Università di Genova | MaLGa

Self-supervised learning

19/03/2025

Vito Paolo Pastore

Deep learning a.y. 2024/2025

Credits

These slides have been built upon the following tutorials or lecture:

- https://cs231n.stanford.edu/slides/2022/lecture_14_jiajun.pdf
- https://gidariss.github.io/self-supervised-learning-cvpr2021/

Some slides from:

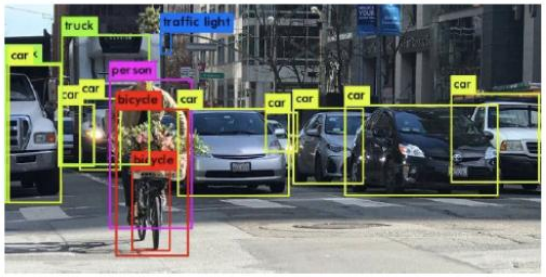
- Vittorio Murino
- Pietro Morerio

UniGe | MaLGa

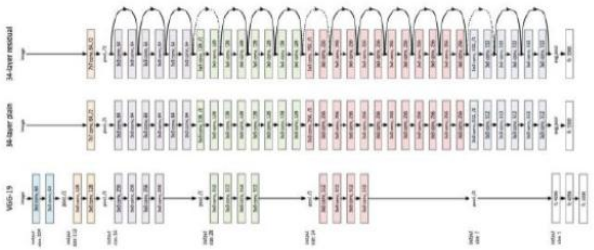
Self-supervised learning

Motivation

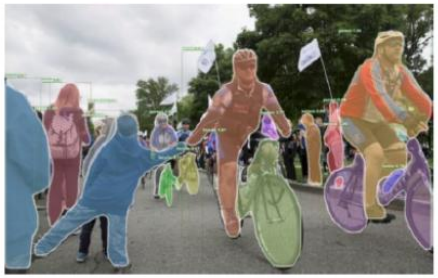
Deep neural networks and supervised learning are a powerful tool



Faster R-CNN, Ren et al. 15



ResNet, He et al. 16



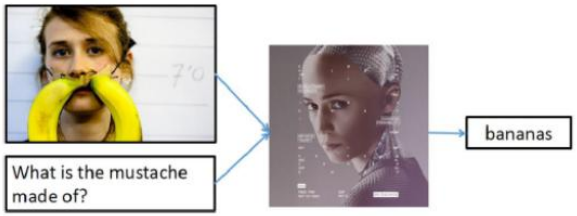
Mask R-CNN, He et al. 17

(a) Mobile phone query

(b) Retrieved image of same place

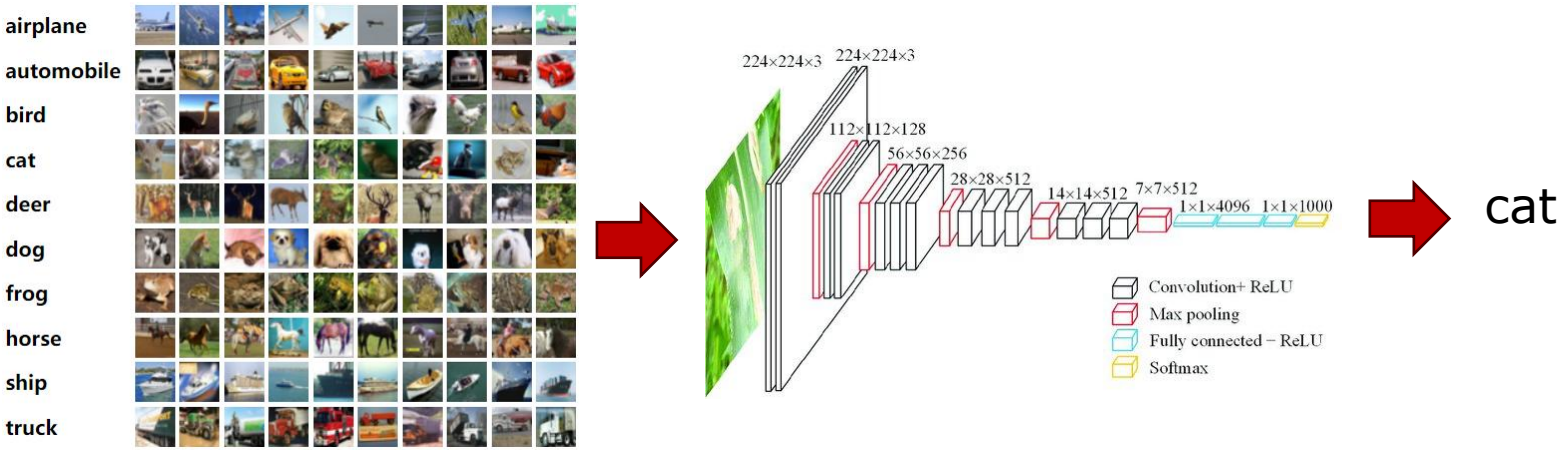
NetVlad: place recognition, Arandjelović et al. 16

Human pose estimation, Newel et al. 17



Visual Question Answering, Antol et al 15

Typical supervised learning framework



- Predefine a set of concepts to learn;
- Collect diverse and large number of examples for each of them;
- Train a deep model on such examples

However, in the real world (Cons of supervised learning)

Difficult to acquire and curate large human-annotated datasets

- Requires intense human labor
  - annotating + cleaning raw data
- Time consuming and expensive
- Error prone (human mistakes)

Annotating such image: ~1.5h

However, in the real world (Cons of supervised learning)

Difficult to keep the pace with an ever-changing world

Men's fashion trends 1980-1989

- Continuous data distribution shift (e.g., fashion trends in the example)
- Infeasible to launch large annotation campaigns each time

