Self-Supervised Contrastive Learning Theory and Application

Pengcheng Xu

2020, May, 25

Self-supervised Learning

- Unsupervised Learning:
  - No labels
- Motivation:
  - Learn the high-level representation by exploring the dataset itself for downstream tasks. (Compared with fine-tuning pretrain)
- Why self-supervised learning:
  - Data itself provide more rich hierarchical information than simple labels
  - Labeling may be extremely expensive like RL
  - May perform even better than the supervised fine-tuning tasks.

Self-supervised learning approaches

- Generative method: Not Good
  - Pixel reconstruction like VAE
  - Pixel level reconstruction is insanely computational inefficiency
  - Pixel level may not be necessary
- Discriminative method: contrastive method is the best so far
  - Pixel info is not necessary; Semantic information is enough
  - Construct supervised learning on unsupervised tasks
  - Find positive and negative examples

Contrastive learning framework

- Construct supervised tasks from unlabeled data (Pretext, Instance discrimination)

- Learn a mapping f which map samples in the same 'class' into similar place and push away the samples in different 'class'.

$$s(f(x), f(x^+)) >> s(f(x), f(x^-))$$

- How to find the mapping f?
- How to define the distance metric?

$$\mathop{\mathbb{E}}_{x,x^+,x^-}\left[-\log\left(\frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + e^{f(x)^T f(x^-)}}\right)\right]$$

- How to find and train the positive and negative samples?

Paper list

- Momentum Contrast for Unsupervised Visual Representation Learning
- A Simple Framework for Contrastive Learning of Visual Representations
- Contrastive Representation Distillation

Momentum Contrast for Unsupervised Visual Representation Learning

Kaiming He   Haoqi Fan   Yuxin Wu   Saining Xie   Ross Girshick

Facebook AI Research (FAIR)

Code: https://github.com/facebookresearch/moco

Method

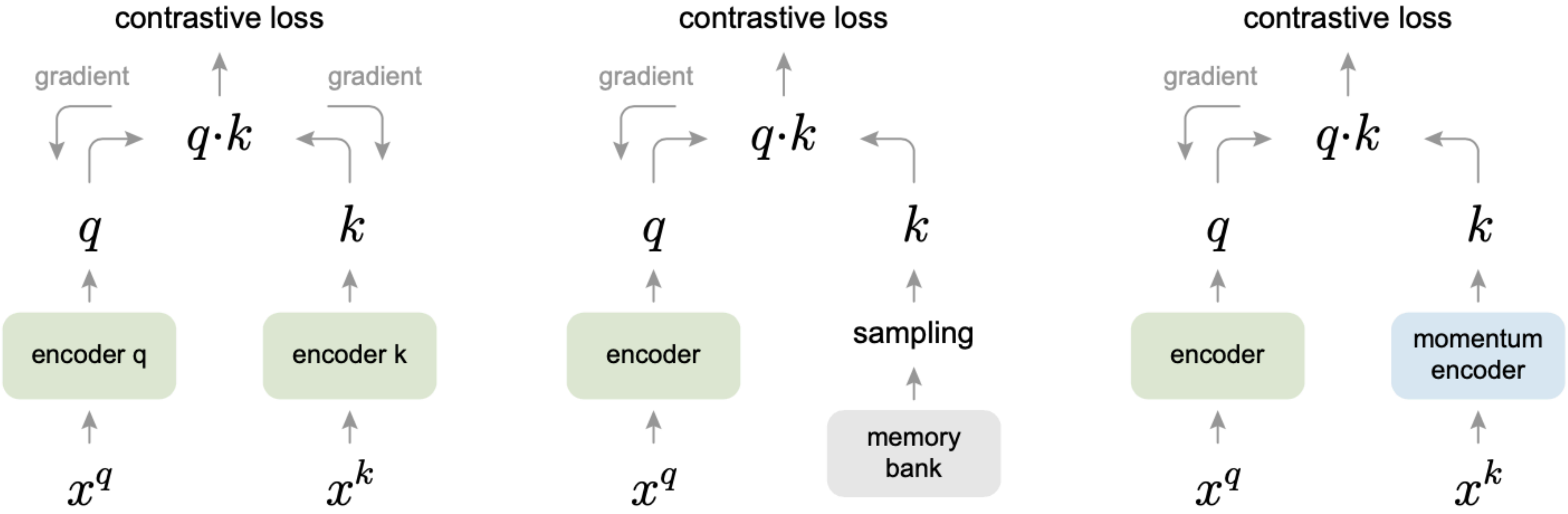
- Task:
  - Simple Instance discrimination: query match a key if they are different crops of the same image;
- Loss: InfoNCE

$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i / \tau)}$$

Push features of the same class close; Push features of different class away

- Momentum update encoder as a queue
  - Decouple the encoder(dictionary) from mini-batches and provide consistent key space.

