A Simple Framework for Contrastive Learning of Visual Representations

Yien Xu, Lichengxi Huang

1. Motivation

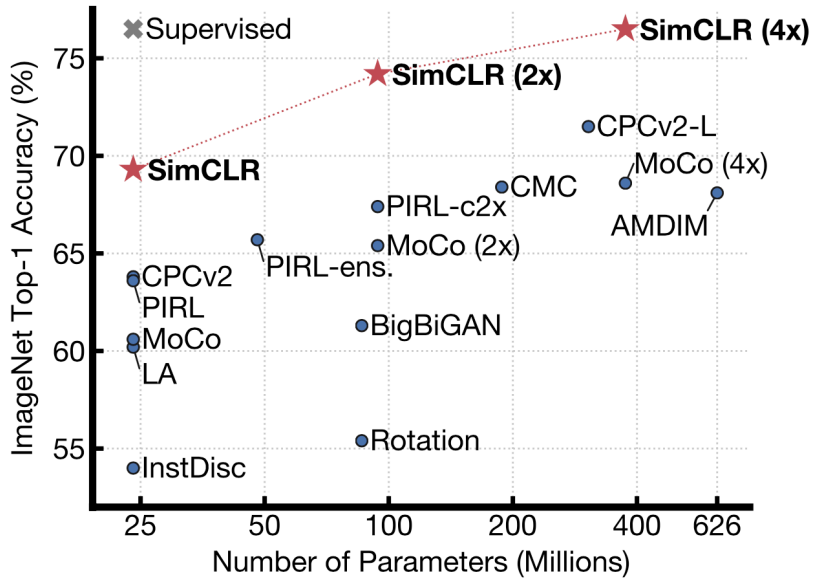
Learning effective visual representations without human supervision is a longstanding problem and most mainstream approaches fall into one of the two classes: generative and discriminative.

Generative approaches:

+ Learns to generate or model pixels in the input space
- Pixel level generation computationally expensive
- May not be necessary for representation learning

Discriminative approach:

+ learn representations using objective functions
+ Train networks to perform pretext tasks
+ Both the inputs and labels are derived from an unlabeled dataset



This figure shows the ImageNet Top-1 accuracy, where we can see SimCLR outperforms other state-of-the-art approaches.

Here we list some of the experiment results of SimCLR:

Evaluated on ImageNet

- SimCLR achieves 76.5% top-1 accuracy
- 7% relative improvement over previous state-of-the-art

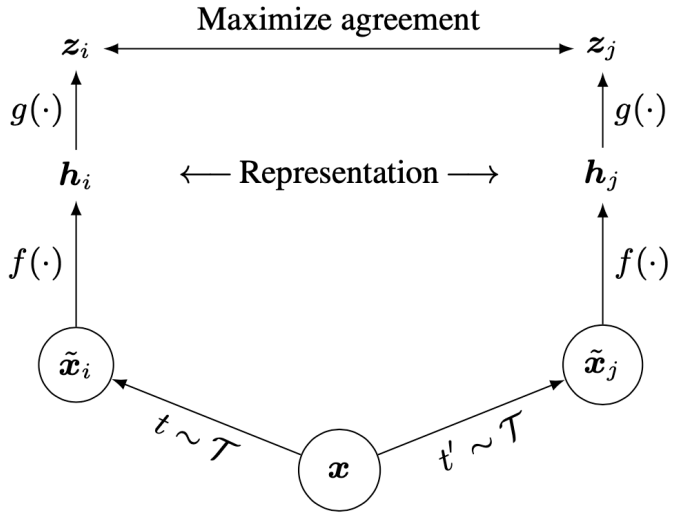
When fine-tuned with only 1% of the ImageNet labels

- SimCLR achieves 85.8% top-5 accuracy
- 10% relative improvement over previous state-of-the-art

When fine-tuned on other natural image classification datasets

- SimCLR performs on par with or better than a strong supervised baseline on 10 out of 12 datasets

2. Framework



2.1. Data Augmentation

The data augmentation step is a very first step of SimCLR. It is a stochastic data augmentation module that transforms any given data example randomly, resulting in two correlated views of the same example denoted as X_i and X_j.