Source Image

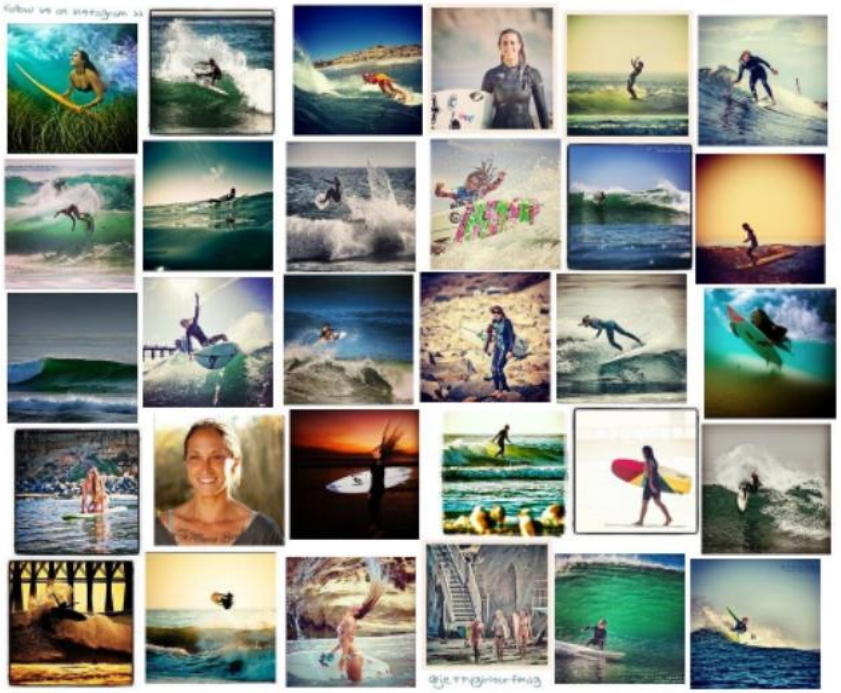
However, in the real world (Cons of supervised learning)

Difficult to keep the pace with an ever-changing world

- Continuous data distribution shift (e.g., fashion trends in the example)
- Sensor specs are frequently upgraded
- Infeasible to launch large annotation campaigns each time

Source Image

Alternative: Exploiting raw data

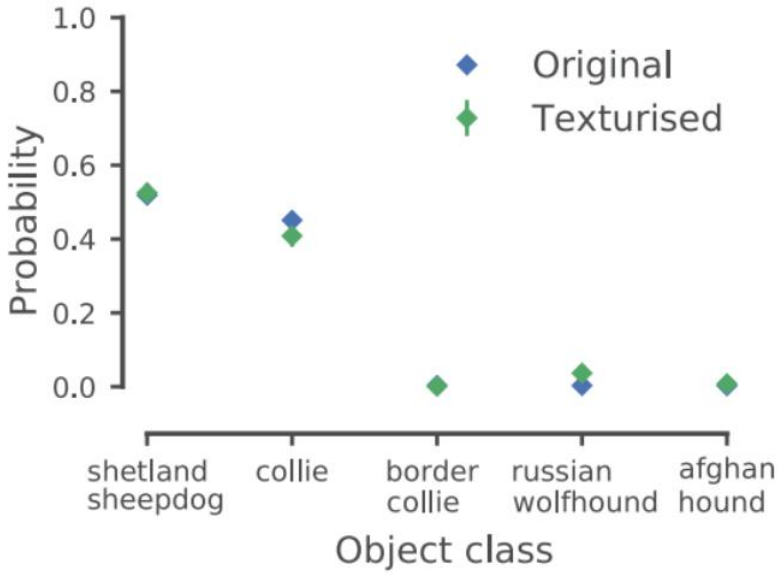


- Supervised learning generally requires large amount of **carefully** labeled data
- Acquiring raw unlabeled data is usually easy
- However, typical supervised methods cannot exploit them

Other potential issues related to labeling: bias

Original Image

Texturised samples



VGG-16 preditions on original and artificially texturised images.

L.A. Gatys et al., Texture and art with deep neural networks, Neurobiology 2017

Even with large amounts of data, supervised learning still has several blind-spots in terms of learning useful and rich representations.

Another example

(a) Texture image

81.4%	**Indian elephant**
10.3%	indri
8.2%	black swan

(b) Content image

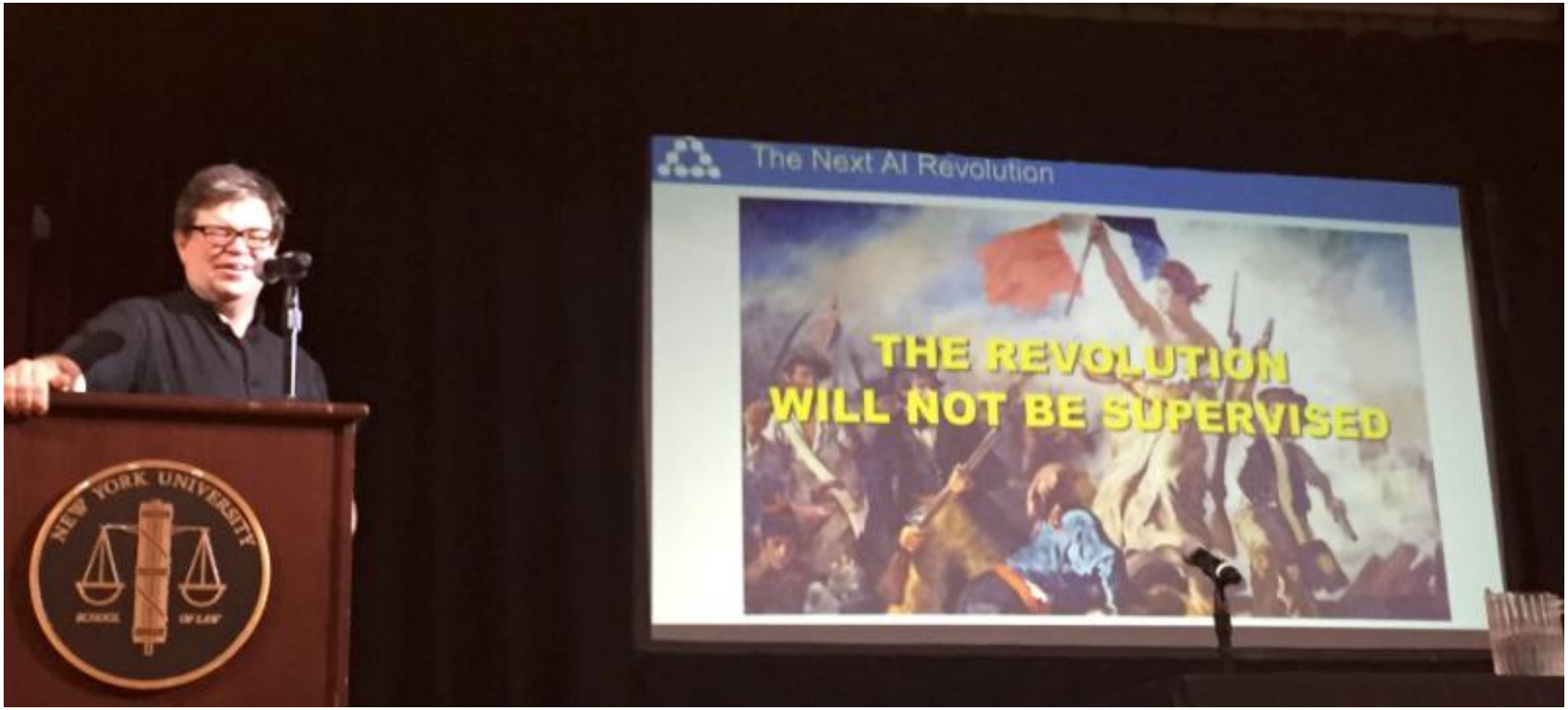
71.1%	**tabby cat**
17.3%	grey fox
3.3%	Siamese cat