Momentum update



(a) end-to-end (b) memory bank (c) MoCo

(a) End-to-End:

Not working well, batch-size equals to dictionary size, limited by GPU mem

(a) Memory bank:

Memory bank includes features of all keys; No backpropagation; Cannot scale up to large dataset

(c) Momentum update

Smooth update of key encoder while keeping a large queue

$$\theta_k \leftarrow m\theta_k + (1 - m)\theta_q$$

Algorithm 1 Pseudocode of MoCo in a PyTorch-like style.

```
# f_q, f_k: encoder networks for query and key
# queue: dictionary as a queue of K keys (CxK)
# m: momentum
# t: temperature

f_k.params = f_q.params # initialize
for x in loader: # load a minibatch x with N samples
   x_q = aug(x) # a randomly augmented version
   x_k = aug(x) # another randomly augmented version

   q = f_q.forward(x_q) # queries: NxC
   k = f_k.forward(x_k) # keys: NxC
   k = k.detach() # no gradient to keys

   # positive logits: Nx1
   l_pos = bmm(q.view(N,1,C), k.view(N,C,1))

   # negative logits: NxK
   l_neg = mm(q.view(N,C), queue.view(C,K))

   # logits: Nx(1+K)
   logits = cat([l_pos, l_neg], dim=1)

   # contrastive loss, Eqn.(1)
   labels = zeros(N) # positives are the 0-th
   loss = CrossEntropyLoss(logits/t, labels)

   # SGD update: query network
   loss.backward()
   update(f_q.params)

   # momentum update: key network
   f_k.params = m*f_k.params+(1-m)*f_q.params

   # update dictionary
   enqueue(queue, k) # enqueue the current minibatch
   dequeue(queue) # dequeue the earliest minibatch
```

bmm: batch matrix multiplication; mm: matrix multiplication; cat: concatenation.

The key encoder is only updated with the query encoder

Only queries contribute to gradients

Features from key encoder are saved in the queue.

Experiment results

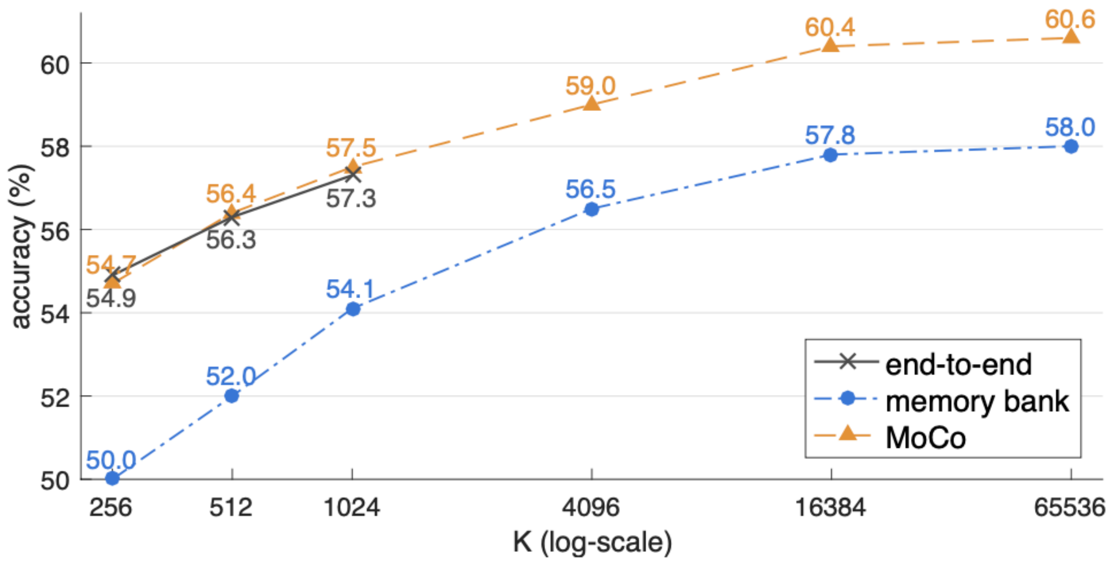


Figure 3. **Comparison of three contrastive loss mechanisms** under the ImageNet linear classification protocol. We adopt the same pretext task (Sec. 3.3) and only vary the contrastive loss mechanism (Figure 2). The number of negatives is K in memory bank and MoCo, and is $K-1$ in end-to-end (offset by one because the positive key is in the same mini-batch). The network is ResNet-50.

The Dictionary size influence the performance

MoCo outperforms the other two

A Simple Framework for Contrastive Learning of Visual Representations

Ting Chen [1] **Simon Kornblith** [1] **Mohammad Norouzi** [1] **Geoffrey Hinton** [1]

