A Simple Framework for Contrastive Learning of Visual Representations (SimCLR)

[Chen et al., ICML 2020]

-Summary-

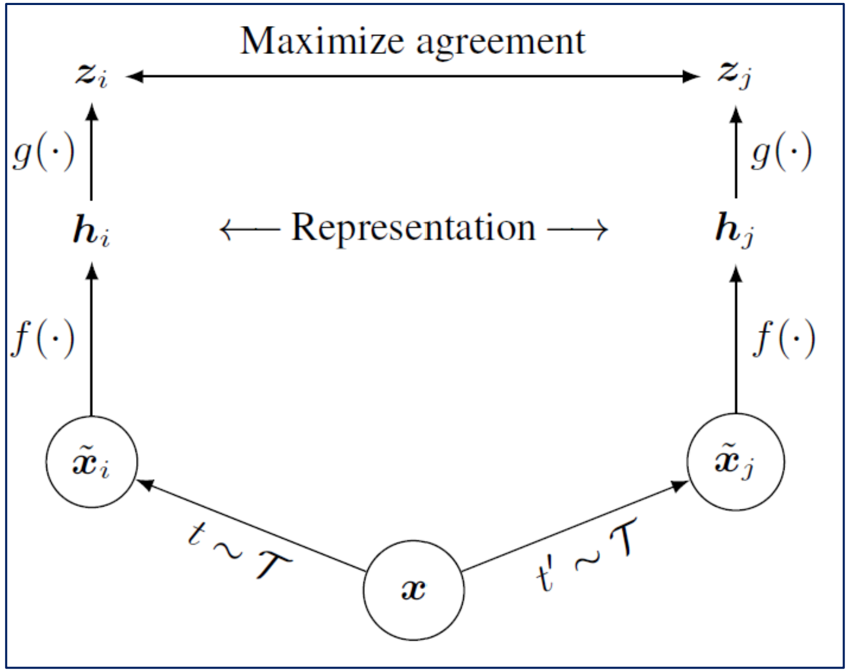
Introduction

- Discriminative approaches based on contrastive learning in the latent space have recently shown promising state-of-the-art results (Oord et al., 2018, Bachman et al., 2019)

- This paper introduce simple framework for contrastive learning of visual representations
  1. Composition of multiple data augmentation : Crop + Color distortion (most effective)
  2. Introducing learnable nonlinear transformation (projection head)
     $g : h_i \rightarrow z_i$ (h_i : representation)
  3. Representation learning with **NT-Xent** loss benefits from normalized embedding (cosine similarity) and adjusted temperature parameter (τ)
  4. CL benefits from larger batch sizes and longer training, also deeper and wider networks

Method

- Basic idea of SimCLR :
  - Learns representations by maximizing agreement between differently augmented views of the same data example via a contrastive loss in the latent space
  - **Framework for SimCLR** :



Notations :

- x **: data /** $\tilde{x}_i, \tilde{x}_j$ **: two different views by applying** t, t' **to** x
- $\mathcal{T}$ **: data augmentation distribution**

  (ex : crop location / gaussian noise injection size)
- f **: encoder network (here, author used ResNet)**
- h_i **: learned representation of** $\tilde{x}_i$
- g **: projection head MLP with one hidden layer**
- z_i **:** $g(h_i)$

Method

> ***Note (intuition behind NT-Xent loss) :***
>
> - $\boldsymbol{l_{i,j} = log\left(1 + \sum_{k=1,k\neq j}^{2N-1} exp\left(f^T f_k - f^T f^+\right)\right)}$
> - ***Use $f \rightarrow z$ by projection head and <u>cosine similarity</u> instead of inner product***

- **Framework of SimCLR**:
  - Sample a minibatch of N examples, which leads to $2(N-1)$ negative examples corresponding one positive pair.
  - Adopts **NT-Xent loss** (originated from ***multi-class N pair loss****, Sohn, 2016*) :

$$l_{i,j} = -\log \frac{\exp\left(\frac{sim(z_i, z_j)}{\tau}\right)}{\sum_{k=1,k\neq i}^{2N} \exp\left(\frac{sim(z_i, z_j)}{\tau}\right)}, \qquad L = \frac{1}{2N}\sum_{k=1}^{N}[l(2k-1,2k) + l(2k, 2k-1)]$$