Three augmentations applied sequentially

- Random cropping followed by a resizing
- Random color distortions
- Random Gaussian blur

Why is the wise data augmentation necessary?

- Data augmentation defines predictive tasks.

Many existing approaches define contrastive prediction tasks by changing the architecture:

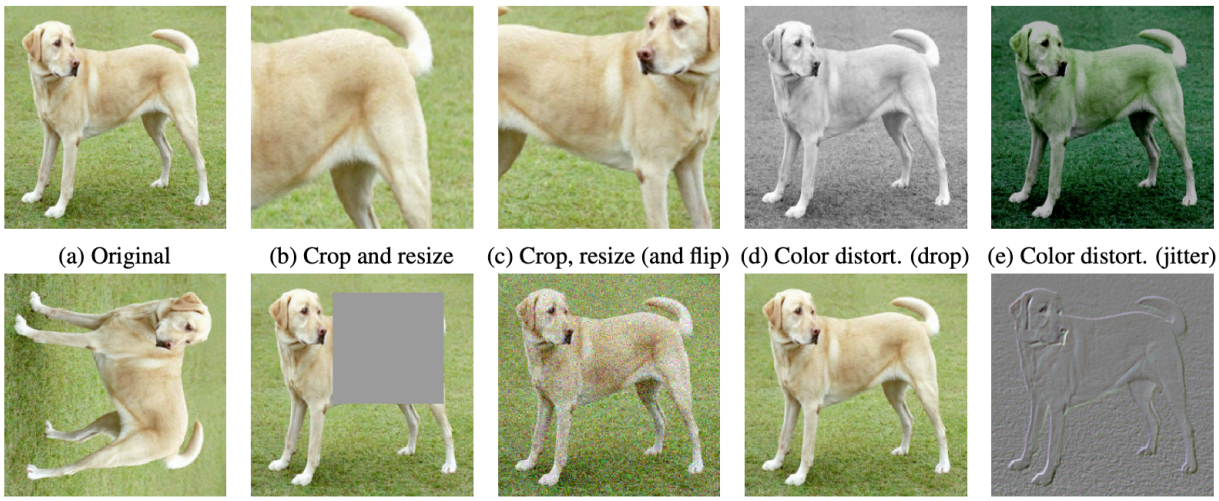
- global-to-local view prediction via constraining the receptive field in the network architecture
- neighboring view prediction via a fixed image splitting procedure and a context aggregation network

This complexity can be avoided by performing simple random cropping with resizing of target images which creates a family of predictive tasks.

Broader contrastive prediction tasks can be defined

- By extending the family of augmentations
- By composing them stochastically

Different data augmentation operators studied:



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

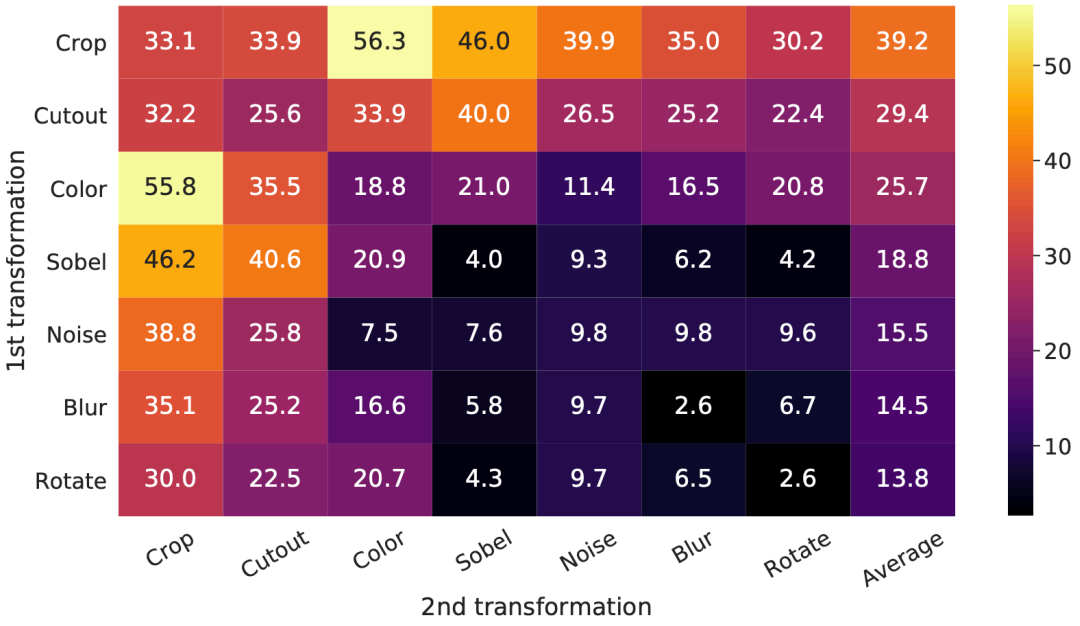
(f) Rotate $\{90°, 180°, 270°\}$ (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

