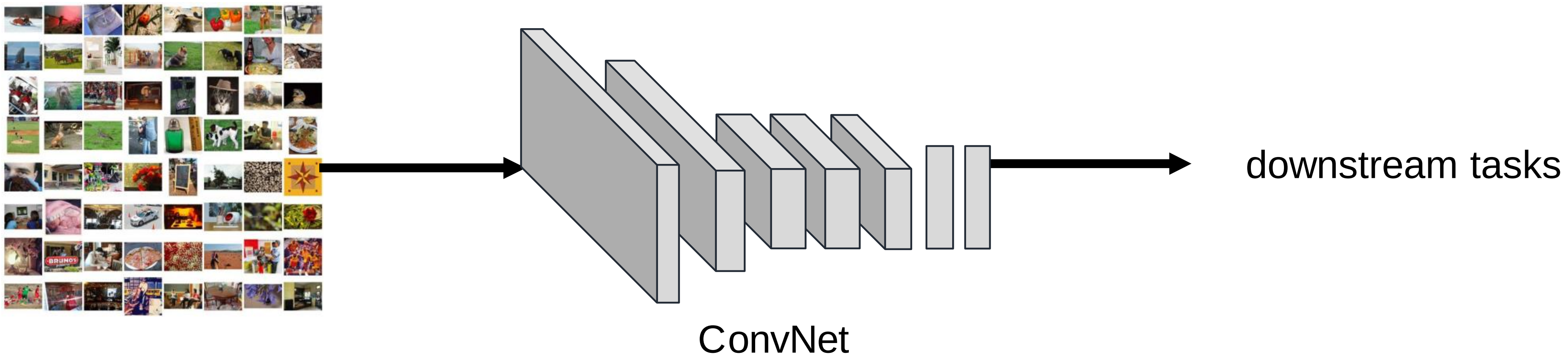
Self-Supervised Learning

Jaegul Choo (주재걸)
KAIST
https://sites.google.com/site/jaegulchoo/
Slides made by my student, Hojoon Lee

Introduction to the Self-Supervised Learning

Success story of supervision: Pre-training

- Features from networks pre-trained on ImageNet can be used for a variety of different downstream tasks



Images from ImageNet

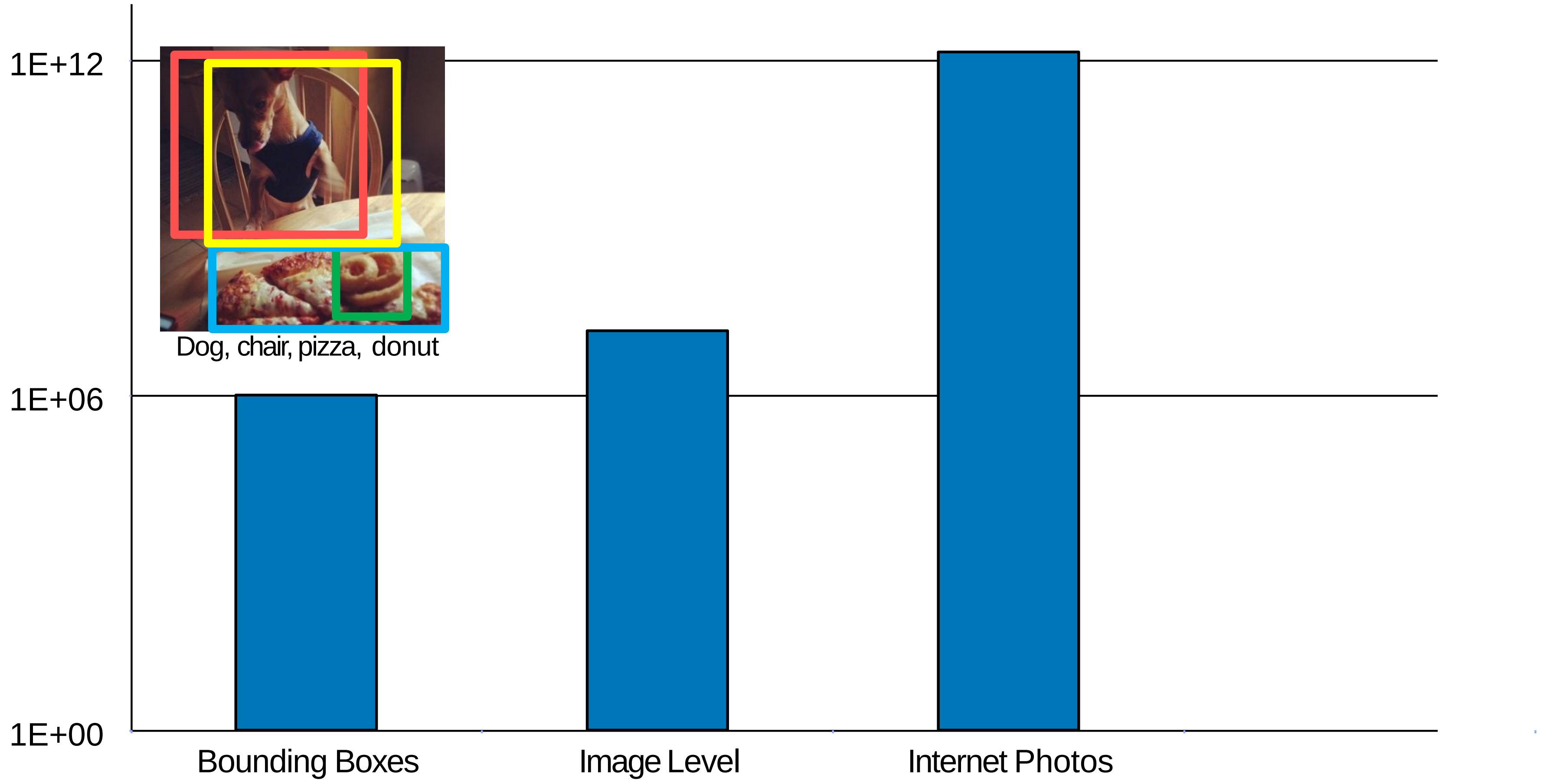
Learn a Representation

Success story of supervision: Pre-training

- Pre-train on large supervised dataset
- Collect a dataset of “supervised” images
- Train a Convolutional Network

Can we get labels for all data?

- Getting "real" labels is difficult and expensive
  - ImageNet with 14M images took 22 human years.

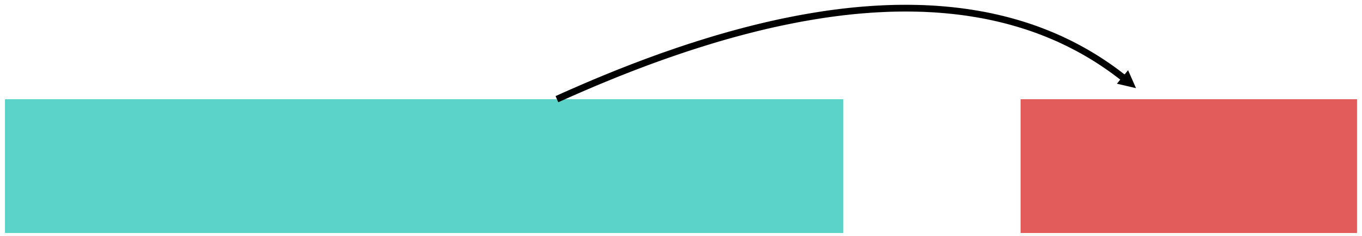


The promise of “alternative” supervision

- Obtain labels using “semi-automatic” process instead
  - Hashtags
  - Locations
  - Using the data itself: **self-supervised**

What is self-supervised Learning?