  - Adopts similarity function (sim) as **cosine similarity** :

$$sim(u, v) \coloneqq u^T v / \|u\| \|v\|$$

Method

- **Main learning algorithm of SimCLR:**

Batch sampling

Data augmentation strategy sampling

Encode & projection

Cosine similarity

NT-Xext loss

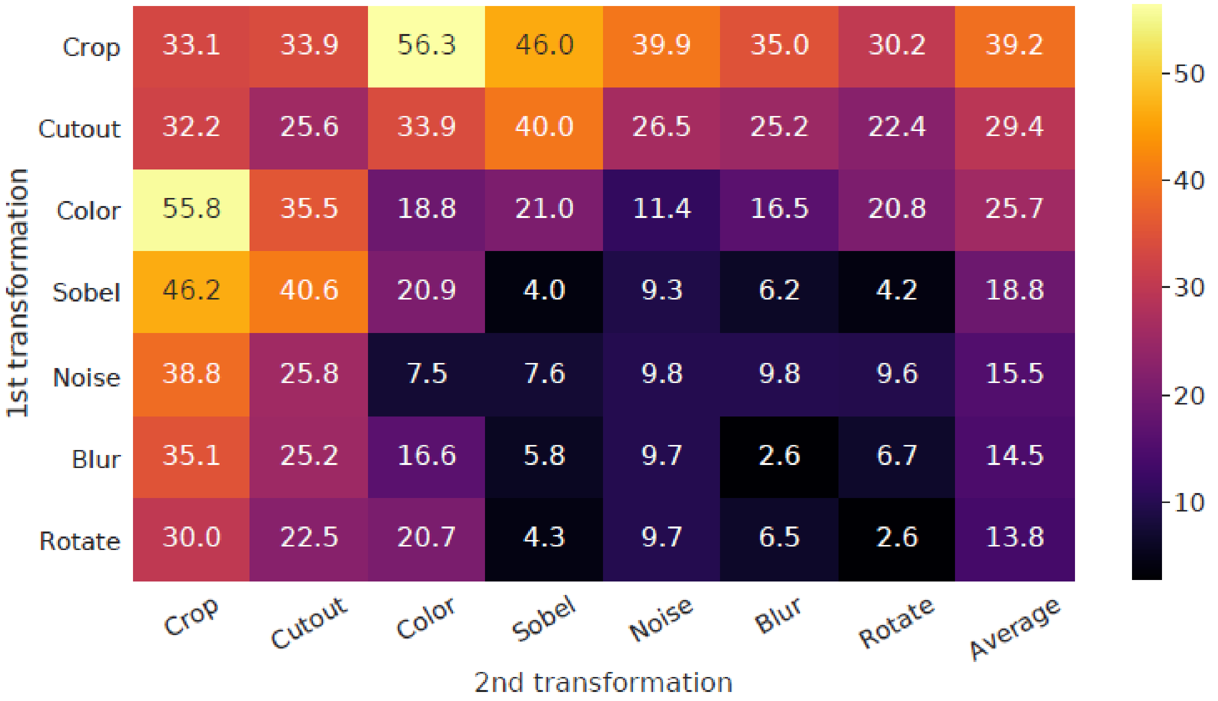
Algorithm 1 SimCLR's main learning algorithm.