To understand the effects of individual data augmentations and the importance of augmentation composition, the authors investigated performance of the framework when applying augmentations individually or in pairs.

Since ImageNet images are of different sizes, we always apply crop and resize images.

Consider asymmetric data transformation setting:

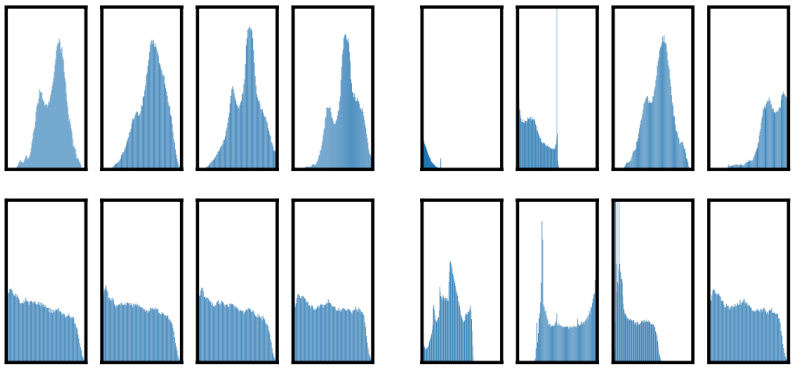
- first randomly crop images and resize them to the same resolution and then apply the target transformations only to one branch of the framework
- leave the other branch as the identity function



No single transformation suffices to learn good representations because the diagonal numbers are pretty low. When we compose augmentations, the quality of the representation improved dramatically.

Compositions of augmentation stand out:

- random cropping
- random color distortion



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

It is really critical to compose cropping with color distortion in order to learn generalizable features.

Without color distortion

- Random cropping of images shares similar color distributions
- Neural networks may take this shortcut