- Obtain “labels” from the data itself by using a “semi-automatic” process
- Train network with predicting the “semi-automatically” obtained labels



Observed data

Hidden data

Hidden property of the data

Simple Self-Supervised Models in Computer vision

- Simple pre-text tasks
  - CE: Fill in the blanks
  - RotNet: Predicting the rotation
  - JigSAW: Solving the Jigsaw-puzzle

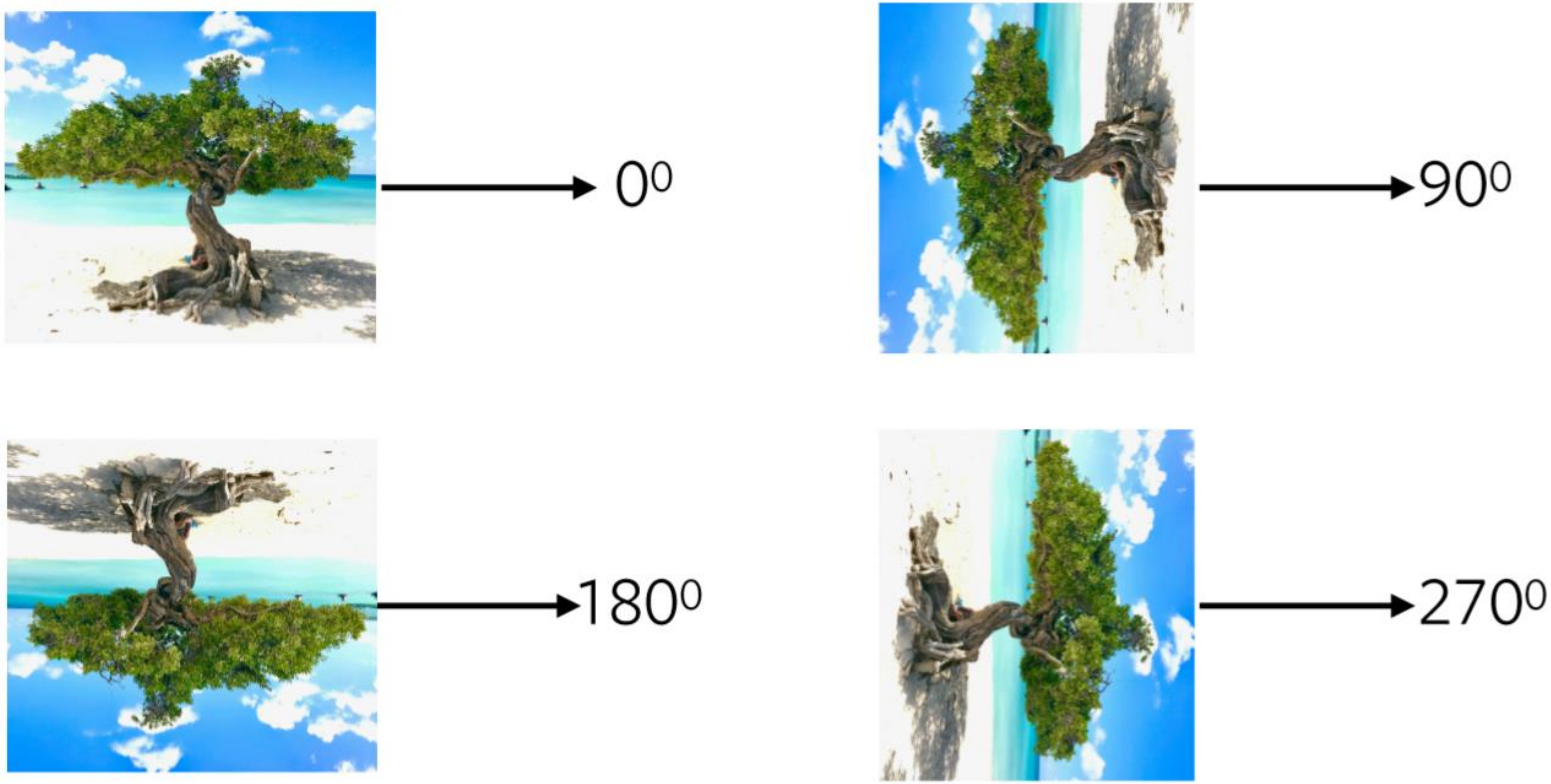
CE: Context Autoencoders

- Fill in the blanks of image

Pathak et al., Context Encoders: Feature Learning by Inpainting., CVPR, 2016

RotNet

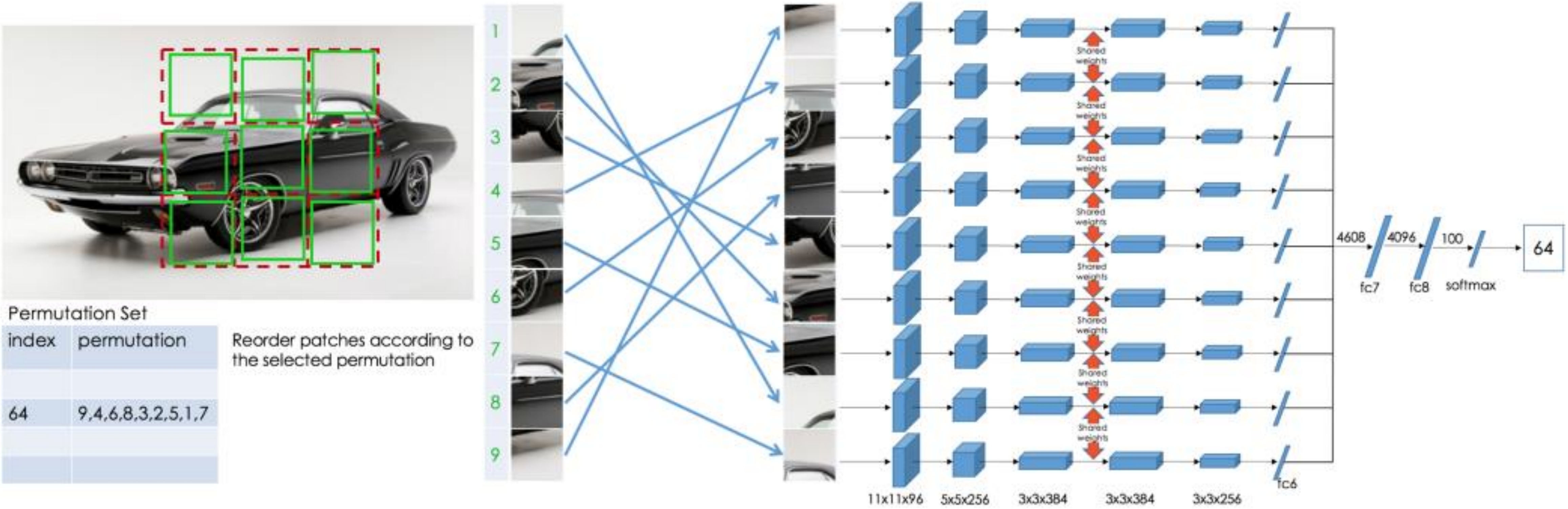
- Predicting Rotation of Images



Gidaris et al., Unsupervised Representation Learning by Predicting Image Rotations., ICLR, 2018