(c) Texture-shape cue conflict

63.9%	**Indian elephant**
26.4%	indri
9.6%	black swan

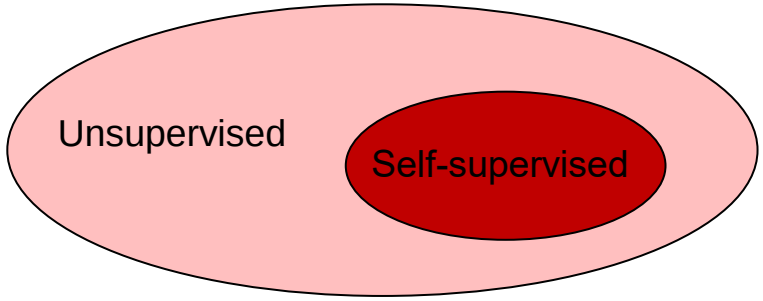
Classification predictions of a ResNet-50 trained on ImageNet

A quote


The Next AI Revolution
THE REVOLUTION
WILL NOT BE SUPERVISED
NEW YORK UNIVERSITY

Self-Supervised Learning (SSL): general concepts (1)

- A form of unsupervised learning where the data (not the human) provides the supervision signal
- Usually, define a pretext task for which the network is forced to learn what we really care about
- For most pretext tasks, a part of the data is withheld and the network has to predict it
- The features/representations learned on the pretext task are subsequently used for a different downstream task, usually where some annotations are available



SSL: general concepts (2)

General:

- hide part of the data;
- get a network to predict the hidden part given the available part.

Define an auxiliary task that, in order to be solved, needs some semantic representation.

A great example from NLP (word2vec)

Input: The man went to the $[MASK]_1$. He bought a $[MASK]_2$ of milk .
Labels: $[MASK]_1$ = store; $[MASK]_2$ = gallon

Missing word prediction task.

Sentence A = The man went to the store.
Sentence B = He bought a gallon of milk.
Label = IsNextSentence

Sentence A = The man went to the store.
Sentence B = Penguins are flightless.
Label = NotNextSentence

Next sentence prediction task.

T. Mikolov et al., Efficient estimation of word representations in vector space, ArXiv 2013
T. Mikolov et al., Distributed representations of words and phrases and their compositionality, NeurIPS 2013
J. Devlin, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, ArXiv 2018

What exactly is self-supervision?

In self-supervised learning, the supervisory signal comes from the data itself.

General:

- hide part of the data;
- get a network to predict the hidden part given the available part.

Define a pretext task that, in order to be solved, needs some semantic representation.

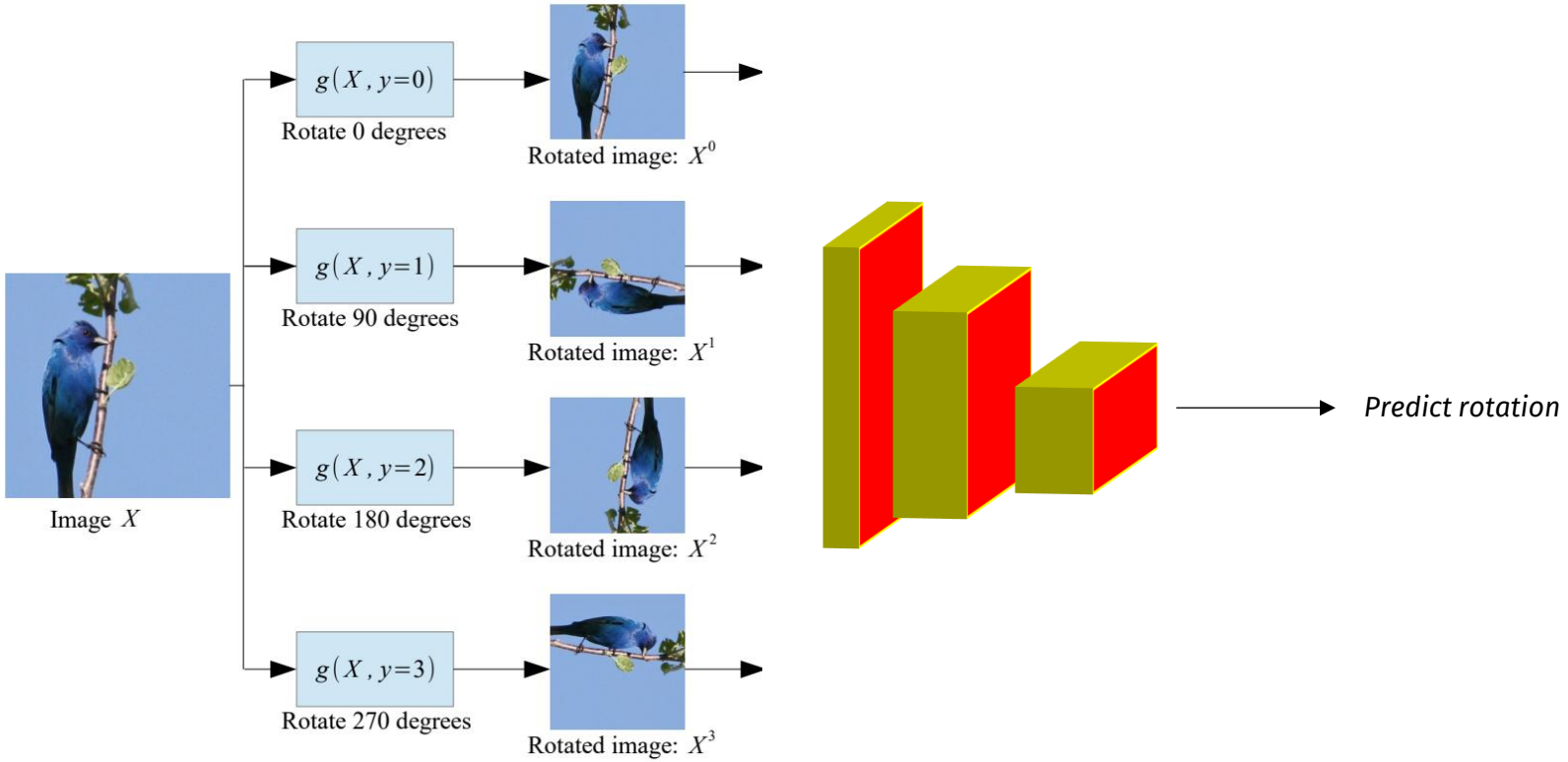
SSL pipeline

Step 1

- **Pretext task(s):**
  - Pre-designed task to learn features
  - Lots of data
  - No human annotation
- **Downstream task(s):**
  - Main task(s) to be solved
  - Possibly few data
  - Human annotation needed
    - Also used to evaluate the quality of features learned by the self-supervised pretext task
  - Can be the same task

