SimCLR: A Simple Framework for Contrastive Learning of Visual Representations

Google Research, Brain Team

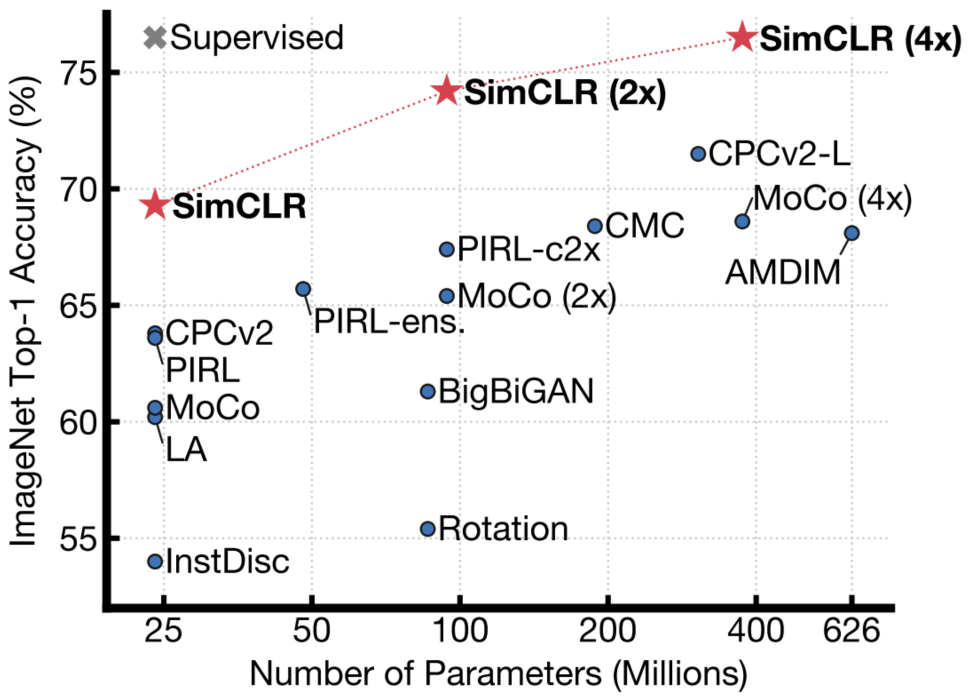
Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton

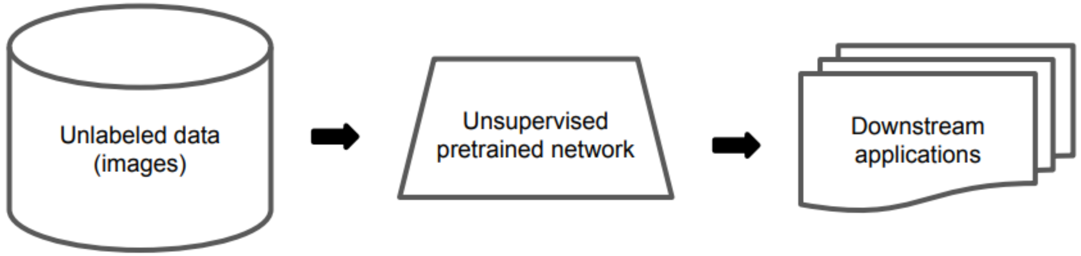
Presenter: Shovito Barua Soumma

Date: November 1, 2023

What is SimCLR?

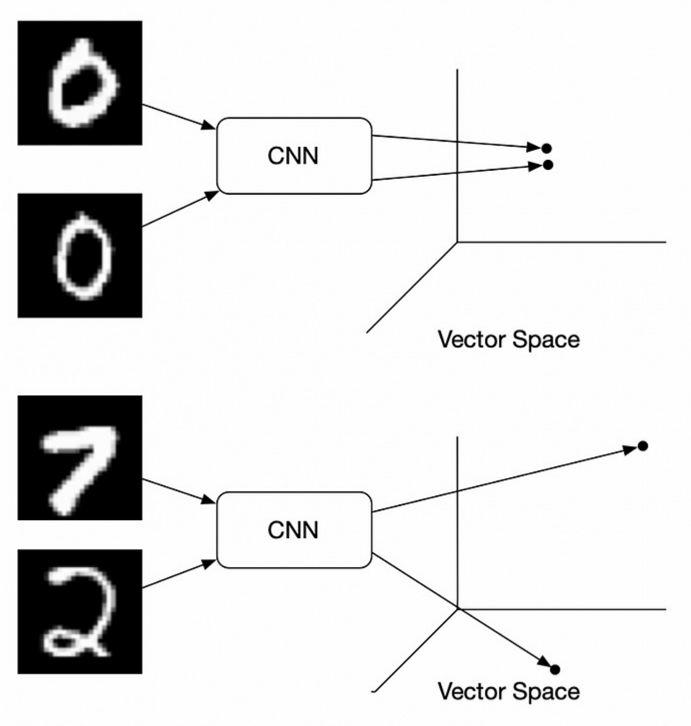
- learns generic representations of images on an unlabeled dataset
- fine-tuned with a small amount of labeled images to achieve good performance