JigSAW

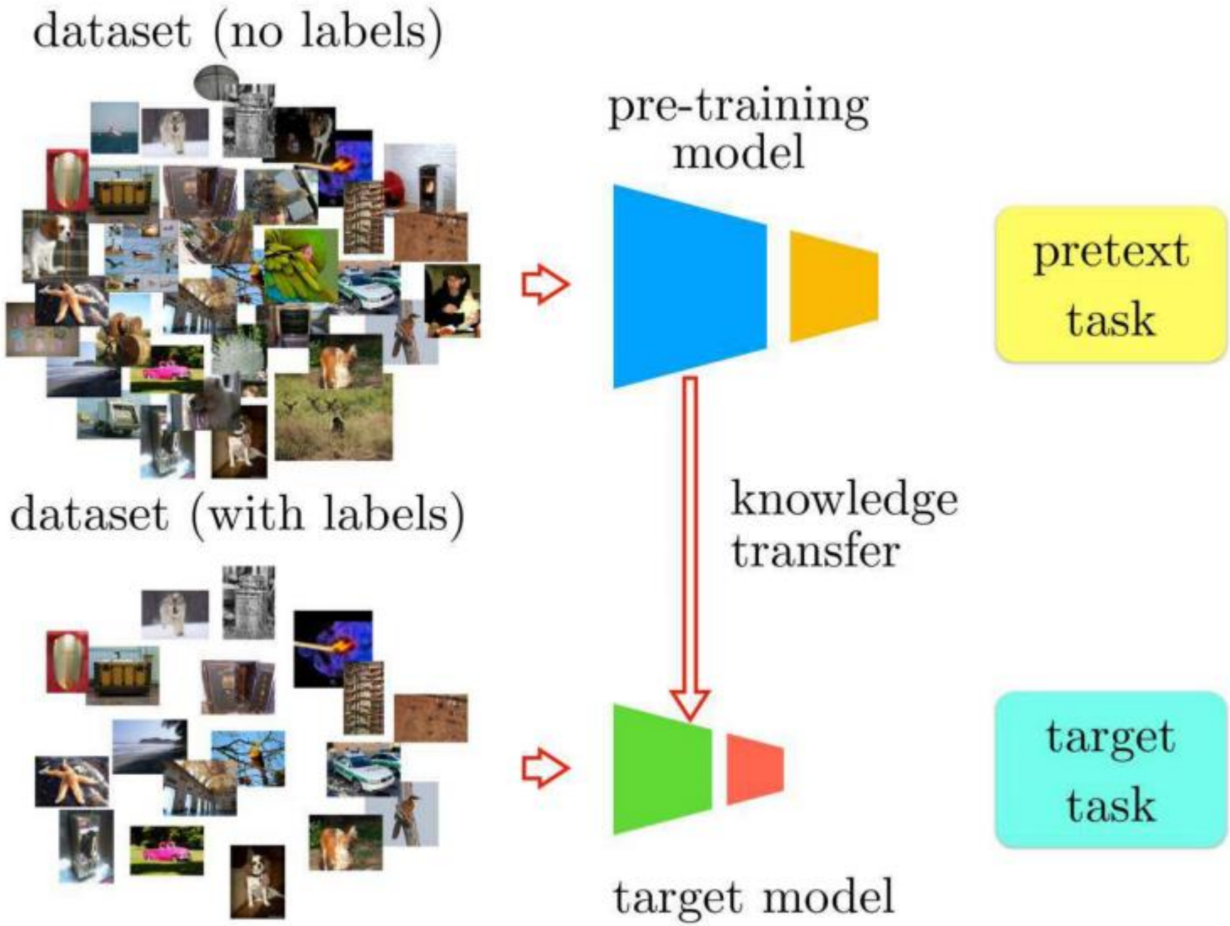
- Solving the jigsaw puzzle



Noroozi et al., Unsupervised Representation Learning by Solving Jigsaw Puzzles., ECCV, 2016

Evaluation Protocol

- Evaluate the pre-trained representations through fine-tuning in a transfer learning setting



- Classification (ImageNet-10K)
  - Freeze a pre-trained model
  - Train a linear layer for down-stream tasks
- Detection, Segmentation (PASCAL VOC)
  - Initialize with pre-trained model
  - Fine-tune the pre-trained model with an additional task-specific model

What is missing in pre-text tasks?

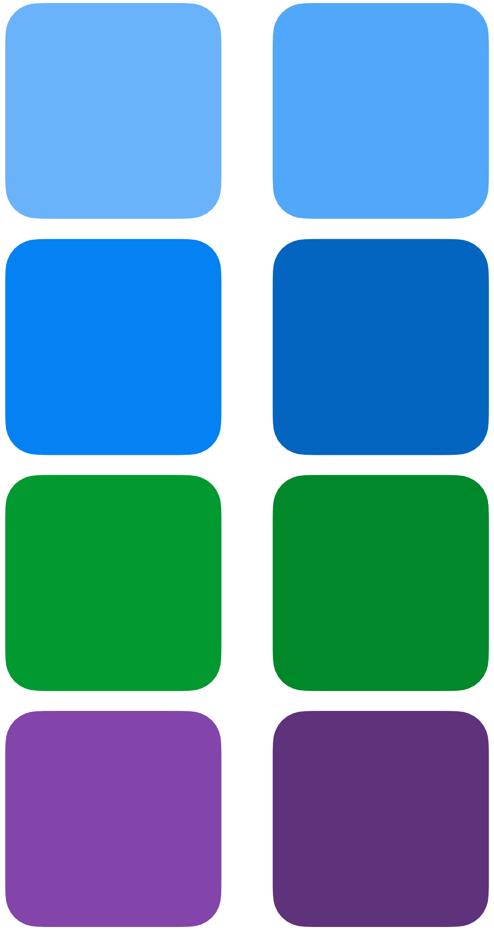
- It is unclear whether aforementioned pre-text tasks really enhance the representation quality

- What do we want from the learned representations?
    - Invariant mapping: representations should be stable for an slightly transformed version of an image
    - Semantic Similarity: semantically related images should be close to each other

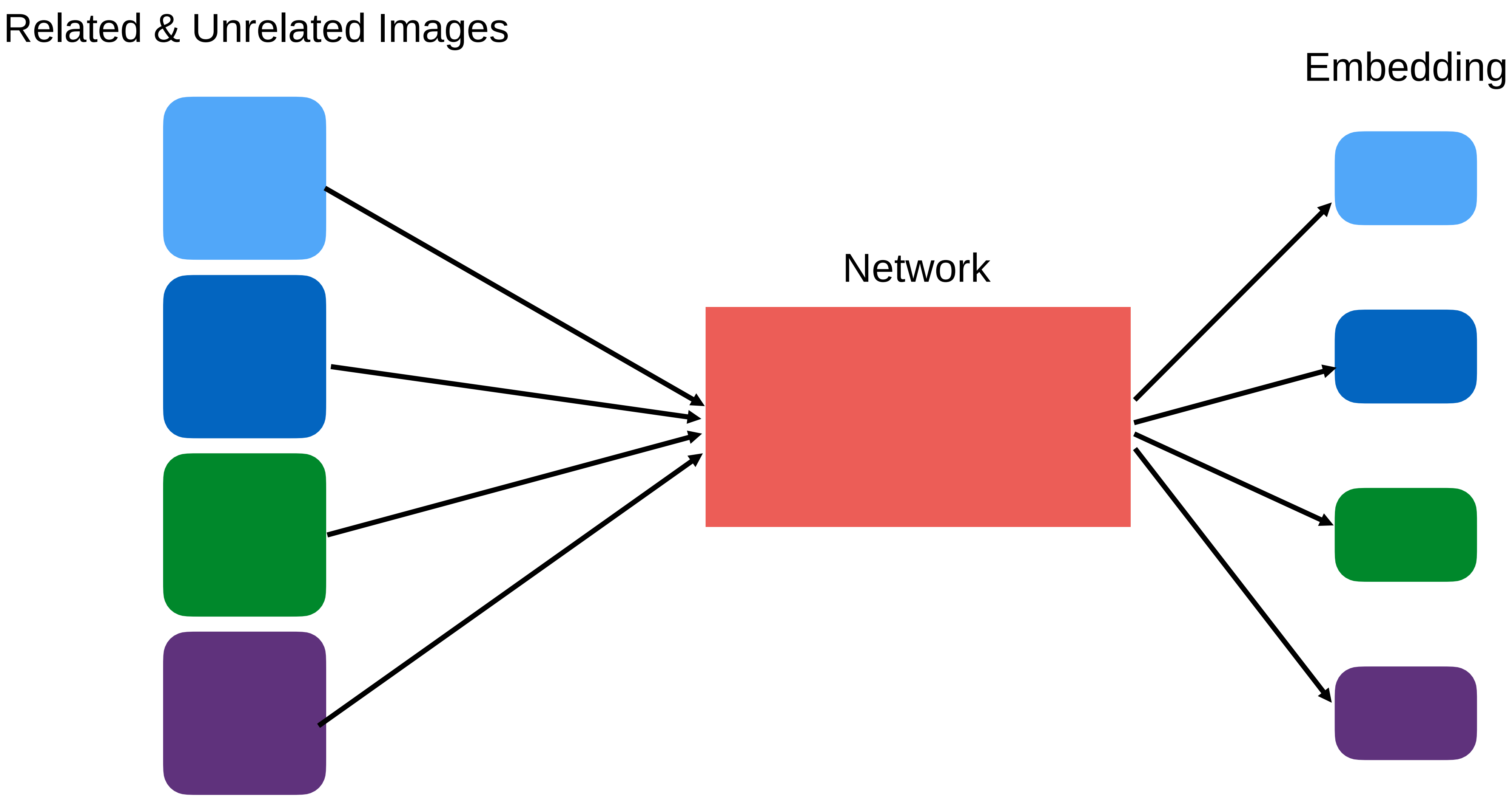
Recent research trends: Contrastive Learning