[1] Longlong Jing, Yingli Tian, *Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey*, https://arxiv.org/abs/1902.06162, 2019

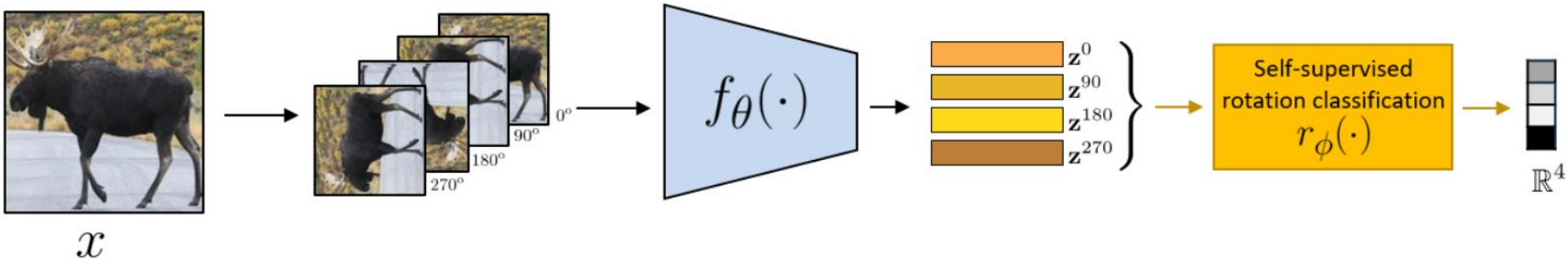
An example of pre-text tasks: Rotation



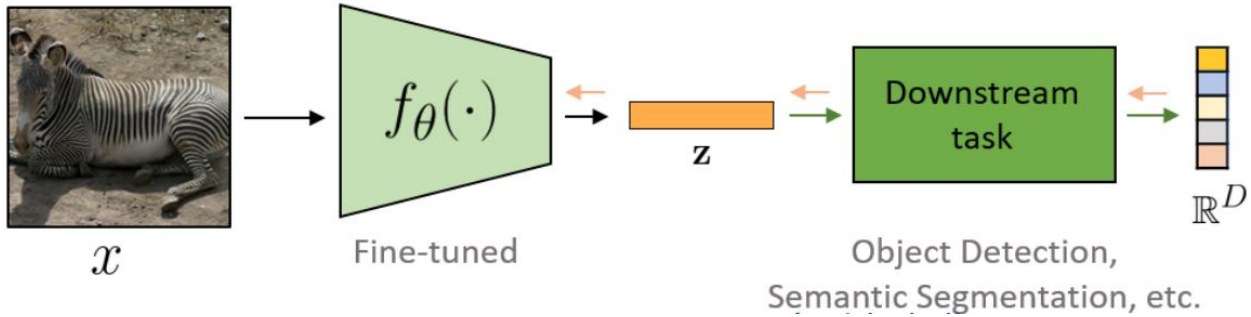
S. Gidaris et al., Unsupervised Representation Learning by Predicting Image Rotations, ICLR 2018

Self-supervised learning pipeline

Stage 1: Train network on pretext task (without human labels)



Stage 2: Fine-tune network for new task with fewer labels



Karate Kid and Self-Supervised Learning

The Karate Kid (1984)

Karate kid example: stage 1

Mr. Miyagi = Deep Learning Practitioner
Daniel LaRusso = ConvNet
daily chores =pretext tasks
learning karate= downstream task (stage 2)