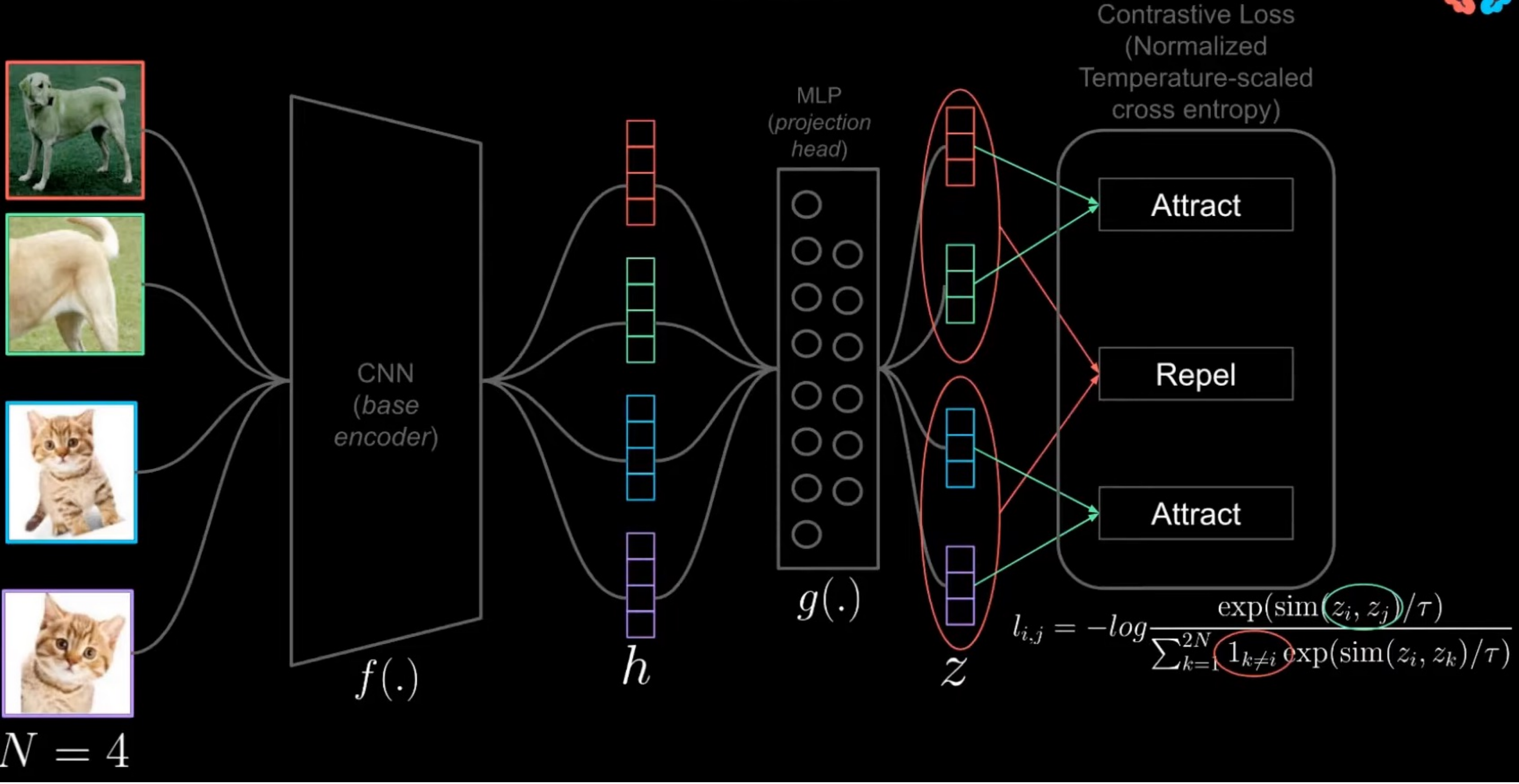


Contrastive Loss

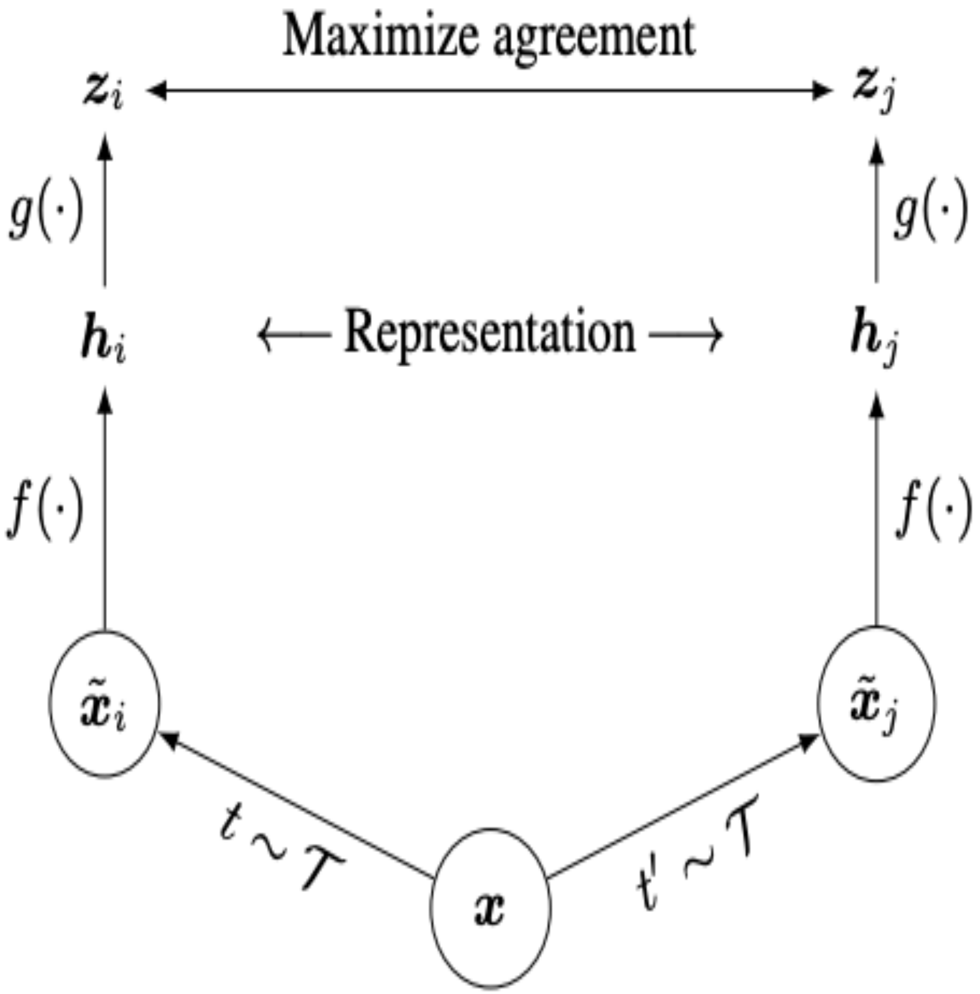
- Same class --> similar embeddings
- Different class --> dissimilar embeddings



Method



$$l_{i,j} = -log \frac{\exp(\mathrm{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} 1_{k \neq i} \exp(\mathrm{sim}(z_i, z_k)/\tau)}$$


Maximize agreement
z_i
z_j
$g(\cdot)$
$g(\cdot)$
h_i
$\longleftarrow$ Representation $\longrightarrow$
h_j
$f(\cdot)$
$f(\cdot)$
$\tilde{x}_i$
$\tilde{x}_j$
$t \sim \mathcal{T}$
$t' \sim \mathcal{T}$
x

Loss function in SimCLR

- For a positive pair (i, j):
- $Cosine\ Similarity = \frac{A.B}{\|A\| \times \|B\|}$