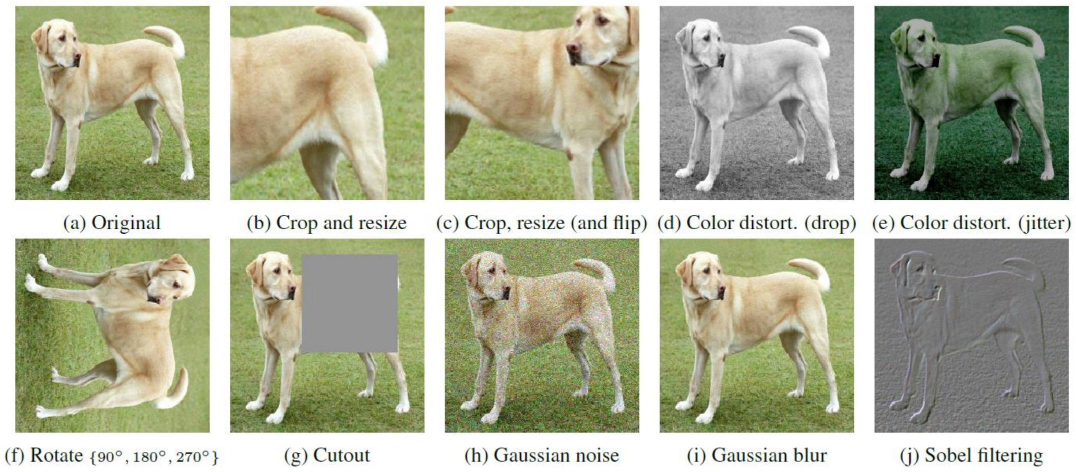
---

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 $g(\cdot)$

Claims and experiments

- Claim ① : Composition of multiple data augmentation is crucial for downstream performance
  - It turns out that single transformation does not suffices to learn good representation
  - The best composition of data augmentation : Crop -> Color distortion (56.3% top-1 acc)

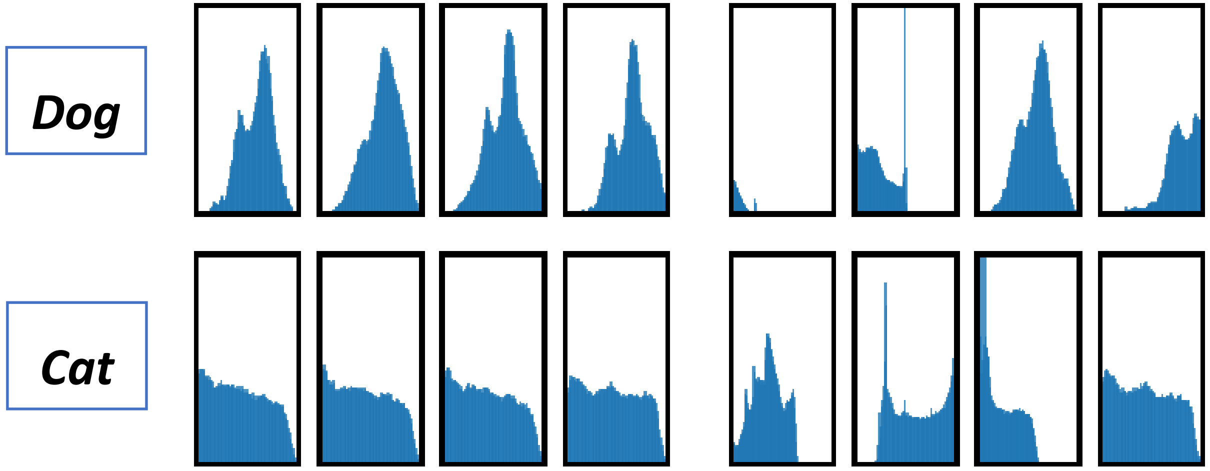




Left : Linear evaluation (by ImageNet top-1 acc) / Right : Illustrations of data augmentation strategies

Claims and experiments

- Claim ① : Composition of multiple data augmentation is crucial for downstream performance
  - Q : Why color distortion leads to distinctively higher downstream performance?
  - If we analyze histograms of pixel intensities, the model easily can distinguish images, biasing only to color histograms (act as short-cut) -> X helpful for generalization.