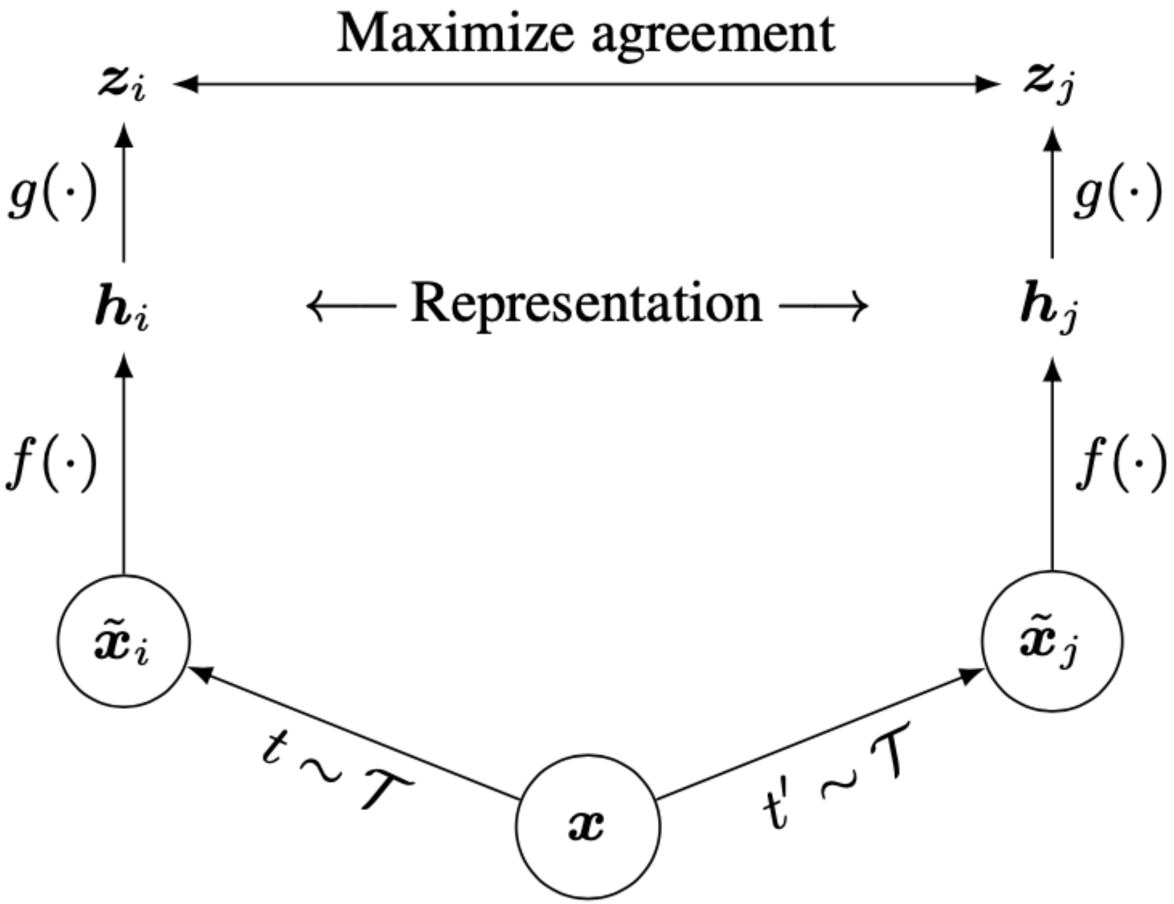
Overview

1. Composition of data augmentations plays a critical role in defining effective predictive tasks;

2. Learnable nonlinear transformation between the representation and the contrastive loss substantially improves the quality of the learned representations;

3. Contrastive learning benefits from larger batch sizes(more concretely, negative samples) and more training steps compared to supervised learning.

Method



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

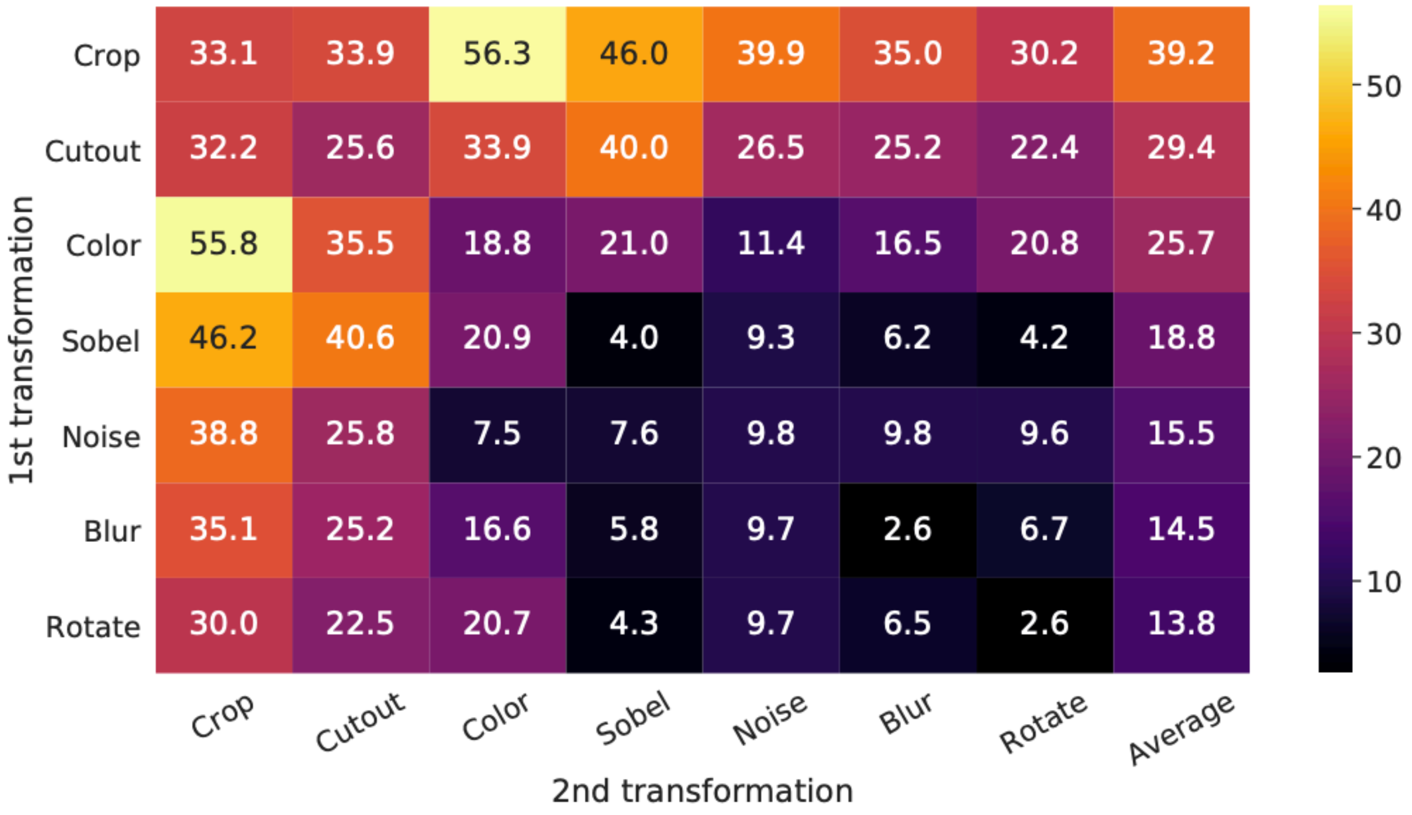
$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)},$$