Softmax function, $S(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}$

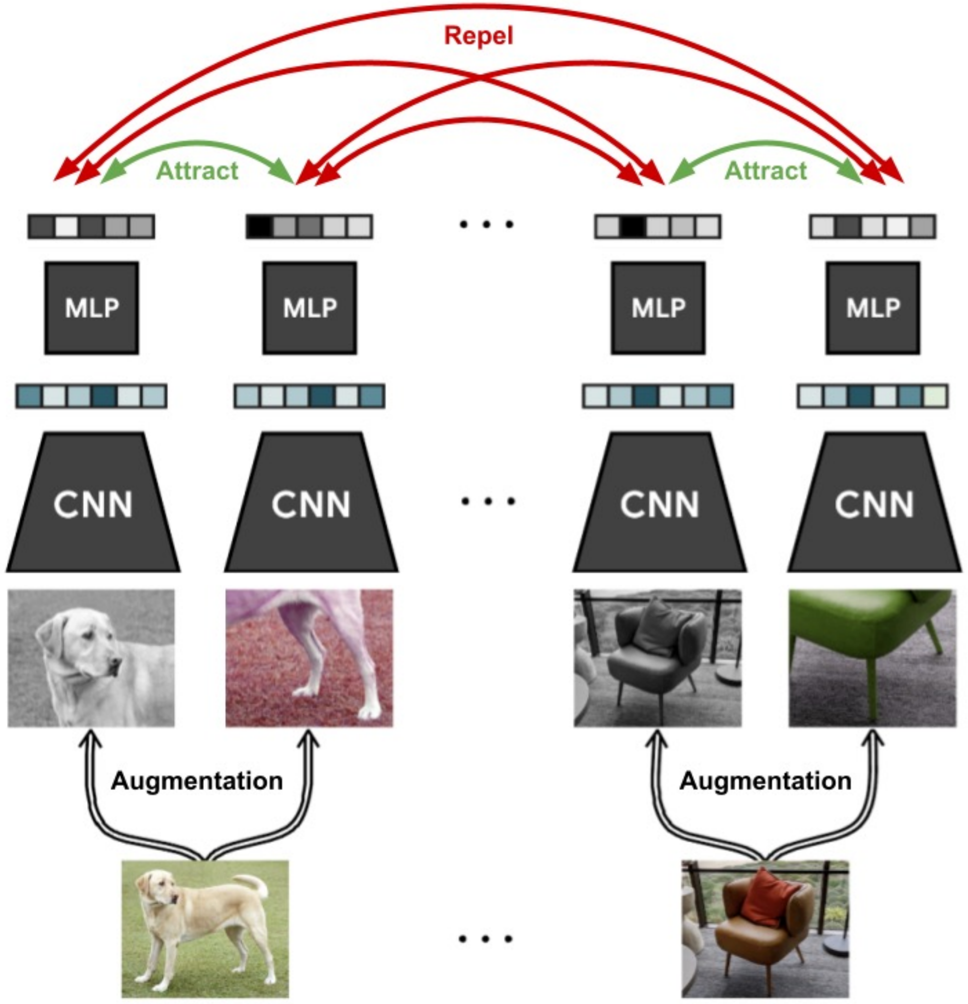
- $$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}, \quad (1)$$

  where $\mathbb{1}_{[k \neq i]} \in \{0, 1\}$ is an indicator function evaluating to 1 iff $k \neq i$ and τ denotes a temperature parameter. The final loss is computed across all positive pairs, both (i, j) and (j, i), in a mini-batch. This loss has been used in

Sampling 2N data points from N data

D1	**D2**	**C1**	**C2**

$\widetilde{D1}$	$\widetilde{D'1}$	$\widetilde{D2}$	$\widetilde{D'2}$	$\widetilde{C1}$	$\widetilde{C'1}$	$\widetilde{C2}$	$\widetilde{C'2}$



Key Findings

Data Augmentation:

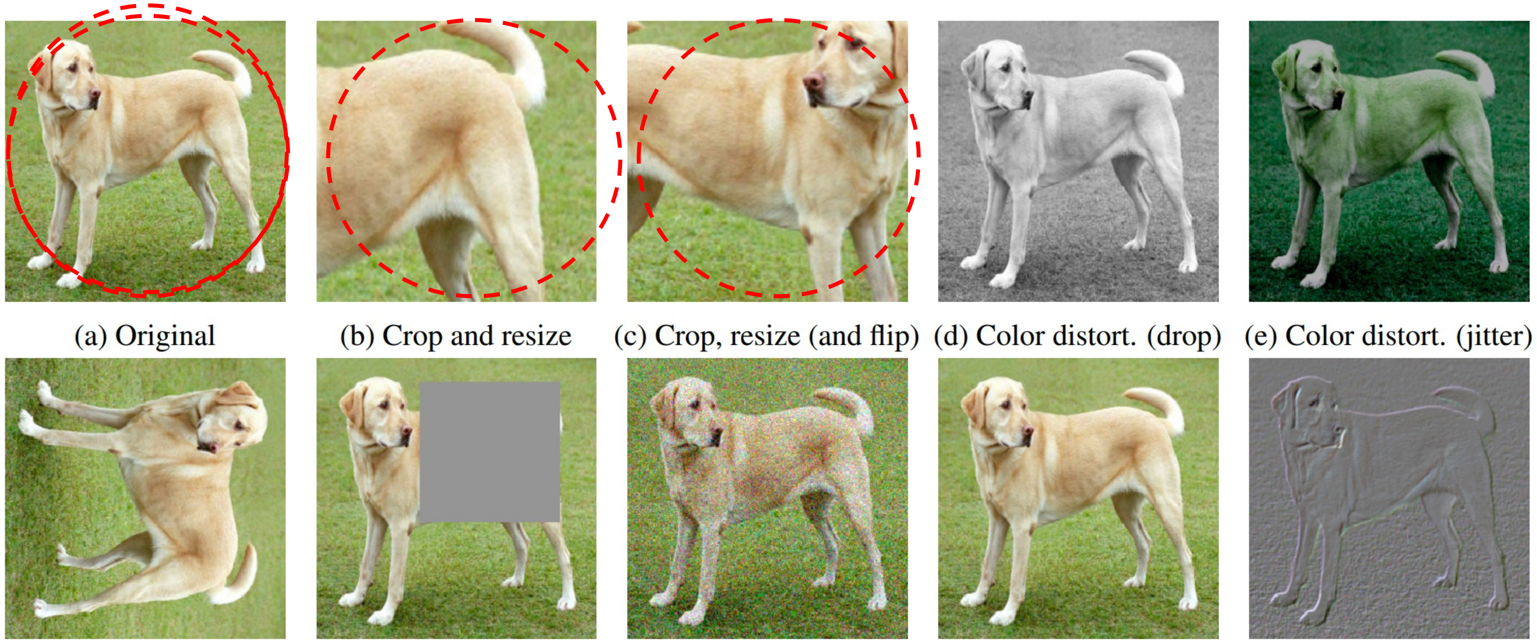
- Previously we need to change the architecture, Now data augmentation (Random Cropping) is enough to learn the contrastive representation

Large Batch Size

More Training Epoch

Wider Network

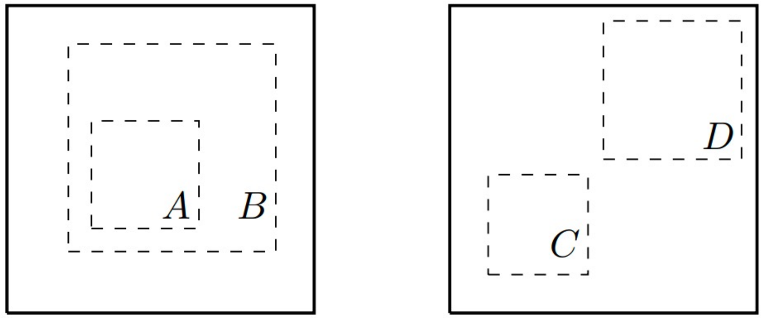
Augmentation



(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

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)