Contrastive Learning

Groups of Related & Unrelated Images



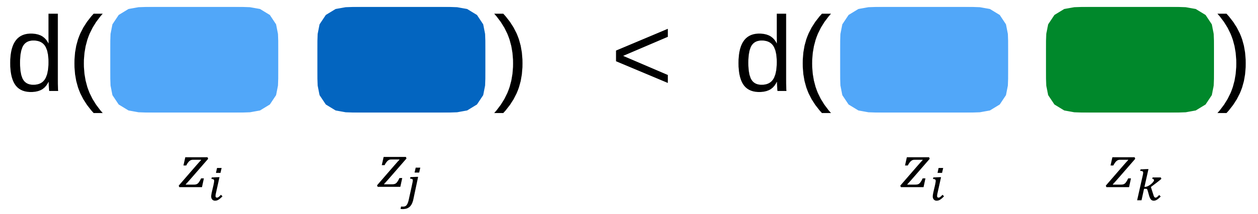
Contrastive Learning



Loss Function

Embeddings from related images should be closer than

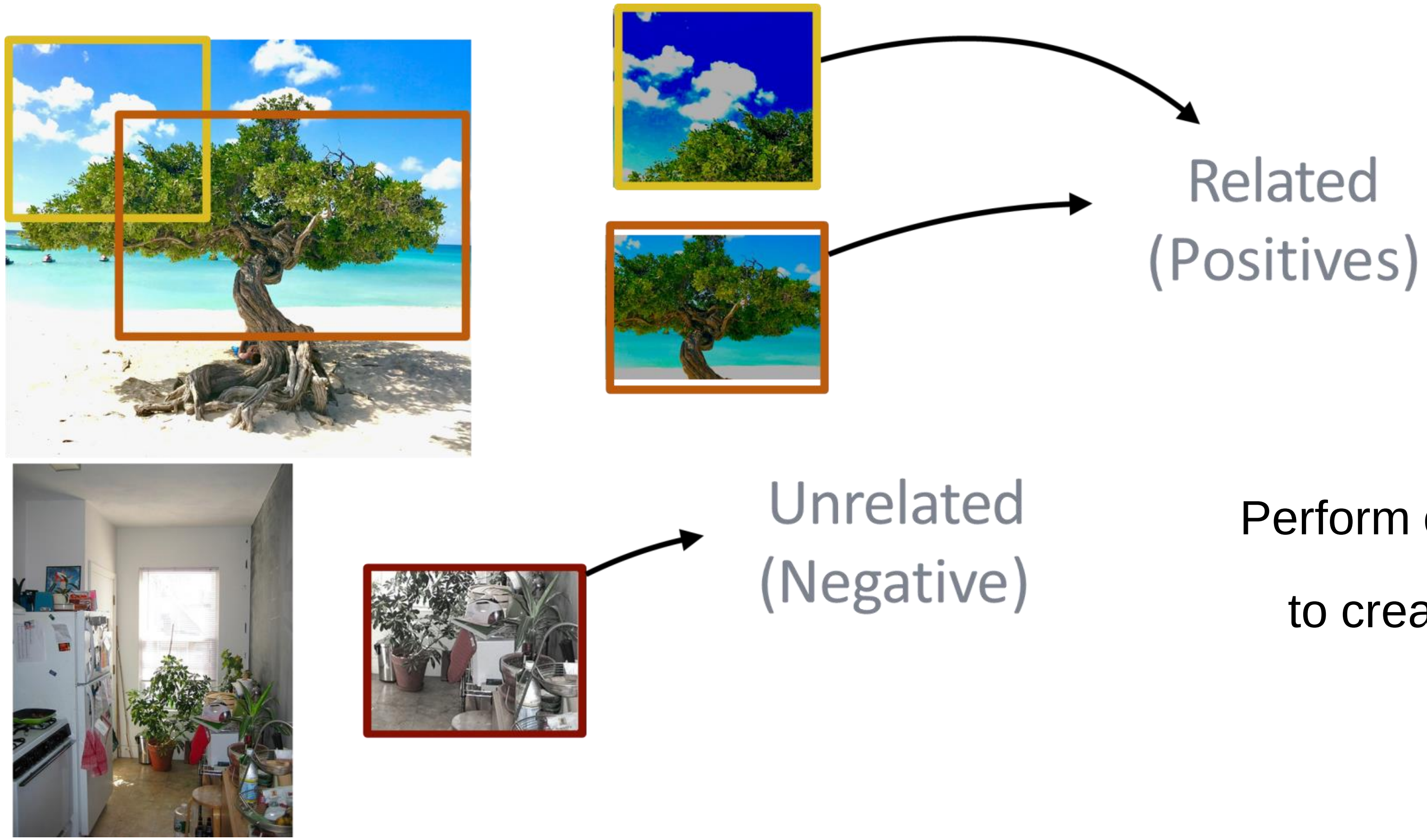
embeddings from unrelated images



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

Data augmentation for contrastive learning

- How to define which images are semantically “related” or “unrelated” without labels?



Perform **data augmentation**

to create related images

Contrastive Learning Framework

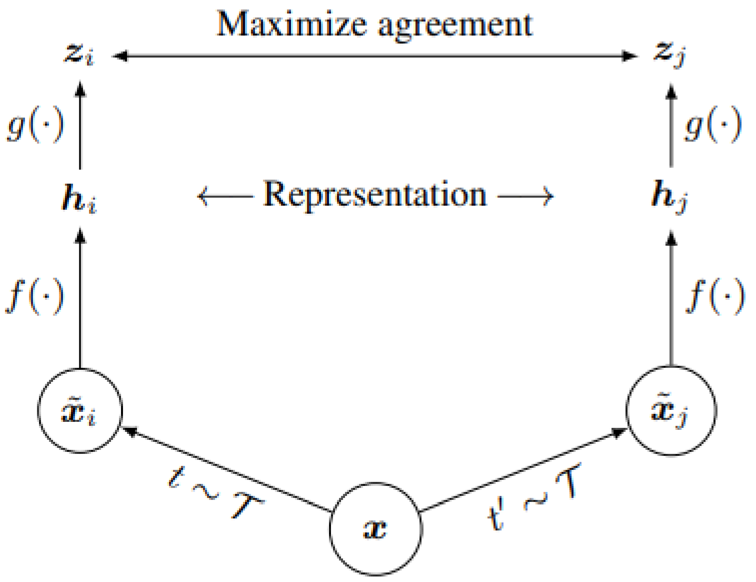


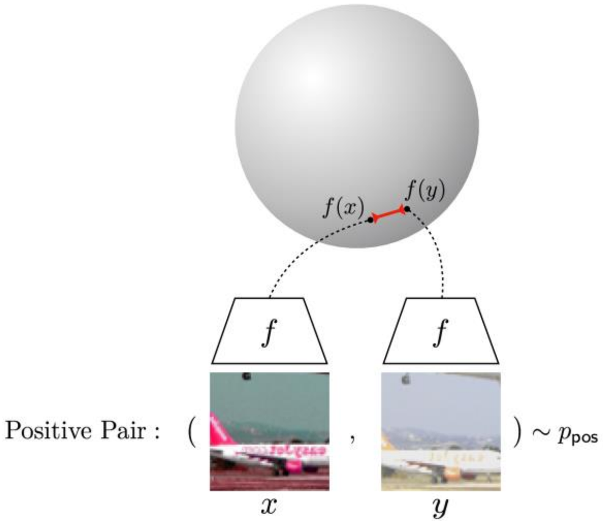
Figure 2. A simple framework for contrastive learning of visual representations. Two separate data augmentation operators are sampled from the same family of augmentations ($t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$) and applied to each data example to obtain two correlated views. A base encoder network $f(\cdot)$ and a projection head $g(\cdot)$ are trained to maximize agreement using a contrastive loss. After training is completed, we throw away the projection head $g(\cdot)$ and use encoder $f(\cdot)$ and representation $\boldsymbol{h}$ for downstream tasks.

