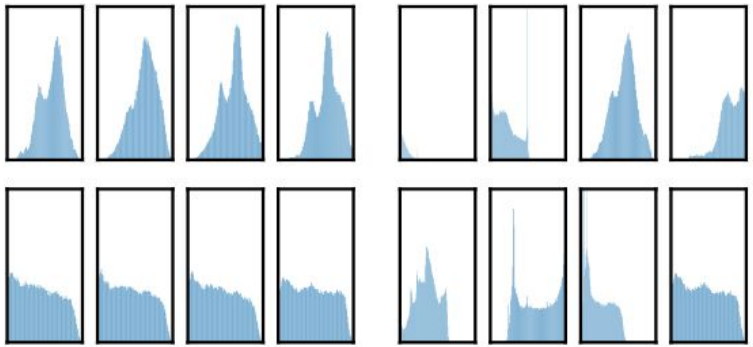
- ImageNet images are of different resolutions, so random crops are typically applied.
- To remove co-founding
  - First random crop an image and resize to a standard resolution.
  - Then apply a single or a pair of augmentations on one branch, while keeping the other as identity mapping.
  - This is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



Composition of augmentations are crucial

Composition of crop and color stands out!

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.

Contrastive learning needs stronger data/color augmentation than supervised learning

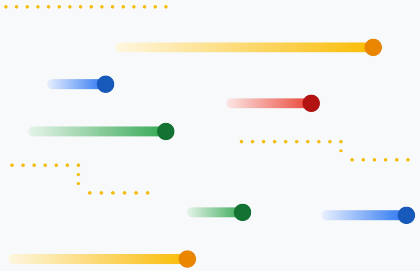
Simply combining crop + color (+ Blur) beats searched AutoAugmentation, a searched policy on supervised learning!

We should rethink data augmentation for self-supervised learning!

Strength	1/8	1/4	1/2	1	1 (+Blur)	AutoAug
Unsup.	59.6	61.0	62.6	63.2	64.5	61.1
Sup.	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.

[5]Supervised models are trained for 90 epochs; longer training improves performance of stronger augmentation by $\sim 0.5\%$.



Google Research

Encoder and Projection Head

Unsupervised contrastive learning benefits (more) from bigger models

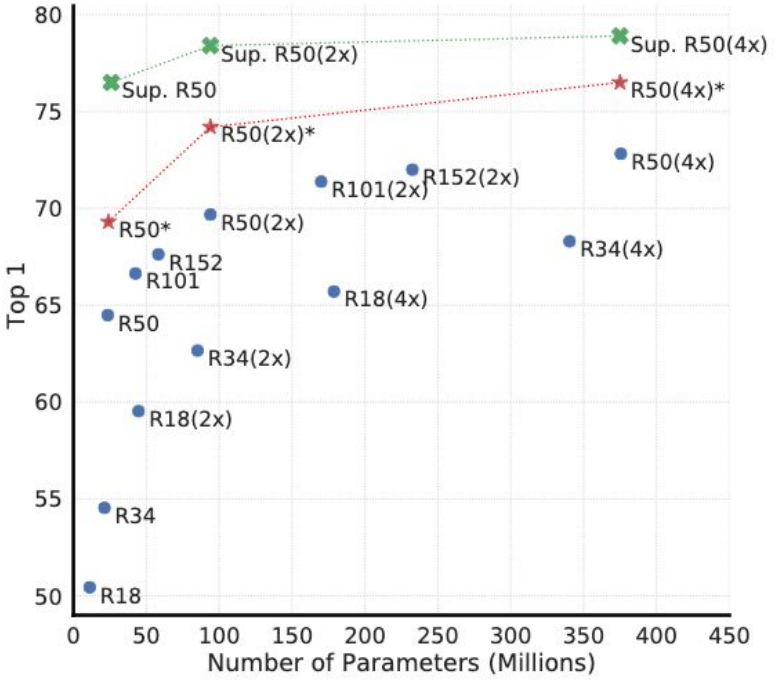


Figure 7. Linear evaluation of models with varied depth and width. Models in blue dots are ours trained for 100 epochs, models in red stars are ours trained for 1000 epochs, and models in green crosses are supervised ResNets (He et al., 2016).[7]

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

We compare three projection head g(.) (after average pooling of ResNet):