Karate kid example: stage 2

Fine-tuning skills for karate moves

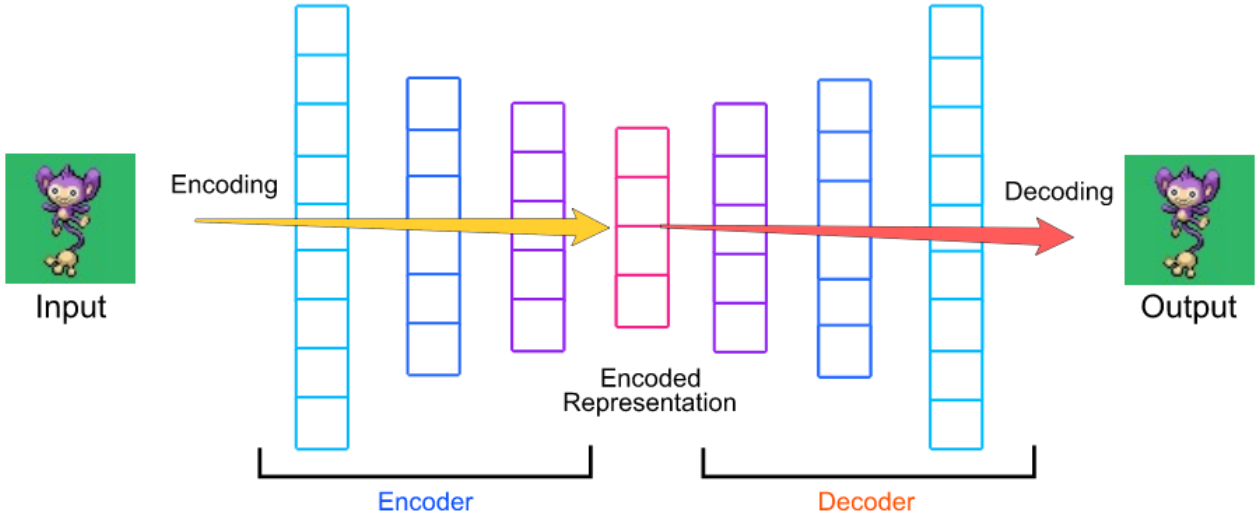
UniGe | MaLGa

Self-supervised learning: Pretext tasks

Using images

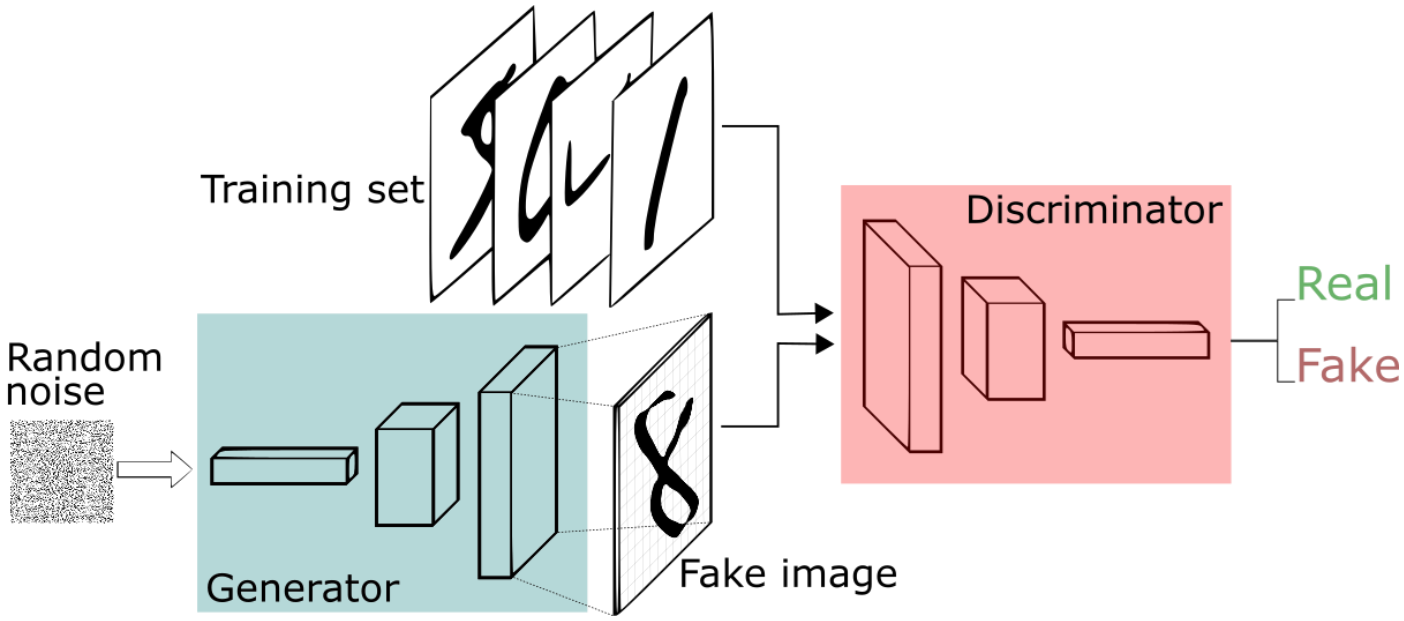
Generation-based pretext task

Autoencoder
Generation-based pretext tasks



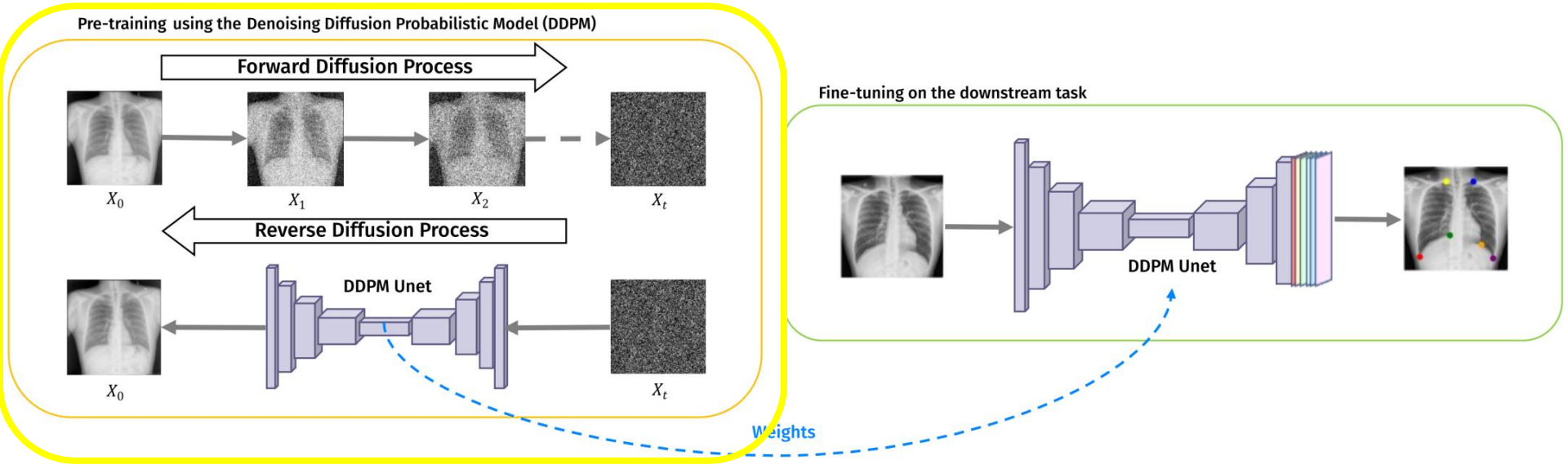
G.E. Hinton and R.R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, 2006

GAN – Generative Adversarial Networks
Generation-based pretext tasks



[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, A. Courville, Y. Bengio, “Generative adversarial nets”, *NIPS 2014*.