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 $g(\cdot)$

Problems of Contrastive Learning

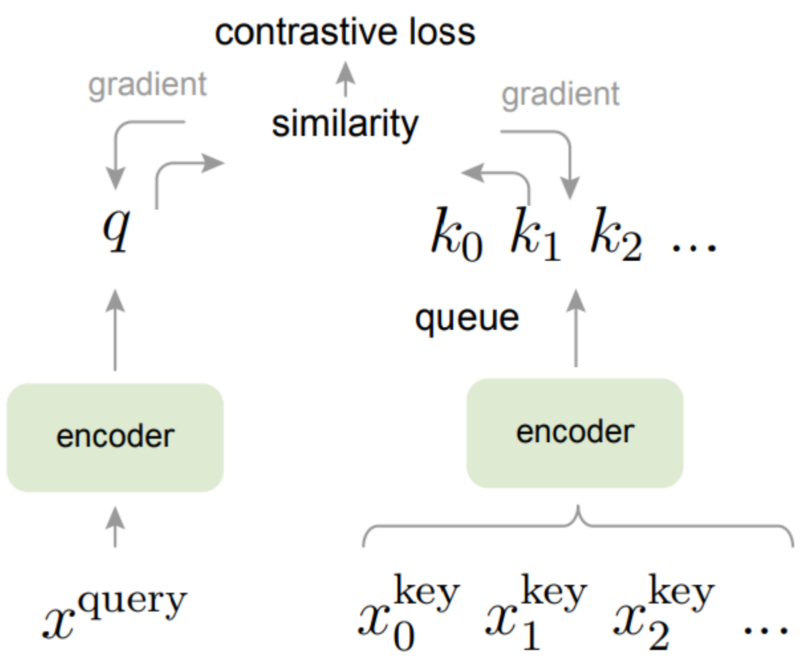
- Unstable training due to the moving targets
  - If f(x) moves closer to f(y), f(y) moves to the different locations since they share the identical network



- Solutions
  - PIRL, SimCLR: Use a lot of negative samples to restrict the representation's movement and avoid trivial solution
  - MoCo: Use a fixed network for the target

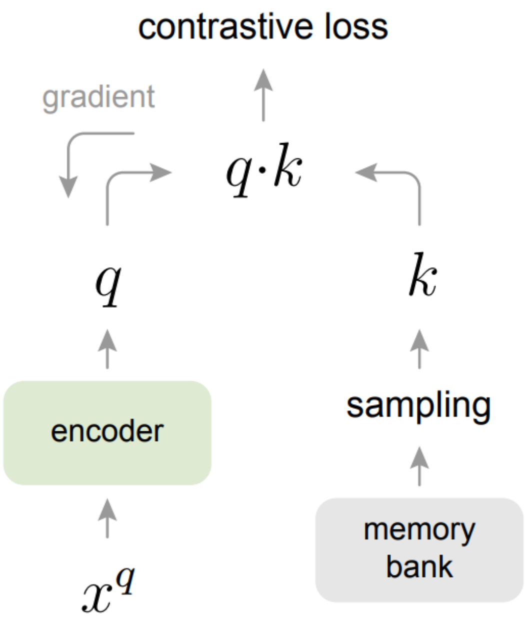
Using a lot of negative samples

- SimCLR



- Just use a huge batch size

- PIRL



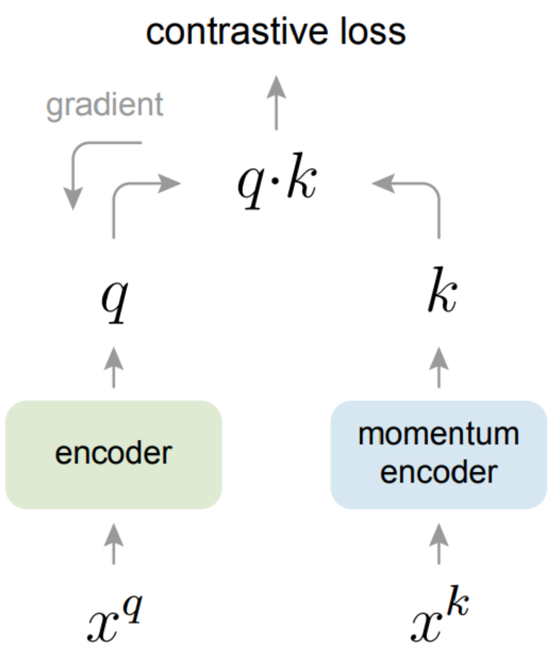
- Use memory-bank to store feature of negative samples.

Chen et al., SimCLR: A simple framework for contrastive learning of visual representations. ICML, 2020

Misra et al., PIRL: Self-supervised learning of pretext-invariant representations., CVPR, 2020

Using a target network

- MoCo

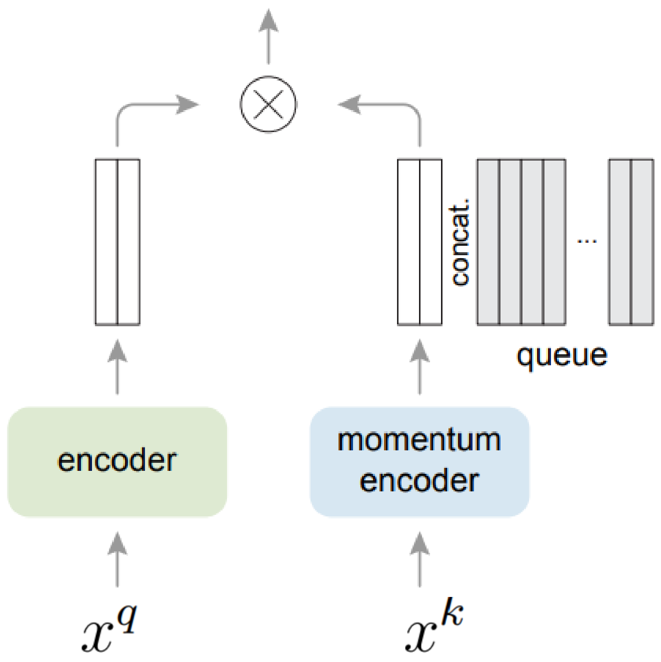


(c) MoCo

- Use a momentum encoder to fix the negative samples

$$\theta_{\mathrm{k}} \leftarrow m\theta_{\mathrm{k}} + (1-m)\theta_{\mathrm{q}}$$

- MoCo v2



- Momentum Encoder + Memory Bank
- This architecture is common in reinforcement learning literature

He et al., MoCo: Momentum contrast for unsupervised visual representation learning., CVPR, 2020

Ingredients for successful training

- Data Augmentations

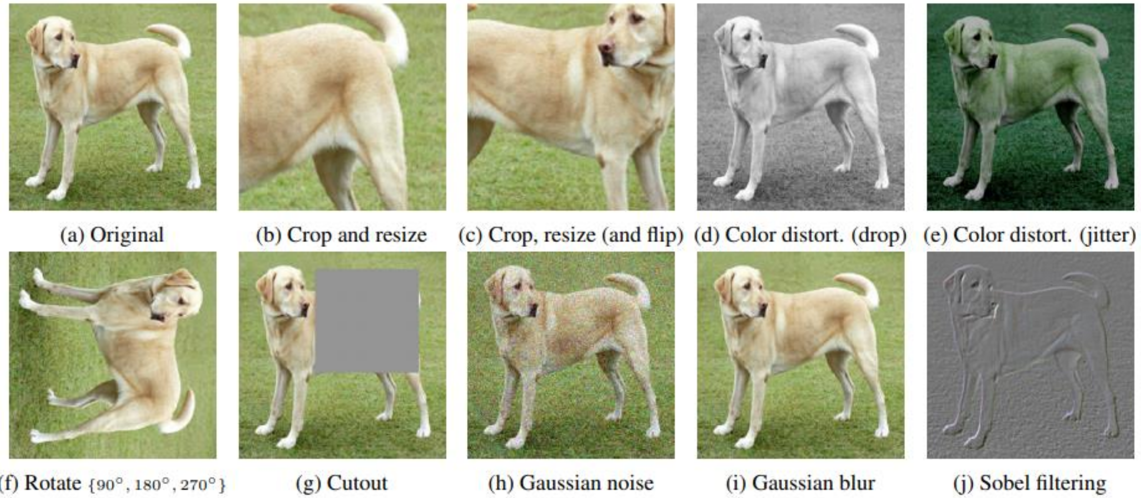


Figure 4. Illustrations of the studied data augmentation operators. Each augmentation can transform data stochastically with some internal parameters (e.g. rotation degree, noise level). Note that we *only* test these operators in ablation, the *augmentation policy used to train our models* only includes *random crop (with flip and resize)*, *color distortion*, and *Gaussian blur*. (Original image cc-by: Von.grzanka)

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.