Algorithm

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 $g(\cdot)$

Results



1.
More compositions of data augmentations
Better Performance

Results

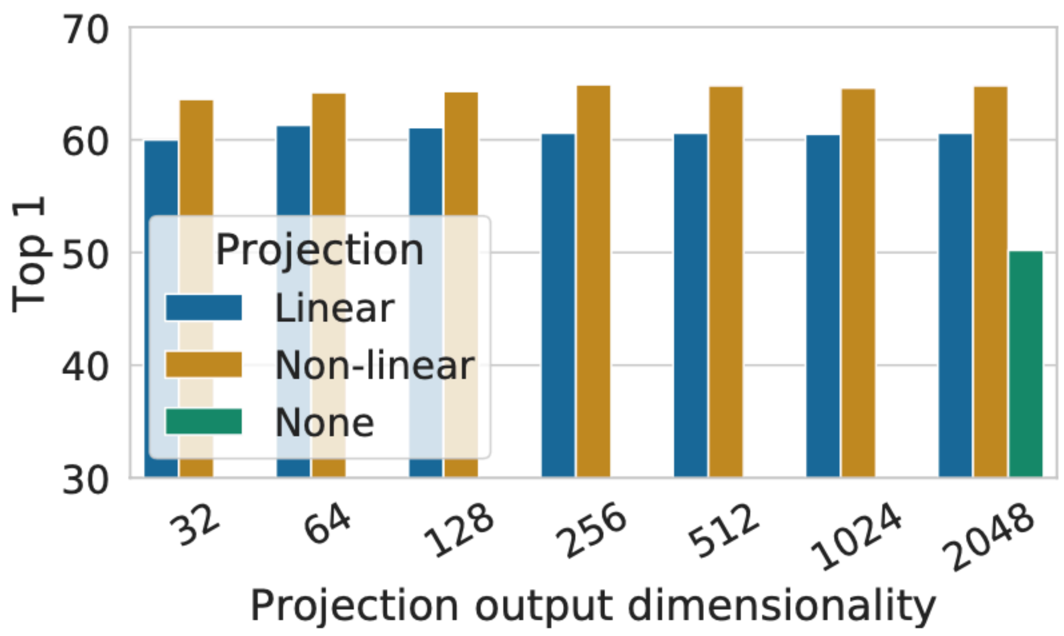


Figure 8. Linear evaluation of representations with different projection heads $g(\cdot)$ and various dimensions of $\boldsymbol{z} = g(\boldsymbol{h})$. The representation $\boldsymbol{h}$ (before projection) is 2048-dimensional here.

2.
Nonlinearity between the feature space and contrastive space helps improve the performance.

Reasons:
Contrastive loss causes loss of information
z is trained to be invariant to data transformation.
g() remove information that may be useful for downstream tasks.