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

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

Side note : Higher color distortion strength helps better on unsupervised LR

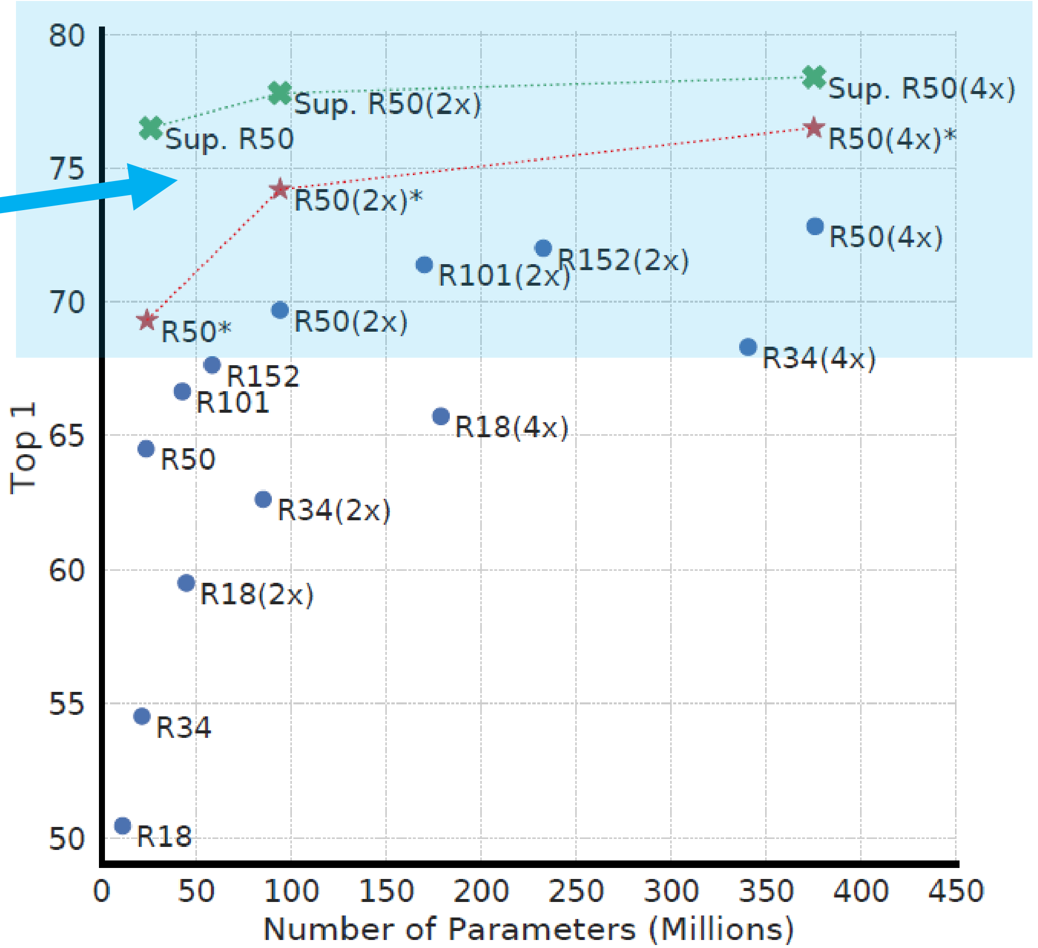
Histogram of pixel intensities (over all channels) for different crops of two different images (two rows)

Claims and experiments

- Claim ② : Unsupervised CL benefits more from bigger models than supervised learning.

- Observe that the gap between supervised model and linear classifiers from CL shrinks as the model size increase

∴ unsupervised learning benefits more from bigger models than its supervised counterparts



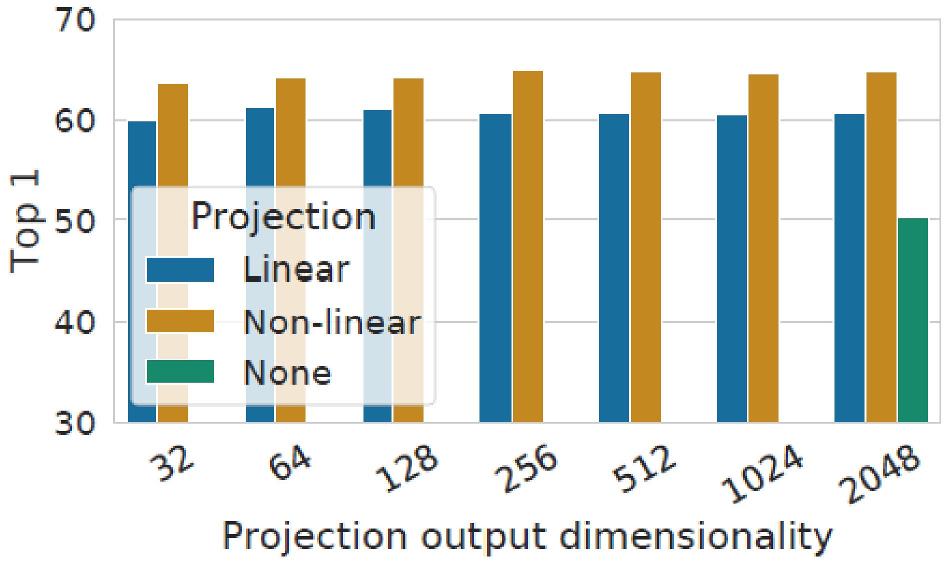
Linear evaluation of models with varied depth and width