Chen et al., SimCLR: A simple framework for contrastive learning of visual representations. ICML, 2020

Ingredients for successful training

- Number of negative samples

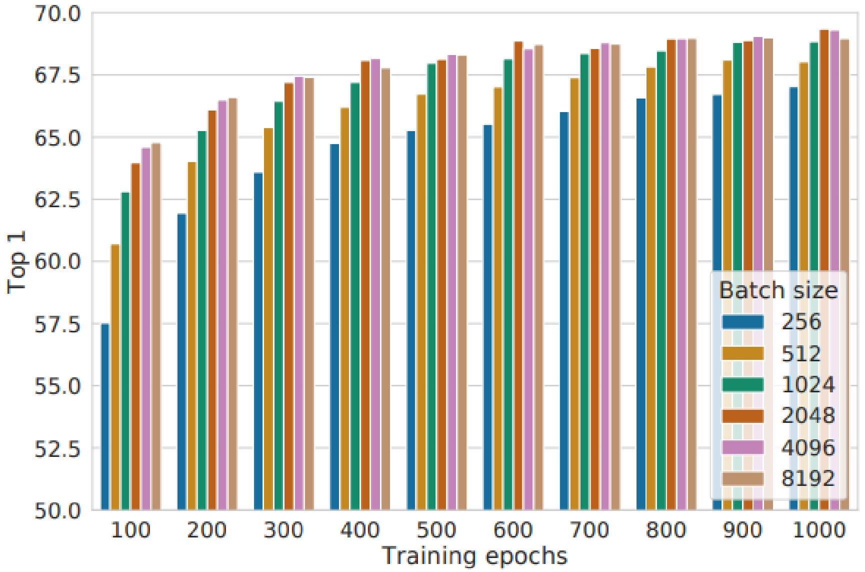


Figure 9. Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch.[10]

- Non-linear projection layer

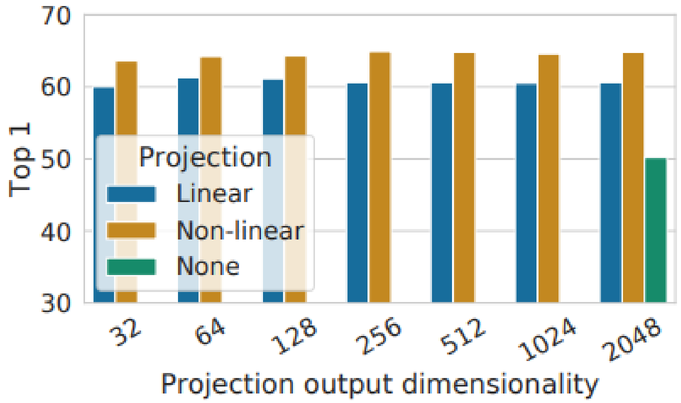


Figure 8. Linear evaluation of representations with different projection heads $g(\cdot)$ and various dimensions of $\boldsymbol{z} = g(\boldsymbol{h})$. The representation $\boldsymbol{h}$ (before projection) is 2048-dimensional here.

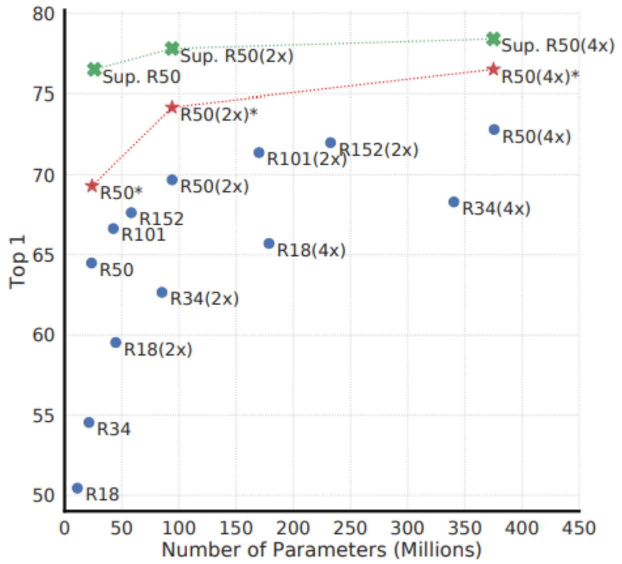
Chen et al., SimCLR: A simple framework for contrastive learning of visual representations. ICML, 2020

Results

- Competitive performance in classification and transfer learning even compared to supervised learning

	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

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.8 for experimental details and results with standard ResNet-50.



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 $(4\times)$	86	55.4	-
BigBiGAN	RevNet-50 $(4\times)$	86	61.3	81.9
AMDIM	Custom-ResNet	626	68.1	-
CMC	ResNet-50 $(2\times)$	188	68.4	88.2
MoCo	ResNet-50 $(4\times)$	375	68.6	-
CPC v2	ResNet-161 $(*)$	305	71.5	90.1
SimCLR (ours)	ResNet-50 $(2\times)$	94	74.2	92.0
SimCLR (ours)	ResNet-50 $(4\times)$	375	**76.5**	**93.2**

Table 6. ImageNet accuracies of linear classifiers trained on representations learned with different self-supervised methods.

Chen et al., SimCLR: A simple framework for contrastive learning of visual representations. ICML, 2020

BYOL

- If we are using the momentum encoder to fix the target network, is pushing away the negative samples are necessary? (we might push away the relevant images)
  - Without negative samples, the network can easily fall into trivial solutions (all images are mapped into the identical representation)

- BYOL has solved this issue by introducing a prediction network

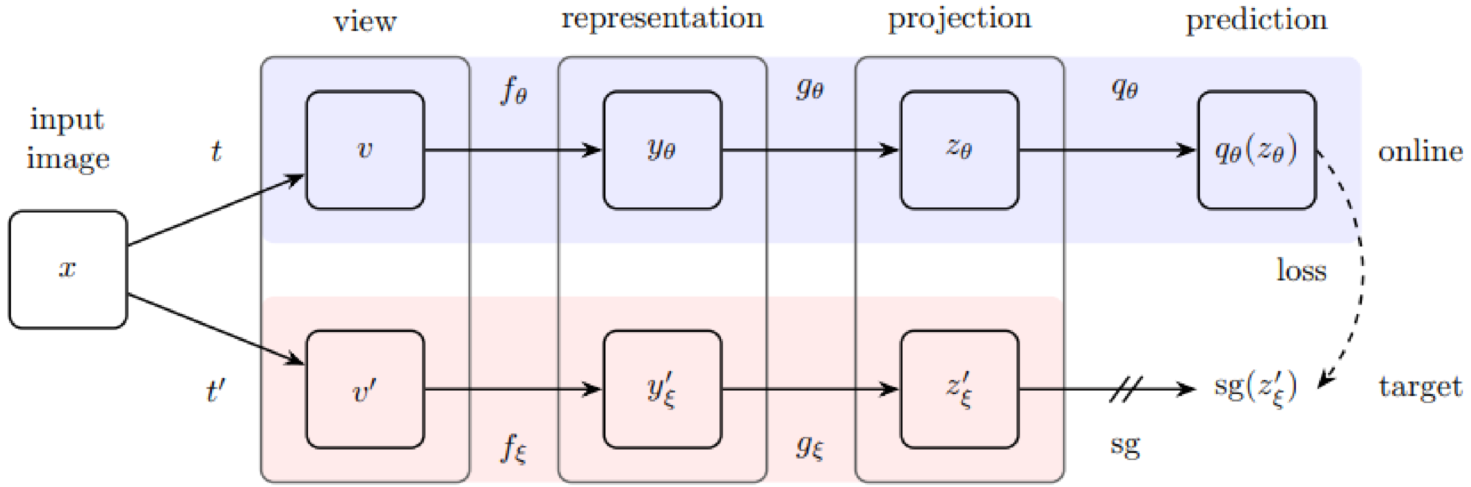
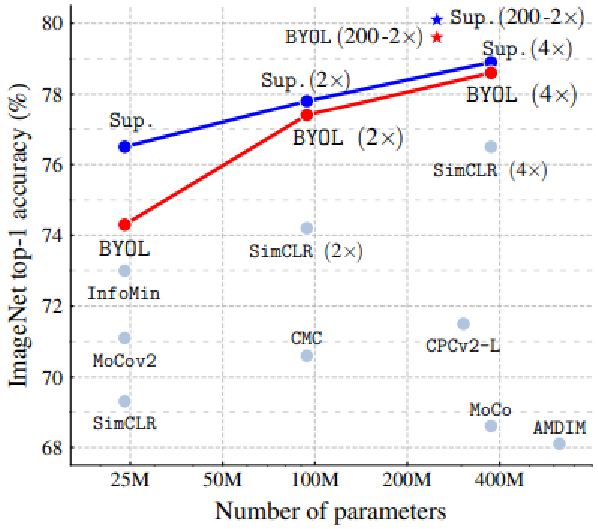


Figure 2: BYOL's architecture. BYOL minimizes a similarity loss between $q_\theta(z_\theta)$ and $\mathrm{sg}(z'_\xi)$, where θ are the trained weights, ξ are an exponential moving average of θ and sg means stop-gradient. At the end of training, everything but f_θ is discarded, and y_θ is used as the image representation.