Results

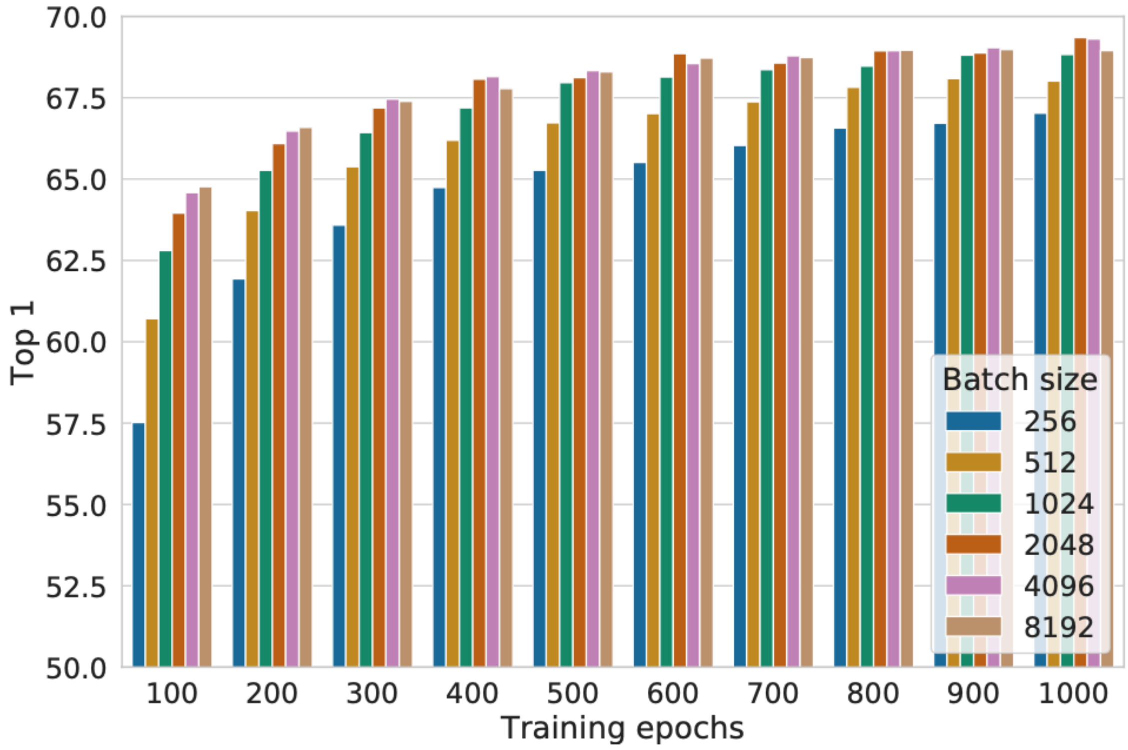


Figure 9. Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch.

3.
Larger Batch Size
More Epochs
Better performance

Results

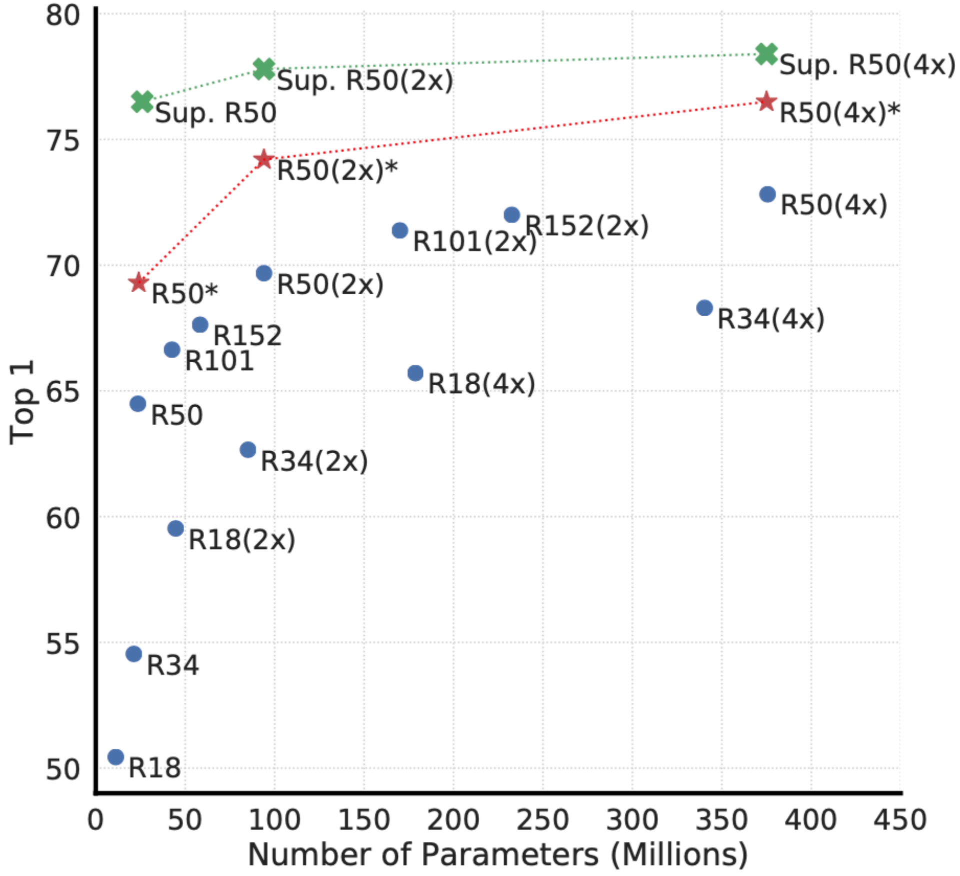


Figure 7. Linear evaluation of models with varied depth and width. Models in blue dots are ours trained for 100 epochs, models in red stars are ours trained for 1000 epochs, and models in green crosses are supervised ResNets trained for 90 epochs[7] (He et al., 2016).

4.
More parameters
Wider neural net
Better Performance

CONTRASTIVE REPRESENTATION DISTILLATION

Yonglong Tian
MIT CSAIL
yonglong@mit.edu

Dilip Krishnan
Google Research
dilipkay@google.com

Phillip Isola
MIT CSAIL
phillipi@mit.edu

Overview

- 1. Contrastive objective for transferring knowledge between deep networks.
- 2. Applications to model compression, cross-modal transfer, and ensemble distillation.
- 3. Benchmarking 12 recent distillation methods and outperforming all methods.