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

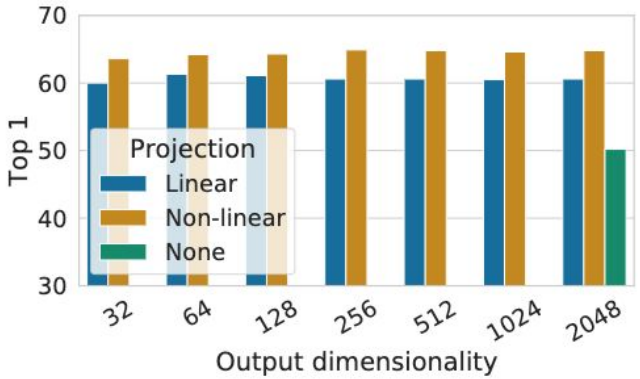


Figure 8. Linear evaluation of pretraining with different projection heads. The dimension of $\boldsymbol{h}$ (before projection) is 2048.

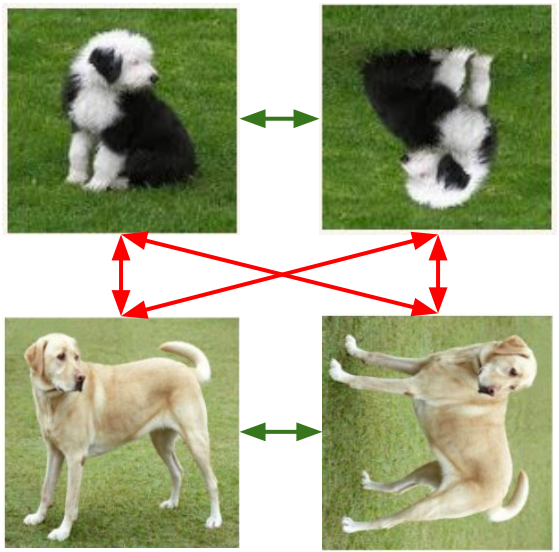
Even when non-linear projection is used, the layer before the projection head,h,is still much better (>10%) than the layer after,z=g(h).

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

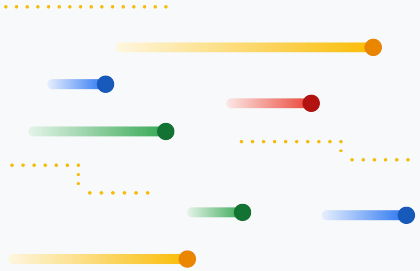
To understand why this happens, we measure information in ***h*** and ***z=g(h)***

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.



Google Research

Loss Function and Batch Size

Normalized cross entropy loss with adjustable temperature works better than alternatives

Name	Negative loss function	Gradient w.r.t. $\boldsymbol{u}$
NT-Xent	$\boldsymbol{u}^T\boldsymbol{v}^+/\tau - \log \sum_{\boldsymbol{v}\in\{\boldsymbol{v}^+,\boldsymbol{v}^-\}} \exp(\boldsymbol{u}^T\boldsymbol{v}/\tau)$	$(1 - \frac{\exp(\boldsymbol{u}^T\boldsymbol{v}^+/\tau)}{Z(\boldsymbol{u})})/\tau\boldsymbol{v}^+ - \sum_{\boldsymbol{v}^-} \frac{\exp(\boldsymbol{u}^T\boldsymbol{v}^-/\tau)}{Z(\boldsymbol{u})}/\tau\boldsymbol{v}^-$
NT-Logistic	$\log \sigma(\boldsymbol{u}^T\boldsymbol{v}^+/\tau) + \log \sigma(-\boldsymbol{u}^T\boldsymbol{v}^-/\tau)$	$(\sigma(-\boldsymbol{u}^T\boldsymbol{v}^+/\tau))/\tau\boldsymbol{v}^+ - \sigma(\boldsymbol{u}^T\boldsymbol{v}^-/\tau)/\tau\boldsymbol{v}^-$
Margin Triplet	$-\max(\boldsymbol{u}^T\boldsymbol{v}^- - \boldsymbol{u}^T\boldsymbol{v}^+ + m, 0)$	$\boldsymbol{v}^+ - \boldsymbol{v}^-$ if $\boldsymbol{u}^T\boldsymbol{v}^+ - \boldsymbol{u}^T\boldsymbol{v}^- < m$ else $\mathbf{0}$

Table 2. Negative loss functions and their gradients. All input vectors, i.e. $\boldsymbol{u}, \boldsymbol{v}^+, \boldsymbol{v}^-$, are ℓ_2 normalized. NT-Xent is an abbreviation for "Normalized Temperature-scaled Cross Entropy". Different loss functions impose different weightings of positive and negative examples.