- Only align loss

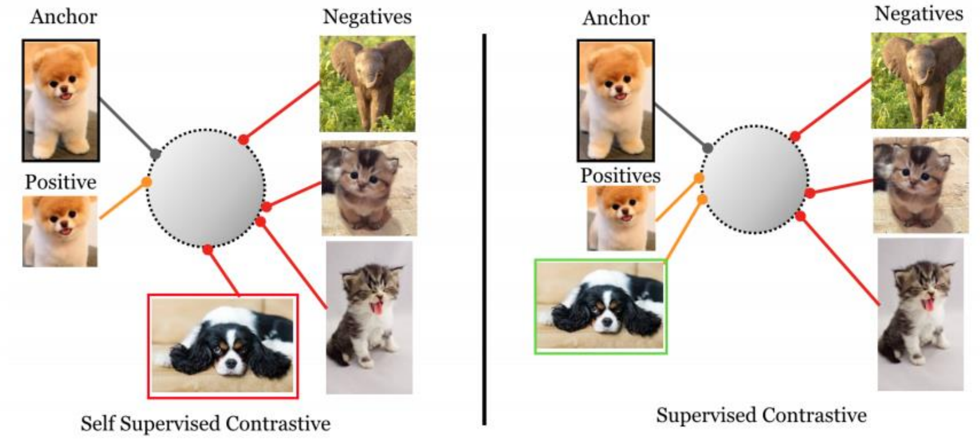
$$\mathcal{L}_{\theta,\xi} \triangleq \left\|\overline{q_\theta}(z_\theta) - \overline{z}'_\xi\right\|_2^2 = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z'_\xi\rangle}{\left\|q_\theta(z_\theta)\right\|_2 \cdot \left\|z'_\xi\right\|_2}.$$



Grill et al., Bootstrap your own latent: A new approach to self-supervised Learning. NeurIPS, 2020

Supervised Contrastive Learning

- Supervised contrastive learning achieved new state-of-the-art in image classification by aligning the related images with class label



$$\mathcal{L}^{sup}_{out} = \sum_{i \in I} \mathcal{L}^{sup}_{out,i} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\boxed{\exp\left(\boldsymbol{z}_i \boldsymbol{\cdot} \boldsymbol{z}_p / \tau\right)}}{\sum_{a \in A(i)} \exp\left(\boldsymbol{z}_i \boldsymbol{\cdot} \boldsymbol{z}_a / \tau\right)}$$

Images with same classes should be aligned together

Loss	Architecture	Augmentation	Top-1	Top-5
Cross-Entropy (baseline)	ResNet-50	MixUp [61]	77.4	93.6
Cross-Entropy (baseline)	ResNet-50	CutMix [60]	78.6	94.1
Cross-Entropy (baseline)	ResNet-50	AutoAugment [5]	78.2	92.9
Cross-Entropy (our impl.)	ResNet-50	AutoAugment [30]	77.6	95.3
SupCon	ResNet-50	AutoAugment [5]	**78.7**	**94.3**
Cross-Entropy (baseline)	ResNet-200	AutoAugment [5]	80.6	95.3
Cross-Entropy (our impl.)	ResNet-200	Stacked RandAugment [49]	80.9	95.2
SupCon	ResNet-200	Stacked RandAugment [49]	**81.4**	**95.9**
SupCon	ResNet-101	Stacked RandAugment [49]	80.2	94.7

Table 3: Top-1/Top-5 accuracy results on ImageNet for AutoAugment [5] with ResNet-50 and for Stacked RandAugment [49] with ResNet-101 and ResNet-200. The baseline numbers are taken from the referenced papers, and we also re-implement cross-entropy.

Loss	Architecture	rel. mCE	mCE
Cross-Entropy (baselines)	AlexNet [28]	100.0	100.0
	VGG-19+BN [44]	122.9	81.6
	ResNet-18 [17]	103.9	84.7
Cross-Entropy (our implementation)	ResNet-50	96.2	68.6
	ResNet-200	69.1	52.4
Supervised Contrastive	ResNet-50	**94.6**	**67.2**
	ResNet-200	**66.5**	**50.6**

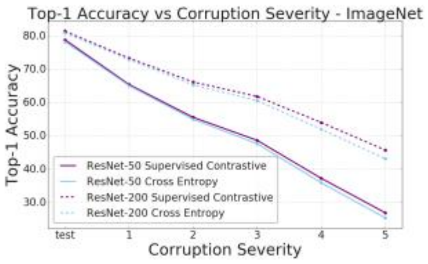


Figure 3: Training with supervised contrastive loss makes models more robust to corruptions in images. **Left**: Robustness as measured by Mean Corruption Error (mCE) and relative mCE over the ImageNet-C dataset [19] (lower is better). **Right**: Mean Accuracy as a function of corruption severity averaged over all various corruptions. (higher is better).

Khosla et al., Supervised Contrastive Learning. NeurIPS, 2020

Future Research Directions

SimCLR v2: Self-Supervised Model as a Semi-Supervised Learner

- Self-supervised model is a strong semi-supervised learner

Table 1: Top-1 accuracy of fine-tuning SimCLRv2 models (on varied label fractions) or training a linear classifier on the representations. The supervised baselines are trained from scratch using all labels in 90 epochs. The parameter count only include ResNet up to final average pooling layer. For fine-tuning results with 1% and 10% labeled examples, the models include additional non-linear projection layers, which incurs additional parameter count (4M for 1× models, and 17M for 2× models). See Table H.1 for Top-5 accuracy.

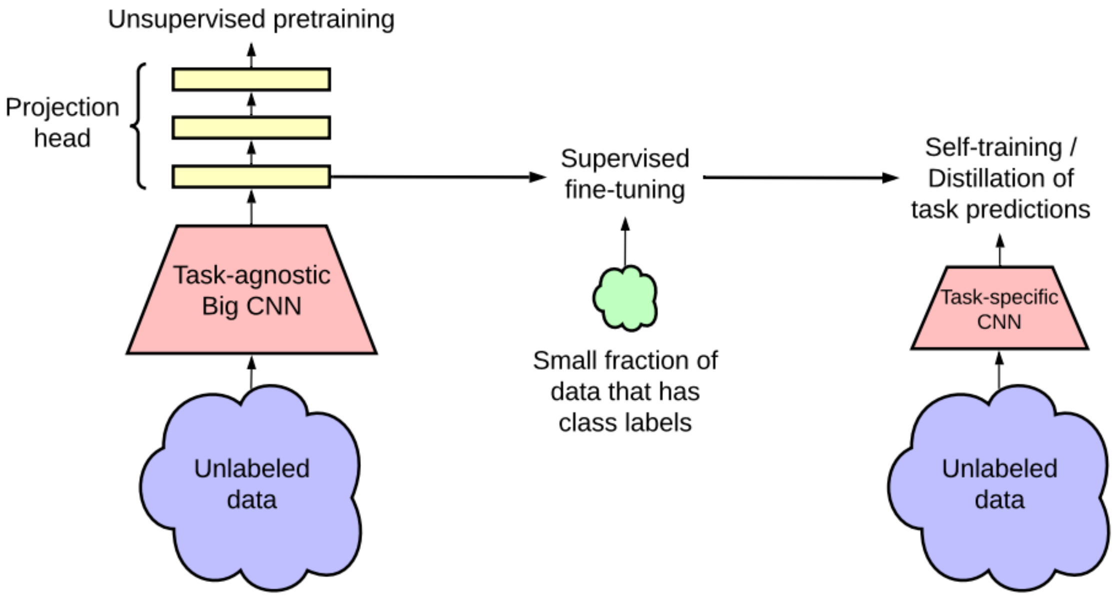
Depth	Width	Use SK [28]	Param (M)	Fine-tuned on			Linear eval	Supervised
				1%	10%	100%		
50	1×	False	**24**	**57.9**	**68.4**	**76.3**	**71.7**	**76.6**
		True	35	64.5	72.1	78.7	74.6	78.5
	2×	False	94	66.3	73.9	79.1	75.6	77.8
		True	140	70.6	77.0	81.3	77.7	79.3
101	1×	False	43	62.1	71.4	78.2	73.6	78.0
		True	65	68.3	75.1	80.6	76.3	79.6
	2×	False	170	69.1	75.8	80.7	77.0	78.9
		True	257	73.2	78.8	82.4	79.0	80.1
152	1×	False	58	64.0	73.0	79.3	74.5	78.3
		True	89	70.0	76.5	81.3	77.2	79.9
	2×	False	233	70.2	76.6	81.1	77.4	79.1
		True	354	74.2	79.4	82.9	79.4	80.4
152	3×	True	**795**	**74.9**	**80.1**	**83.1**	**79.8**	**80.5**