Method

- Limitations:
  - 1. Original knowledge distillation loss (KL divergence) treat all dimensions independently, thus cannot transfer **structural** knowledge.

    $$\psi(\mathbf{y}^S, \mathbf{y}^T) = \sum_i \phi_i(\mathbf{y}_i^S, \mathbf{y}_i^T)^*$$

  - 2. KL divergence does not exist for cross-modal transferring (Image to Sound)

  - Contrastive loss better transfer all information of the teacher's representation rather than transferring knowledge about conditionally independent output class probabilities.

Method

- 1. Minimize the mutual information of Student and Teacher.
- 2. Construct the mutual information with contrastive method
- 3. Find the lower bound of mutual information
- 4. Maximize the lower bound (contrastive learning)

Method

$$
\begin{aligned}
&x \sim p_{\texttt{data}}(x) && \triangleleft \quad \textbf{data} \\
&S = f^S(x) && \triangleleft \quad \textbf{student's representation} \\
&T = f^T(x) && \triangleleft \quad \textbf{teacher's representation}
\end{aligned}
$$

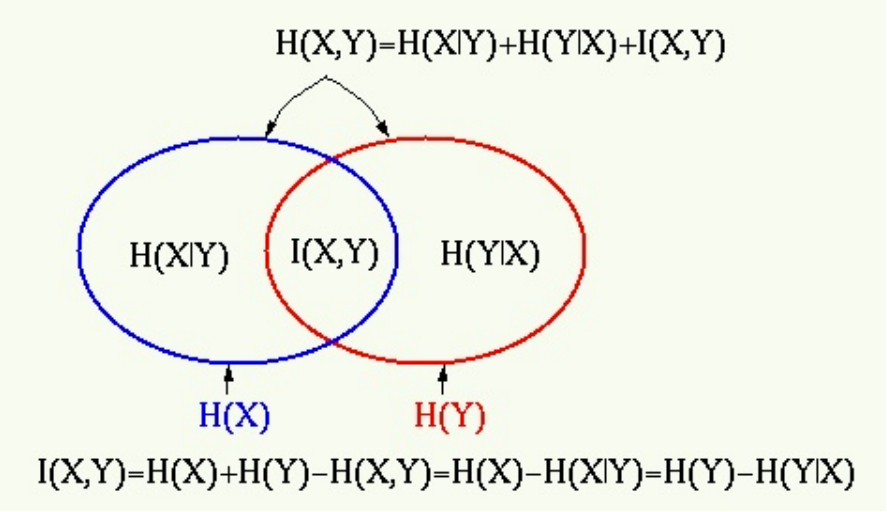
$$q(T, S|C = 1) = p(T, S), \quad q(T, S|C = 0) = p(T)p(S)$$

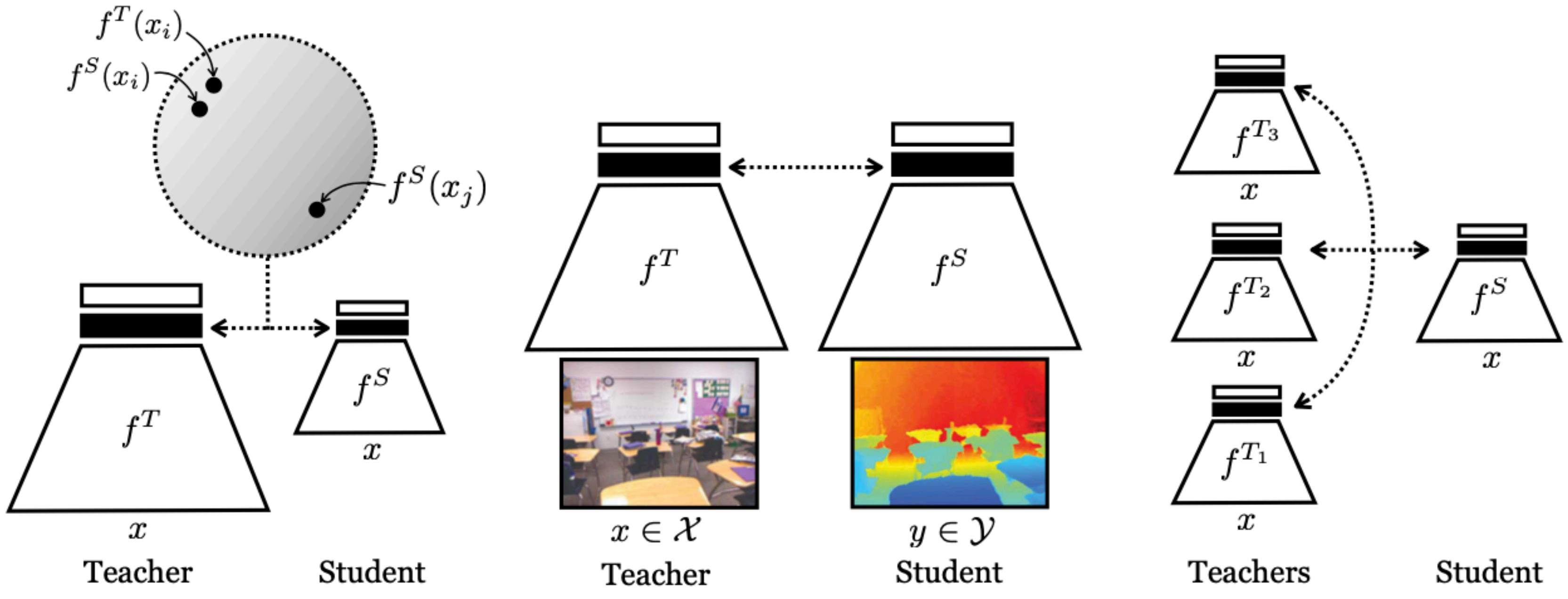
$$q(C = 1) = \frac{1}{N+1}, \quad q(C = 0) = \frac{N}{N+1}$$

$$I(T; S) \geq \log(N) + \boxed{\mathbb{E}_{q(T,S|C=1)} \log q(C = 1|T, S)} \qquad \triangleleft \quad \textbf{MI bound}$$