Diffusion model
Generation-based pretext tasks

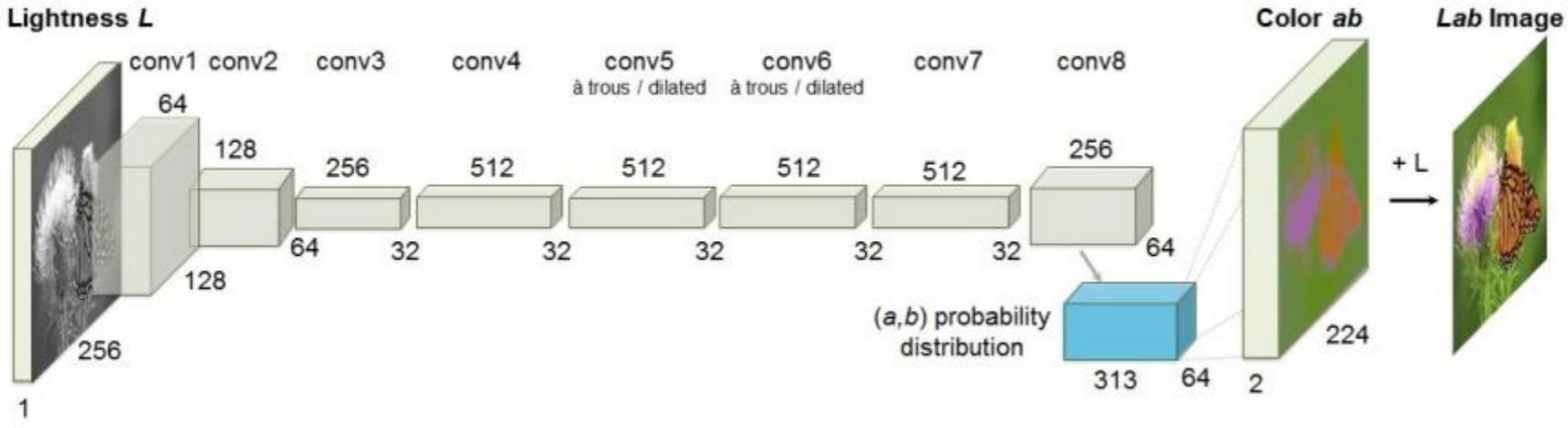


Di Via, R., Odone, F., & Pastore, V. P. (2024). Self-supervised pre-training with diffusion model for few-shot landmark detection in x-ray images. *arXiv preprint arXiv:2407.18125*.

Image colorization
Generation-based pretext tasks

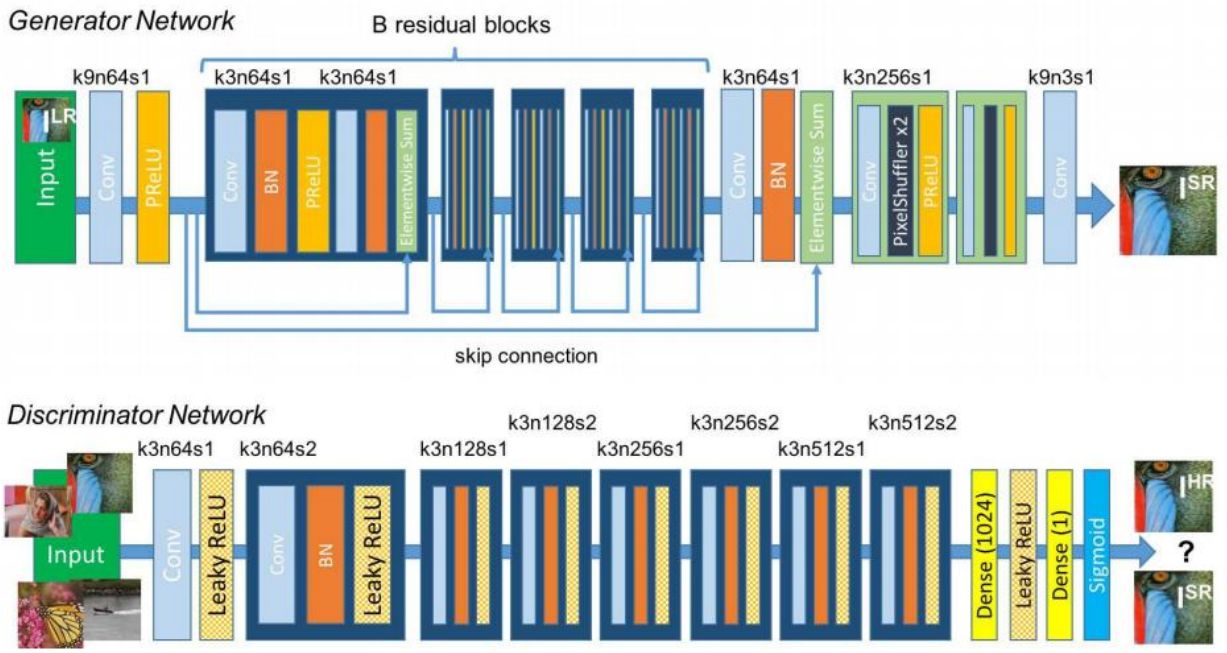
Train network to predict pixel colour from a monochrome input

Classification with class re-balancing based on ImageNet statistics



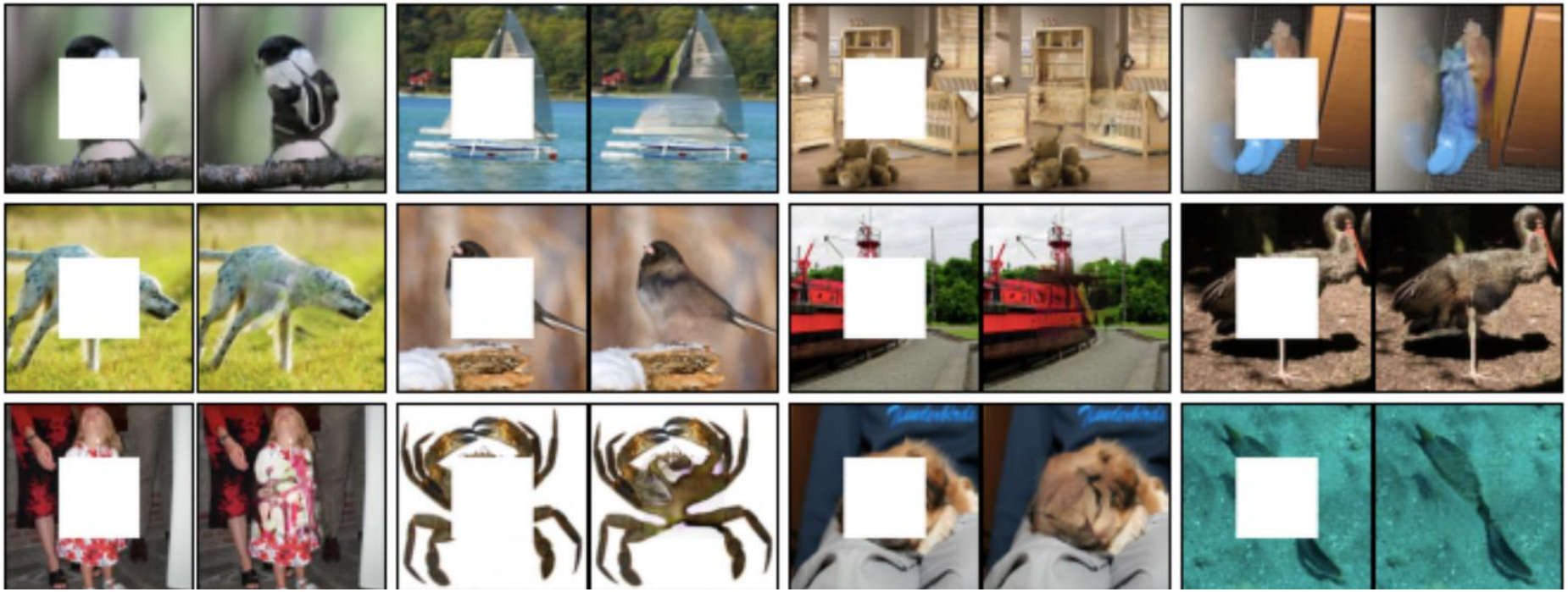
[8] R. Zhang, P. Isola, and A. A. Efros, “Colorful image colorization,” *ECCV 2016*.