Claims and experiments

- Claim ③ : Nonlinear projection head improves the representation quality of the layer before it.
  - As the penultimate layer forms useful representation on supervised learning, we can expect the layer before projection head to have useful representations.
  - By experiments, it turns out that using the layer before projection head improves the downstream performance dramatically.
  - Linear : $g(h) = W^{(1)}(h)$

    Non-linear : $g(h) = W^{(2)}\sigma\left(W^{(1)}h\right)$

    None : $g(h) = h$ (identity)



Linear evaluation of models with different projection heads and various dimensions of $z = g(h)$

Claims and experiments

- Claim ③ : Nonlinear projection head improves the representation quality of the layer before it.
  - Q : Fundamentally, why do we need projection head, and use the h rather than $g(h)$?

    1. $z = g(h)$ is trained to be invariant to data transformation. (only positive / negative)

    2. Hence, g can remove useful information, which might be useful for downstream task (ex : color or orientation of objects)

    3. h can capture this information much more than $g(h)$ **(Answer)**

What to predict?	Random guess	Representation h	Representation $g(h)$
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

Left : Accuracy of training additional MLPs on different representations to predict the transformation applied

∴ h contains more information about transformation than $g(h)$

Claims and experiments

- Claim ④ : Normalized cross entropy loss with adjustable temperature works better than alternatives
  - What is the 'Normalized cross entropy loss with adjustable temperature' ?
    - Recall the NT-Xent loss as below:

Normalized (blue)

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

Cross entropy loss (yellow)

(assuming u, v^+, v^- are l_2 normalized)

temperature (red)

Claims and experiments

- Claim ④ : Normalized cross entropy loss with adjustable temperature works better than alternatives
  - If we check the input gradient with respect to anchor u for various CL loss,
    1. l_2 normalization (cosine similarity) effectively weights different examples.
    2. Appropriate temperature τ helps the model learn from hard negatives. (~entropy)

       (ex : Assume $u^T v_{easy-} = -0.4$, $u^T v_{hard-} = -0.1$, then appropriate τ can exploit the exponential function to highlight the value (=$\exp(u^T v_{hard-}/\tau)$) effectively)

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 $\mathbf{0}$

Negative loss function and their gradients from well-known CL loss

Claims and experiments

- Claim ④ : Normalized cross entropy loss with adjustable temperature works better than alternatives
  - If we check the input gradient with respect to anchor u for various CL loss,
    1. l_2 normalization (cosine similarity) effectively weights different examples.
    2. Appropriate temperature τ helps the model learn from hard negatives. (~entropy)

    (ex : Assume $u^T v_{easy-} = -0.4$, $u^T v_{hard-} = -0.1$, then appropriate τ can exploit the exponential function to highlight the value (=$\exp(u^T v_{hard-}/\tau)$) effectively)

ℓ_2 norm?	τ	Entropy	Contrastive 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

Linear evaluation for models trained with different choices of l_2 normalization and temperature τ for NT-Xent loss

Entropy : entropy of softmax score (next slide)

Contrastive acc : accuracy to discriminate positive or negative.

Claims and experiments

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

Linear evaluation for models trained with different loss

- Claim ④ : Normalized cross entropy loss with adjustable temperature works better than alternatives
  - Unlike cross-entropy, other objective functions (Margin Triplet) do not weight the negatives by their relative hardness.
    - NT-Xent : Use Softmax score as the measure for hardness of data.
    - NT-Logistic : Use $Sigmoid(u^T v^- / \tau)$ for data valuation (but may not be effective)
    - Margin Triplet : Not consider
  - Hence, NT-Logistic / Margin Triplet benefits a lot from semi-hard negative mining (=sh)