Maximize it w.r.t Student

$$
\begin{aligned}
\mathcal{L}_{critic}(h) &= \mathbb{E}_{q(T,S|C=1)}[\log h(T, S)] + N\mathbb{E}_{q(T,S|C=0)}[1 - \log(h(T, S))] \\
h^* &= \arg\max_h \mathcal{L}_{critic}(h) \qquad \triangleleft \quad \textbf{optimal critic}
\end{aligned}
$$





(a) Model compression (b) Cross-modal transfer (c) Ensemble distillation

Figure 1: The three distillation settings we consider: (a) compressing a model, (b) transferring knowledge from one modality (e.g., RGB) to another (e.g., depth), (c) distilling an ensemble of nets into a single network. The constrastive objective encourages the teacher and student to map the same input to close representations (in some metric space), and different inputs to distant representations, as indicated in the shaded circle.

Model compression

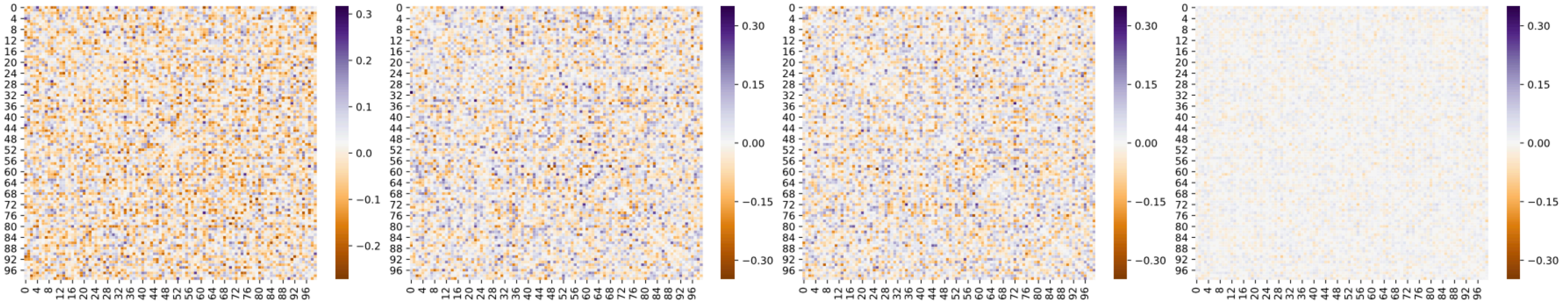
Teacher Student	WRN-40-2 WRN-16-2	WRN-40-2 WRN-40-1	resnet56 resnet20	resnet110 resnet20	resnet110 resnet32	resnet32x4 resnet8x4	vgg13 vgg8
Teacher Student	75.61 73.26	75.61 71.98	72.34 69.06	74.31 69.06	74.31 71.14	79.42 72.50	74.64 70.36
KD*	74.92	73.54	70.66	70.67	73.08	73.33	72.98
FitNet*	73.58 (↓)	72.24 (↓)	69.21 (↓)	68.99 (↓)	71.06 (↓)	73.50 (↑)	71.02 (↓)
AT	74.08 (↓)	72.77 (↓)	70.55 (↓)	70.22 (↓)	72.31 (↓)	73.44 (↑)	71.43 (↓)
SP	73.83 (↓)	72.43 (↓)	69.67 (↓)	70.04 (↓)	72.69 (↓)	72.94 (↓)	72.68 (↓)
CC	73.56 (↓)	72.21 (↓)	69.63 (↓)	69.48 (↓)	71.48 (↓)	72.97 (↓)	70.71 (↓)
VID	74.11 (↓)	73.30 (↓)	70.38 (↓)	70.16 (↓)	72.61 (↓)	73.09 (↓)	71.23 (↓)
RKD	73.35 (↓)	72.22 (↓)	69.61 (↓)	69.25 (↓)	71.82 (↓)	71.90 (↓)	71.48 (↓)
PKT	74.54 (↓)	73.45 (↓)	70.34 (↓)	70.25 (↓)	72.61 (↓)	73.64 (↑)	72.88 (↓)
AB	72.50 (↓)	72.38 (↓)	69.47 (↓)	69.53 (↓)	70.98 (↓)	73.17 (↓)	70.94 (↓)
FT*	73.25 (↓)	71.59 (↓)	69.84 (↓)	70.22 (↓)	72.37 (↓)	72.86 (↓)	70.58 (↓)
FSP*	72.91 (↓)	n/a	69.95 (↓)	70.11 (↓)	71.89 (↓)	72.62 (↓)	70.23 (↓)
NST*	73.68 (↓)	72.24 (↓)	69.60 (↓)	69.53 (↓)	71.96 (↓)	73.30 (↓)	71.53 (↓)
CRD	**75.48** (↑)	**74.14** (↑)	**71.16** (↑)	**71.46** (↑)	**73.48** (↑)	**75.51** (↑)	**73.94** (↑)
CRD+KD	75.64 (↑)	74.38 (↑)	71.63 (↑)	71.56 (↑)	73.75 (↑)	75.46 (↑)	74.29 (↑)