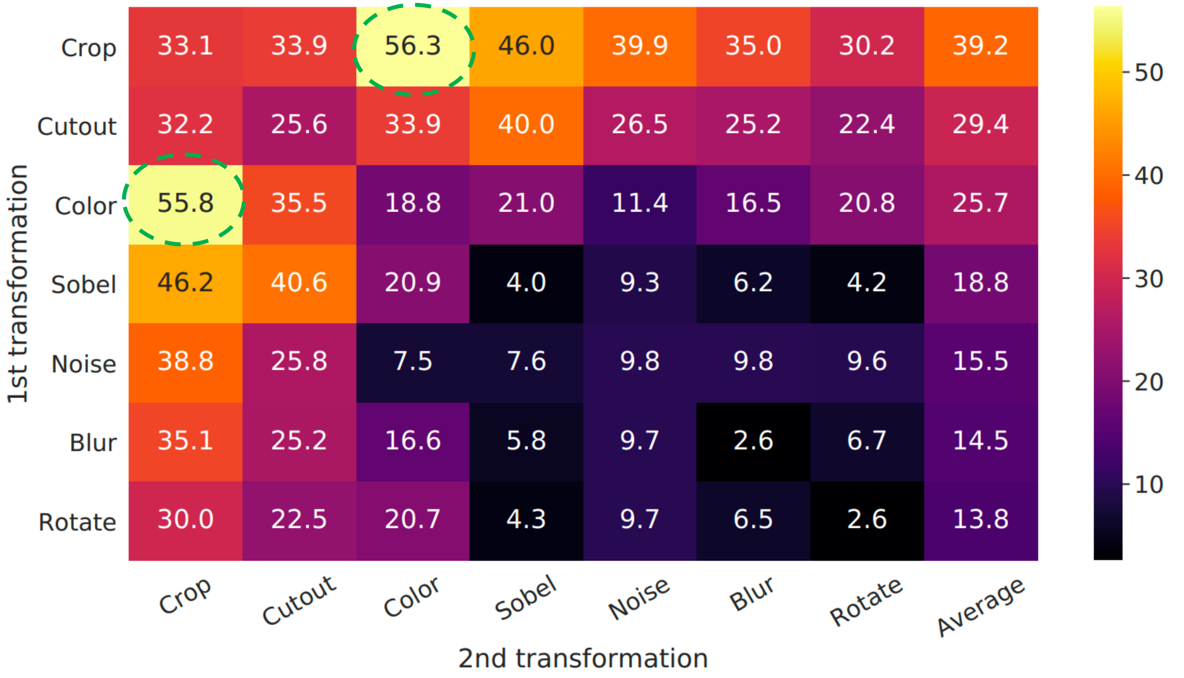


(a) Global and local views. (b) Adjacent views.

Figure 3. Solid rectangles are images, dashed rectangles are random crops. By randomly cropping images, we sample contrastive prediction tasks that include global to local view ($B \rightarrow A$) or adjacent view ($D \rightarrow C$) prediction.

Composition of Augmentation

- Apply two series of augmentation (one after another)
  - Significantly improves the quality of representation
- Random Cropping + Color Jitter stand out



Composition of Augmentation

- How Much Augmenation should we do?
- What will be the strength for color distortion?
- Very High!!! (Color Distortion =1)

Methods	Color distortion strength					AutoAug
	1/8	1/4	1/2	1	1 (+Blur)	
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

Model Size & Projection Head

- Increasing the size → increases the accuracy significantly
- a nonlinear projection is better than a linear projection (+3%), and much better than no projection

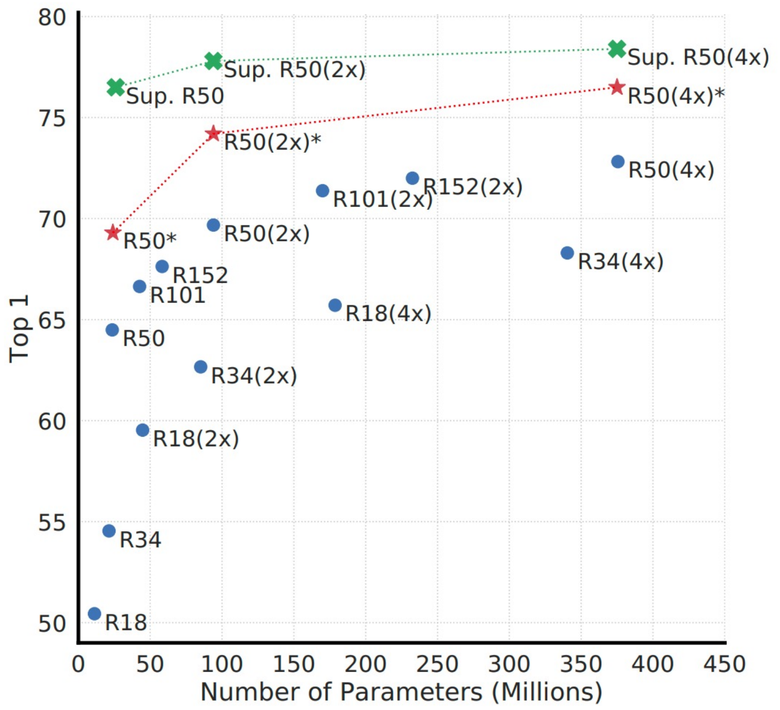


Figure 7. Linear evaluation of models with varied depth and width. Models in blue dots are ours trained for 100 epochs, models in red stars are ours trained for 1000 epochs, and models in green crosses are supervised ResNets trained for 90 epochs[7] (He et al., 2016).