With color distortion

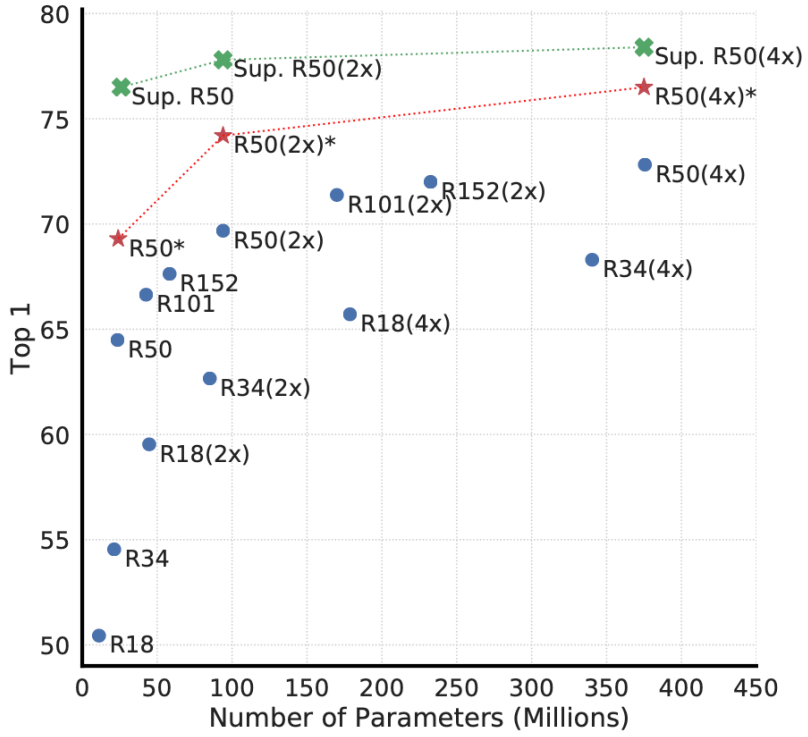
- Suffices to distinguish images

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

The above table shows unsupervised contrastive learning benefits from Stronger (color) data augmentation than supervised learning.

2.2. Base Encoder

A new network extracts the representation vectors from the augmented data examples.
The framework allows various choices of the network architectures without any constraints
SimCLR chooses ResNet: $h_i = f(\tilde{\boldsymbol{x}}_i) = ResNet(\tilde{\boldsymbol{x}}_i)$



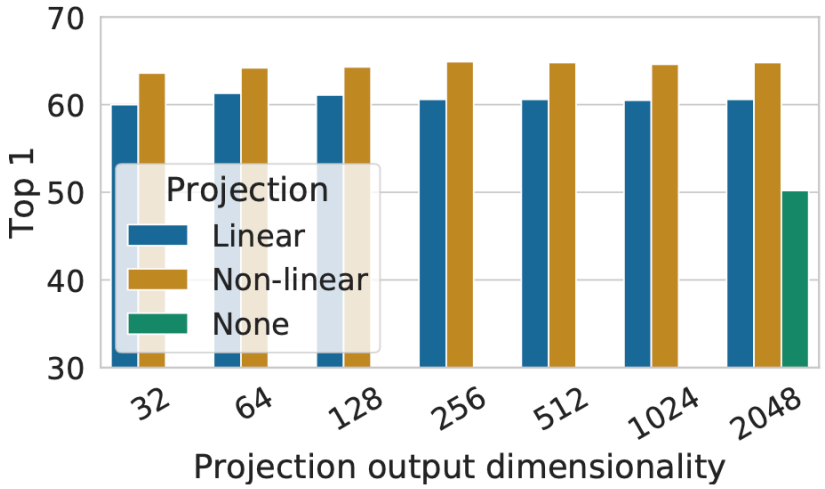
With this figure, we can see performance gap shrinks as model size increases Unsupervised learning benefits more from bigger models.

2.3. Projection head

Projection head is a relatively simple neural network the maps the representations to the space where contrastive loss is applied.

Specifically:

- Multilayer Perceptron (MLP)
- $\boldsymbol{z}_i = g(\boldsymbol{h}_i) = W^{(2)}\sigma\left(W^{(1)}\boldsymbol{h}_i\right)$, where σ is ReLU (non-linearity)



From the above figure, we can see Nonlinear is better than linear projection, and linear projection is better than no projection.

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

h contains more much more information about the transformation applied, while $g(h)$ loses some information. However, the authors conjectured that the importance of using the representation before the nonlinear projection is due to the loss of information introduced by the contrasted loss. g can remove the information that may be useful for the downstream tasks such as color orientation of the object.

2.4. Contrastive Loss Function

A contrastive loss function defined for a contrastive prediction task. Given a set $\{\tilde{\boldsymbol{x}}_k\}$ including a positive pair of examples $\tilde{\boldsymbol{x}}_i$ and $\tilde{\boldsymbol{x}}_j$, the contrastive prediction task aims to identify $\tilde{\boldsymbol{x}}_j$ in $\{\tilde{\boldsymbol{x}}_k\}_{k\neq i}$ for a given $\tilde{\boldsymbol{x}}_i$.