GAN - Super resolution
Generation-based pretext tasks



[6] C. Ledig et al., "Photo-realistic single image super-resolution using a generative adversarial network," *CVPR 2017*.

GAN - Super resolution
Generation-based pretext tasks



Pathak, Deepak, et al. "Context encoders: Feature learning by inpainting." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Video
Generation-based pretext tasks

Video generation with GANs - VideoGAN [9]

Video colorization [10]

Future frame generation [11]

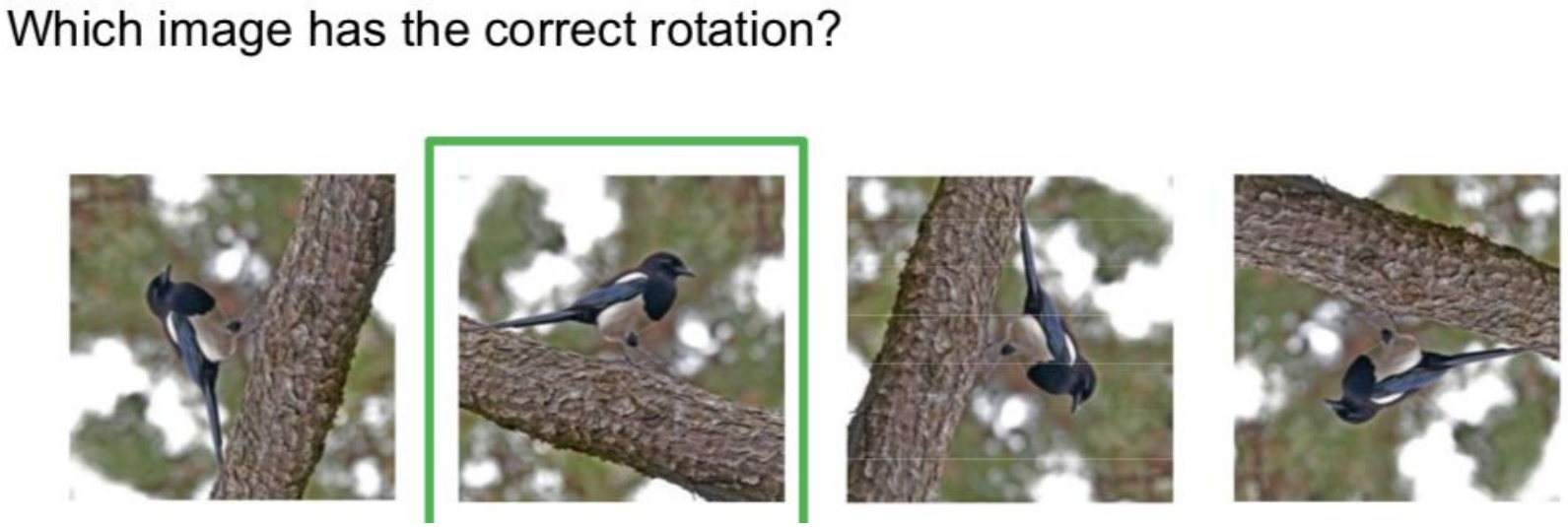
[9] C. Vondrick, H. Pirsiavash, and A. Torralba, “Generating videos with scene dynamics,” *NIPS 2016.*
[10] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Deep end2end voxel2voxel prediction,” *CVPRW 2016.*
[11] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, “Decomposing motion and content for natural video sequence prediction,”, *ICLR 2017*.

UniGe | MaLGa

Context-based pretext tasks

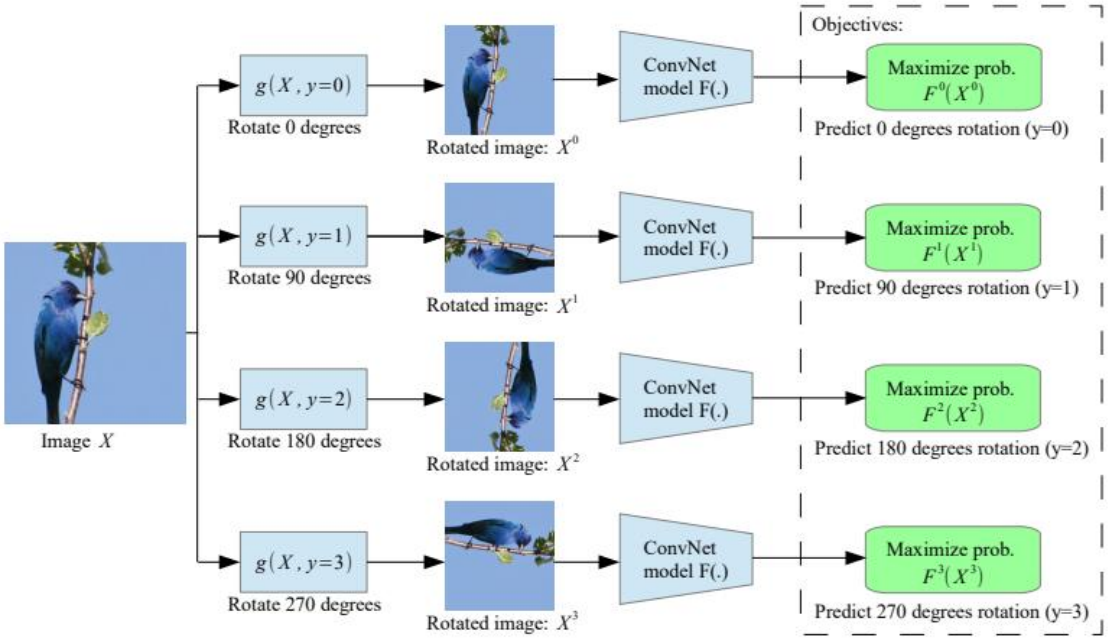
Geometric transformation: Image Rotations