Chen et al., Big self-supervised models are strong semi-supervised learners. NeurIPS, 2020

SimCLR v2: Self-Supervised Model as a Semi-Supervised Learner

- Knowledge-distillation with a fine-tuned self-supervised model even surpassed the state-of-the-art supervised methods by a huge margin

- Self-Distillation Procedure



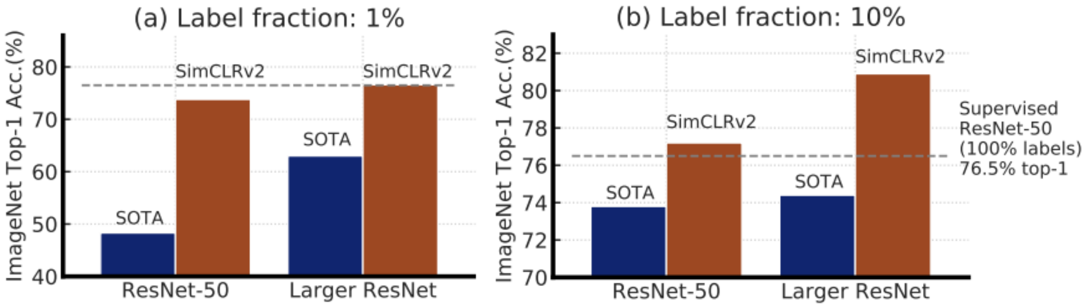
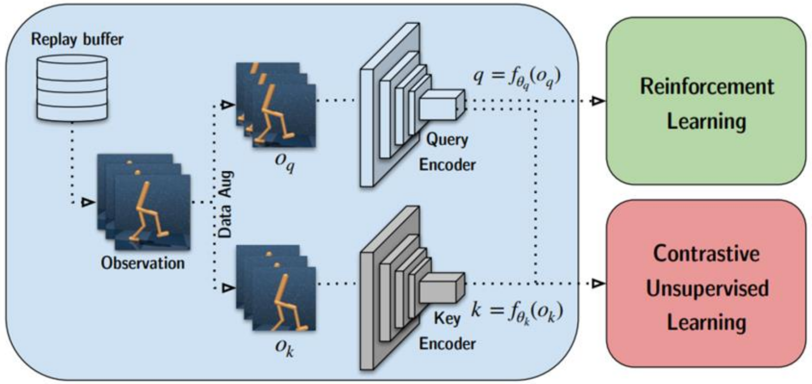


Figure 2: Top-1 accuracy of previous state-of-the-art (SOTA) methods [1, 2] and our method (SimCLRv2) on ImageNet using only 1% or 10% of the labels. Dashed line denotes fully supervised ResNet-50 trained with 100% of labels. Full comparisons in Table 3.

Chen et al., Big self-supervised models are strong semi-supervised learners. NeurIPS, 2020

Application to various other fields

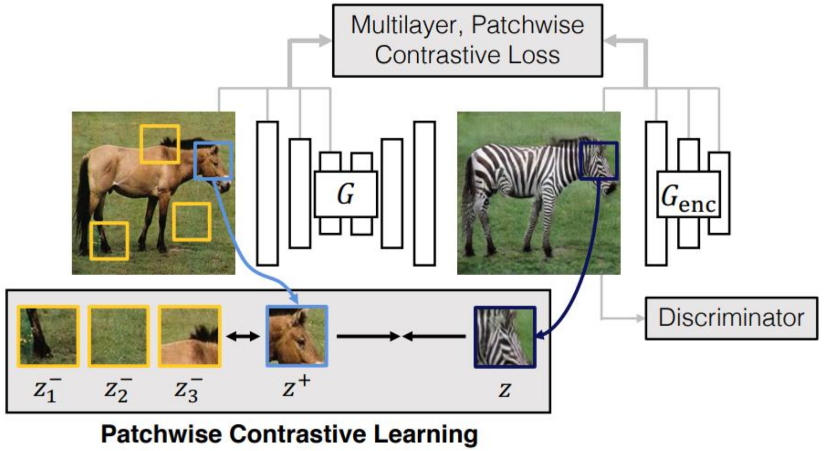
- Sample-efficient reinforcement learning with increased visual perception

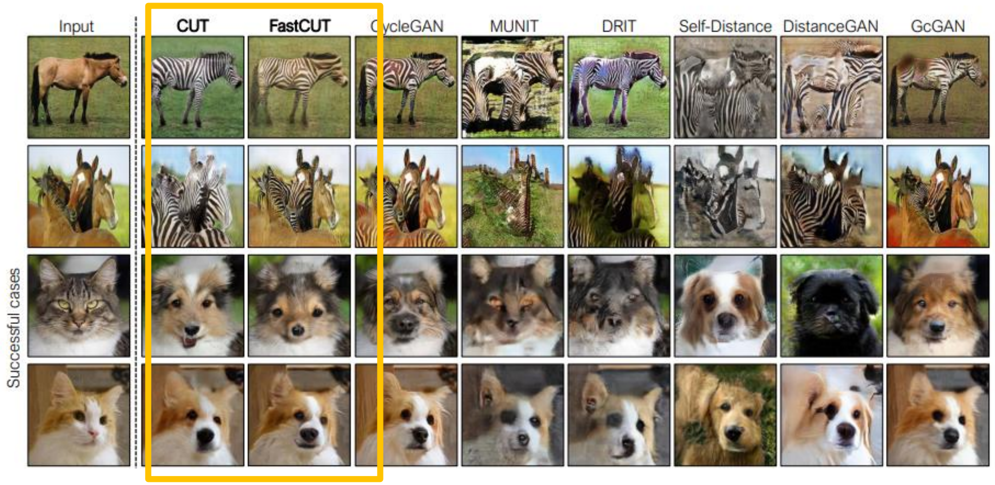


500K step scores	CURL	PlaNet	Dreamer	SAC+AE	SLACv1	Pixel SAC	State SAC
Finger, spin	**926 ± 45**	561 ± 284	796 ± 183	884 ± 128	673 ± 92	179 ± 166	923 ± 21
Cartpole, swingup	**841 ± 45**	475 ± 71	762 ± 27	735 ± 63	-	419 ± 40	848 ± 15
Reacher, easy	**929 ± 44**	210 ± 390	793 ± 164	627 ± 58	-	145 ± 30	923 ± 24
Cheetah, run	518 ± 28	305 ± 131	570 ± 253	550 ± 34	**640 ± 19**	197 ± 15	795 ± 30
Walker, walk	**902 ± 43**	351 ± 58	897 ± 49	847 ± 48	842 ± 51	42 ± 12	948 ± 54
Ball in cup, catch	**959 ± 27**	460 ± 380	879 ± 87	794± 58	852 ± 71	312± 63	974 ± 33
100K step scores							
Finger, spin	**767 ± 56**	136 ± 216	341 ± 70	740 ± 64	693 ± 141	179 ± 66	811±46
Cartpole, swingup	**582±146**	297±39	326±27	311±11	-	419±40	835±22
Reacher, easy	**538±233**	20±50	314±155	274±14	-	145±30	746±25
Cheetah, run	299 ±48	138±88	235± 137	267±24	**319±56**	197±15	616±18
Walker, walk	**403±24**	224±48	277±12	394±22	361±73	42±12	891±82
Ball in cup, catch	**769 ± 43**	0 ± 0	246 ± 174	391± 82	512 ± 110	312± 63	746±91

Srinivas et al., Curl: Contrastive unsupervised representations for reinforcement learning. ICML, 2020

- One-sided image-to-image translation with patch-wise contrastive learning





Park et al., Contrastive Learning for Unpaired Image-to-Image Translation. ECCV, 2020

Q & A