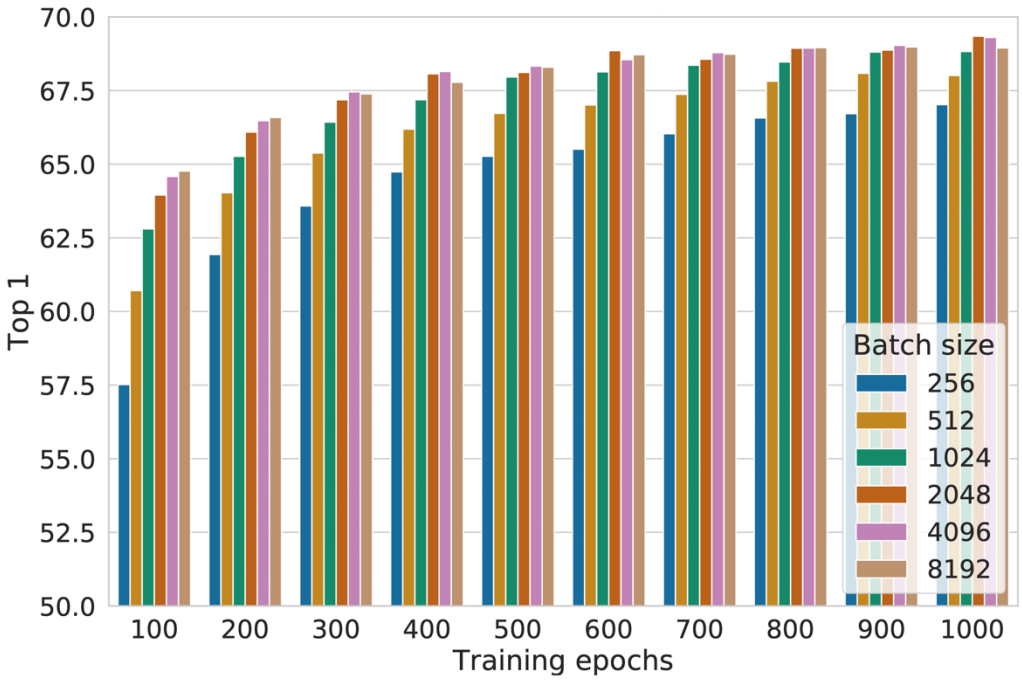
- Randomly sample a minibatch of N examples
- Define the task on pairs of augmented examples (2N images)
- Pick out a positive pair, 2 examples
- Treat the other 2(N − 1) augmented examples as negatives
- Loss function:

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

-
- SimCLR computes the loss from all positive pairs in a mini-batch

2.5. Batch Size

Vary the training batch size N from 256 to 8192.
Training with larger batch size and using the standard SGD or momentum with linear learning rate scale may be unstable.
To stabilize the training, use LARS optimizer (You et al., 2017) for all the batch sizes.



From the above figure, we see contrastive learning benefits from larger batch sizes and longer training.

Algorithm 1 SimCLR's main learning algorithm.

input: batch size N, constant τ, structure of $f, g, \mathcal{T}$.
for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ **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**
    $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$ # pairwise similarity
  end for
  define $\ell(i, j)$ **as** $\ell(i, j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away $g(\cdot)$

3. Evaluation

Dataset: ImageNet ILSVRC-2012
- also evaluated on CIFAR-10 and others (for transfer learning)

Protocol: linear evaluation
- A linear classifier is trained on top of the frozen base network
- Test accuracy is used as a proxy for representation quality

Data Augmentation: crop & resize, color distortion, and Gaussian blur
Optimizer: LARS with LR=4.8
Batch size: 4096
Epochs: 100

3.1. Comparison with State-of-the-art

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**

3.2. Evaluation on Semi-Supervised Learning and Transfer Learning

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**

	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

4. Conclusion

i. Composition of multiple data augmentation operations is crucial, in defining the contrastive prediction tasks. The unsupervised contrastive learning benefits from stronger data augmentation and supervised learning.

ii. Nonlinear transformation (g) substantially improves the quality of the learned representations.

iii. Contrastive learning benefits from larger batch sizes and more training steps. Like supervised learning, contrastive learning also benefits from deeper and wider networks