Margin	NT-Logi.	Margin (sh)	NT-Logi.(sh)	NT-Xent
50.9	51.6	57.5	57.9	63.9

Table 4. Linear evaluation (top 1) for models trained with different loss functions. "sh" means using semi-hard negative mining.

NT-Xent loss needs N and T

We compare variants of NT-Xent loss

- 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.

Contrastive learning benefits from larger batch sizes and longer training

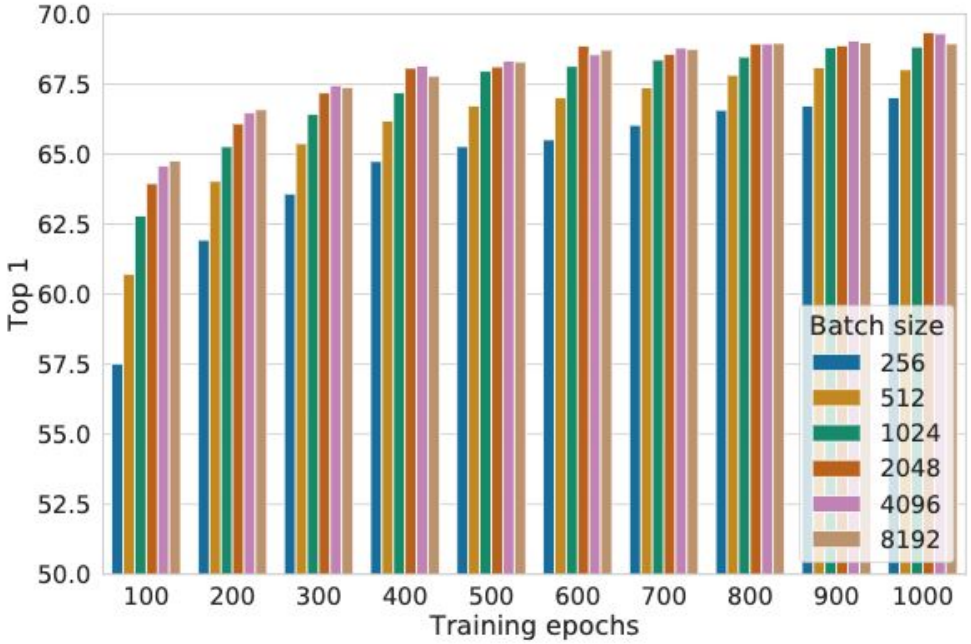
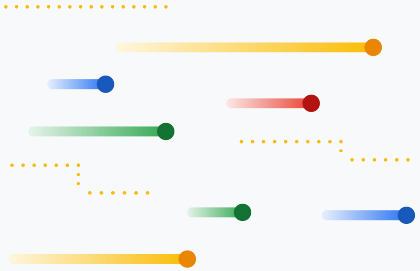


Figure 9. Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch.



Google Research

Comparison Against State-of-The-Art

Baselines

We mainly compare to existing work on self-supervised visual representation learning, including those that are also based on contrastive learning, e.g. Exemplar, InstDist, CPC, DIM, AMDIM, CMC, MoCo, PIRL, ...