Table 1: Test *accuracy* (%) of student networks on CIFAR100 of a number of distillation methods (ours is CRD); see Appendix for citations of other methods. ↑ denotes outperformance over KD and ↓ denotes underperformance. We note that CRD is the *only* method to always outperform KD (and also outperforms all other methods). We denote by * methods where we used our reimplementation based on the paper; for all other methods we used author-provided or author-verified code. Average over 5 runs.

Teacher Student	vgg13 MobileNetV2	ResNet50 MobileNetV2	ResNet50 vgg8	resnet32x4 ShuffleNetV1	resnet32x4 ShuffleNetV2	WRN-40-2 ShuffleNetV1
Teacher Student	74.64 64.6	79.34 64.6	79.34 70.36	79.42 70.5	79.42 71.82	75.61 70.5
KD*	67.37	67.35	73.81	74.07	74.45	74.83
FitNet*	64.14 (↓)	63.16 (↓)	70.69 (↓)	73.59 (↓)	73.54 (↓)	73.73 (↓)
AT	59.40 (↓)	58.58 (↓)	71.84 (↓)	71.73 (↓)	72.73 (↓)	73.32 (↓)
SP	66.30 (↓)	68.08 (↑)	73.34 (↓)	73.48 (↓)	74.56 (↑)	74.52 (↓)
CC	64.86 (↓)	65.43 (↓)	70.25 (↓)	71.14 (↓)	71.29 (↓)	71.38 (↓)
VID	65.56 (↓)	67.57 (↑)	70.30 (↓)	73.38 (↓)	73.40 (↓)	73.61 (↓)
RKD	64.52 (↓)	64.43 (↓)	71.50 (↓)	72.28 (↓)	73.21 (↓)	72.21 (↓)
PKT	67.13 (↓)	66.52 (↓)	73.01 (↓)	74.10 (↑)	74.69 (↑)	73.89 (↓)
AB	66.06 (↓)	67.20 (↓)	70.65 (↓)	73.55 (↓)	74.31 (↓)	73.34 (↓)
FT*	61.78 (↓)	60.99 (↓)	70.29 (↓)	71.75 (↓)	72.50 (↓)	72.03 (↓)
NST*	58.16 (↓)	64.96 (↓)	71.28 (↓)	74.12 (↑)	74.68 (↑)	74.89 (↑)
CRD	**69.73** (↑)	**69.11** (↑)	**74.30** (↑)	**75.11** (↑)	**75.65** (↑)	**76.05** (↑)
CRD+KD	69.94 (↑)	69.54 (↑)	74.58 (↑)	75.12 (↑)	76.05 (↑)	76.27 (↑)

Table 2: Top-1 test *accuracy* (%) of student networks on CIFAR100 of a number of distillation methods (ours is CRD) for transfer across very different teacher and student architectures. CRD outperforms KD and all other methods. Importantly, some methods that require very similar student and teacher architectures perform quite poorly. E.g. FSP (Yim et al., 2017) cannot even be applied; AT (Ba & Caruana, 2014) and FitNet (Zagoruyko & Komodakis, 2016a) perform very poorly etc. We denote by * methods where we used our reimplementation based on the paper; for all other methods we used author-provided or author-verified code. Average over 3 runs.

Correlations



(a) Student: vanilla (b) Student: AT (c) Student: KD (d) Student: ours (CRD)

Figure 2: The correlations between class logits of a teacher network are ignored by regular cross-entropy. Distillation frameworks use "soft targets" (Hinton et al., 2015) which effectively capture such correlations and transfer them to the student network, leading to the success of distillation. We visualize here the *difference* of correlation matrices of student and teacher logits, for different student networks on a CIFAR-100 knowledge distillation task: (a) Student trained without distillation, showing that the teacher and student cross-correlations are very different; (b) Student distilled by attention transfer (Zagoruyko & Komodakis, 2016a); showing reduced difference (see axis); (c) Student distilled by KL divergence (Hinton et al., 2015), also showing reduced difference; (d) Student distilled by our contrastive objective, showing significant matching between student's and teacher's correlations. In this visualization, we use WRN-40-2 as teacher and WRN-40-1 as student.

Transferability of Representations

	Student	KD	AT	FitNet	CRD	CRD+KD	Teacher
CIFAR100→STL-10	69.7	70.9	70.7	70.3	71.6	**72.2**	68.6
CIFAR100→TinyImageNet	33.7	33.9	34.2	33.5	**35.6**	35.5	31.5

Table 4: We transfer the representation learned from CIFAR100 to STL-10 and TinyImageNet datasets by freezing the network and training a linear classifier on top of the last feature layer to perform 10-way (STL-10) or 200-way (TinyImageNet) classification. For this experiment, we use the combination of teacher network WRN-40-2 and student network WRN-16-2. Classification accuracies (%) are reported.

Q&A

Thank You Very Much