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 $\mathbf{0}$

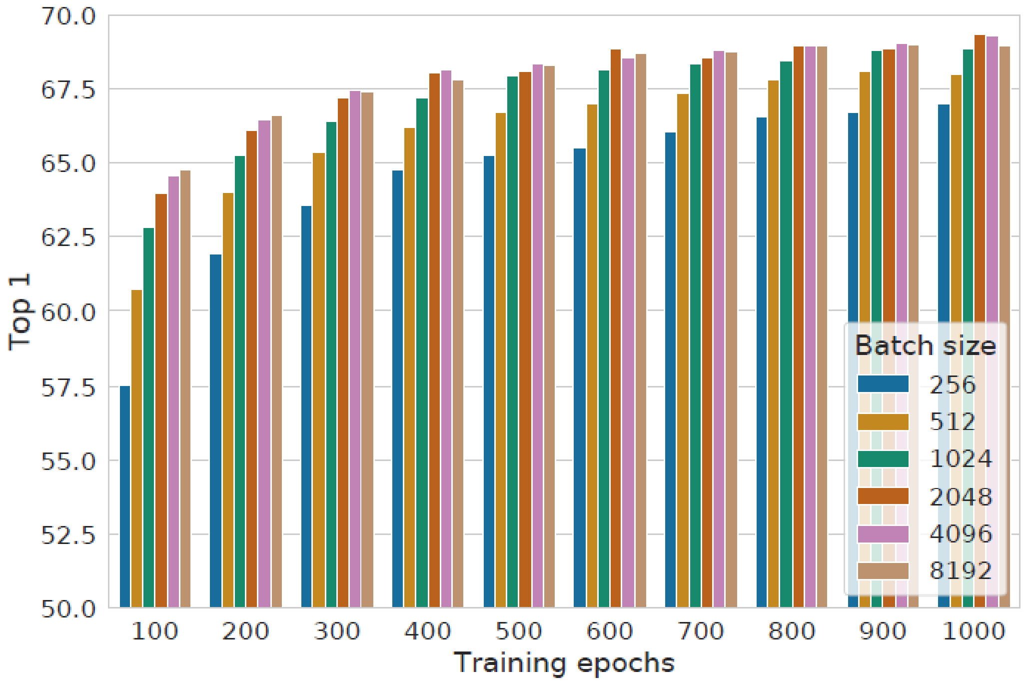
Claims and experiments

- Claim ⑤ : Contrastive learning benefits more from larger batch sizes and longer training.
  - It turns out that larger batch size (>= 2048) have a significant advantage over the smaller ones.
  - With more training epochs, the performance gaps between different batch sizes decrease of disappear.

Linear evaluation trained with different batch size and epochs

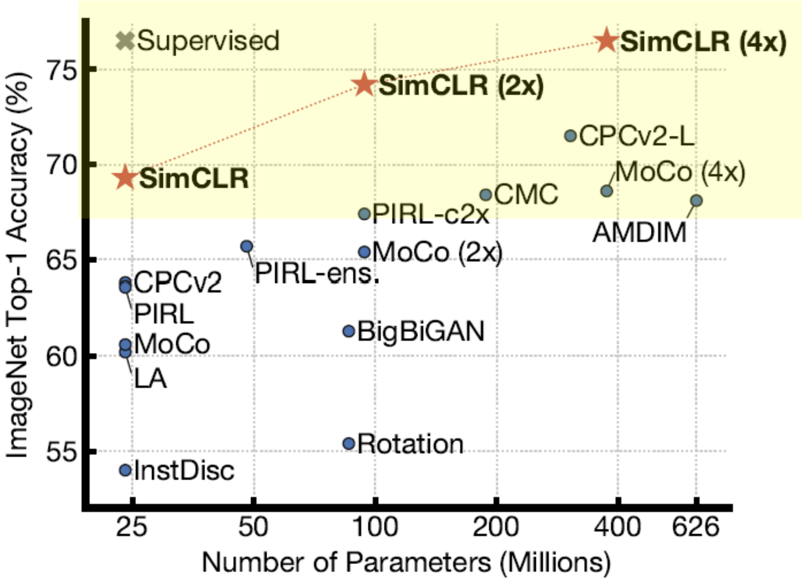
Note : why this happens ? (conjecture)

1. ***Larger batch size -> more negative samples per batch***
2. ***Longer training -> more negative samples for entire training***



Claims and experiments - Summary

- Claim ① : **<u>Composition of multiple data augmentation</u>** is crucial for downstream performance
- Claim ② : Unsupervised CL benefits more from **<u>bigger models</u>** than supervised learning.
- Claim ③ : **<u>Nonlinear projection head</u>** improves the representation quality of the layer before it.
- Claim ④ : **<u>Normalized cross entropy loss with adjustable temperature</u>** works better than alternatives.
- Claim ⑤ : Contrastive learning benefits more from **<u>larger batch sizes and longer training</u>**.



ImageNet Top-1 accuracy of linear classifiers trained on representations learned with different self-supervised methods