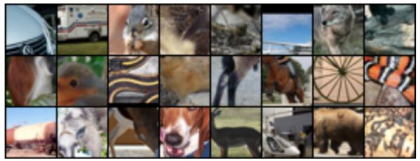


Figure 1: Exemplary patches sampled from

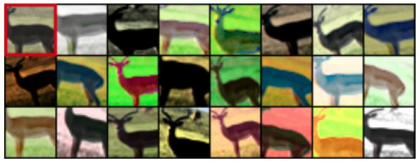
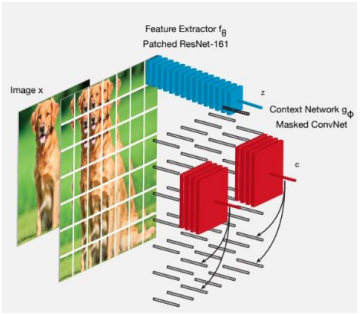
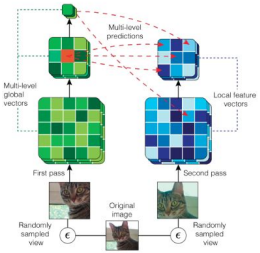
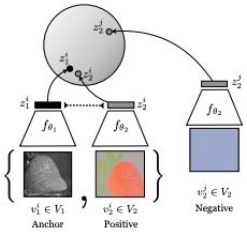
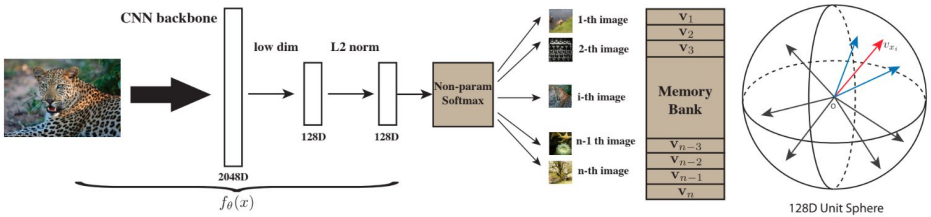


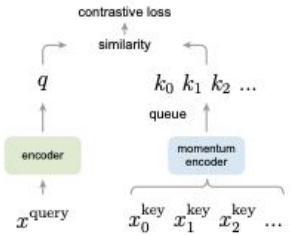
Figure 2: Several random transformations

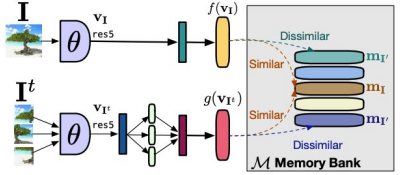












Linear evaluation

7% relative improvement over previous SOTA (cpc v2), matching fully-supervised ResNet-50.

Method	Architecture	Param.	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
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
Ours	ResNet-50 (2×)	94	74.2	92.0
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.

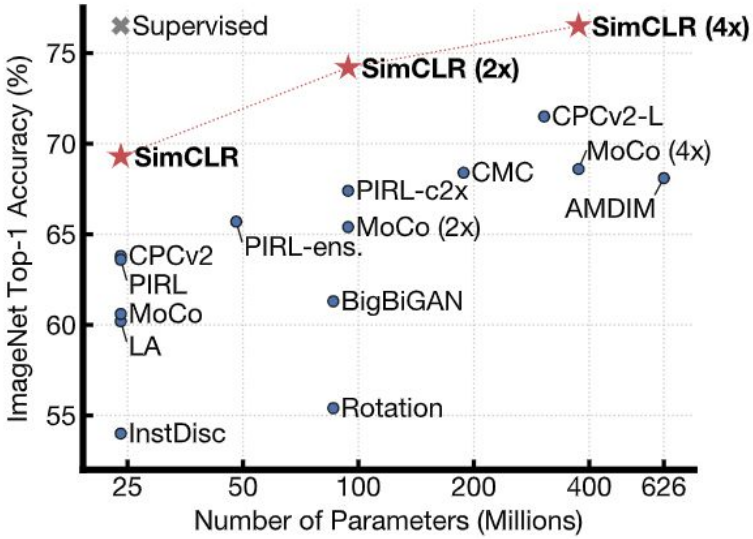


Figure 1. ImageNet top-1 accuracy of linear classifiers trained on representations learned with different self-supervised methods (pretrained on ImageNet). Our method, SimCLR, is shown in bold.

Semi-supervised learning

10% relative improvement over previous SOTA (cpc v2), outperforms AlexNet with 100X fewer labels.

Method	Architecture	Label fraction (Top 5) 1%	10%
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

When fine-tuned, SimCLR significantly outperforms the supervised baseline on **5** datasets, whereas the supervised baseline is superior on only **2***. On the remaining **5** datasets, the models are statistically tied.

	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 $(4\times)$ 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.

Conclusion

- SimCLR is a simple yet effective self-supervised learning framework, advancing state-of-the-art by a large margin.
- The superior performance of SimCLR is not due to any *single* design choice, but *a combination of* design choices.
- Our studies reveal several important factors that enable effective representation learning, which could help future research.

Code & checkpoints available in github.com/google-research/simclr.