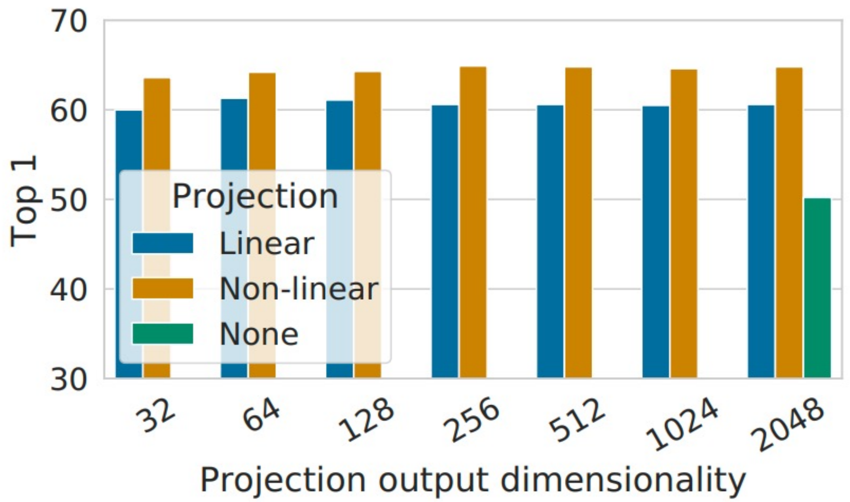
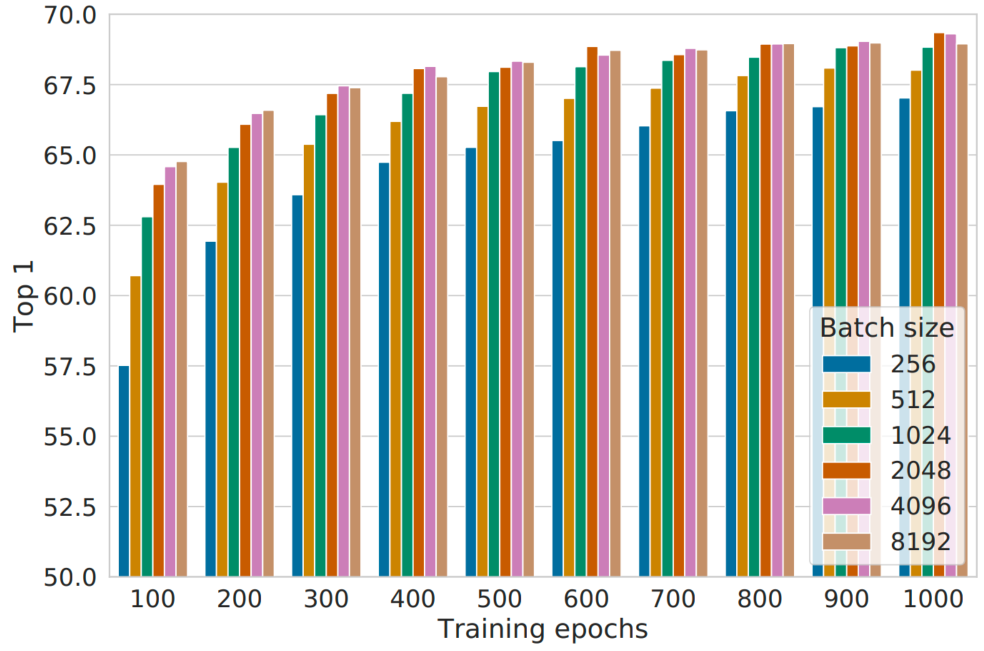


Figure 8. Linear evaluation of representations with different projection heads $g(\cdot)$ and various dimensions of $\boldsymbol{z} = g(\boldsymbol{h})$. The representation $\boldsymbol{h}$ (before projection) is 2048-dimensional here.

Batch Size & Epoch

- Larger Batch
- Longer Epoch



Summary of SimCLR Framework!

- Projection head is important to get good representation
- Random crop, flip and color jitter are best
- Stronger augmentation
- Longer Epoch with Large batch size → Many GPUs