Context-based pretext tasks



S. Gidaris, P. Singh, N. Komodakis, “Unsupervised representation learning by predicting image rotations,” *ICLR 2018*.

Geometric transformation
Context-based pretext tasks



S. Gidaris, P. Singh, N. Komodakis, "Unsupervised representation learning by predicting image rotations," *ICLR 2018*.

Image Rotations

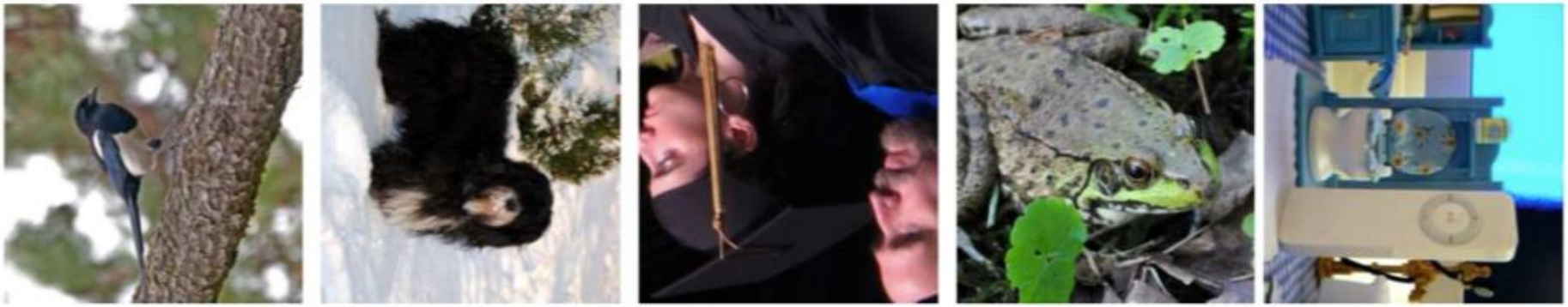
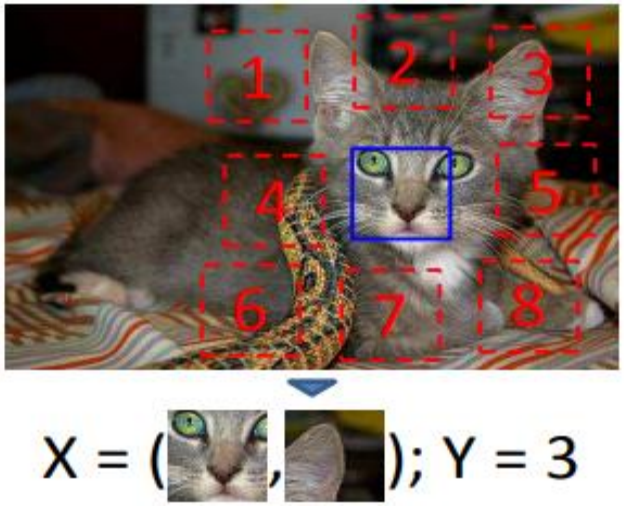
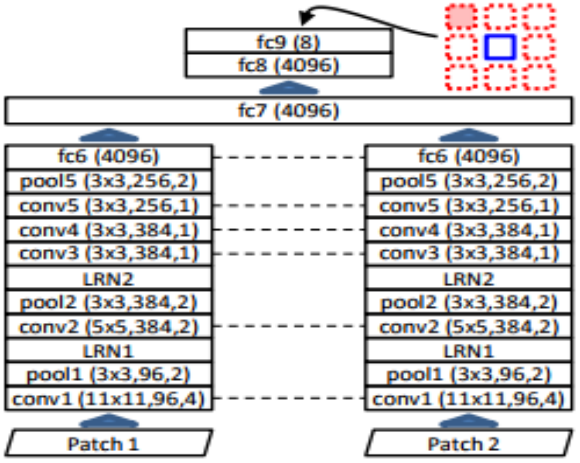


Figure 1: Images rotated by random multiples of 90 degrees (e.g., 0, 90, 180, or 270 degrees). The core intuition of our self-supervised feature learning approach is that if someone is not aware of the concepts of the objects depicted in the images, he cannot recognize the rotation that was applied to them.

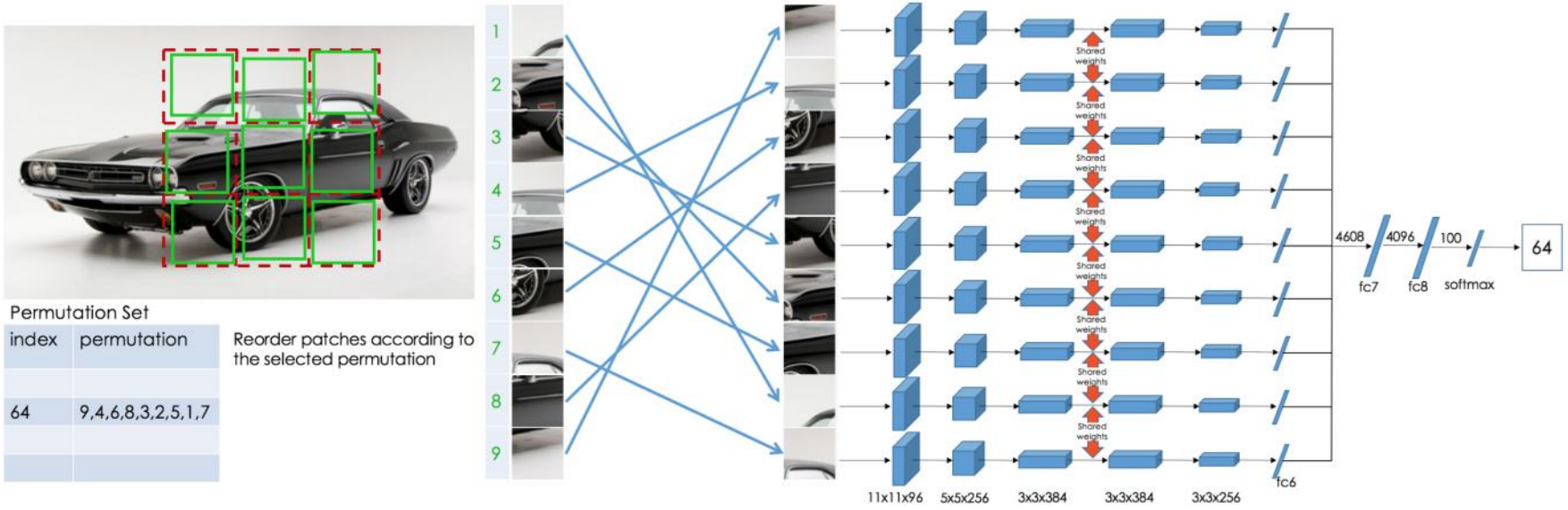
Relative Position task





C. Doersch, A. Gupta, A. Efros, “Unsupervised visual representation learning by context prediction, ” *ICCV 2015.*

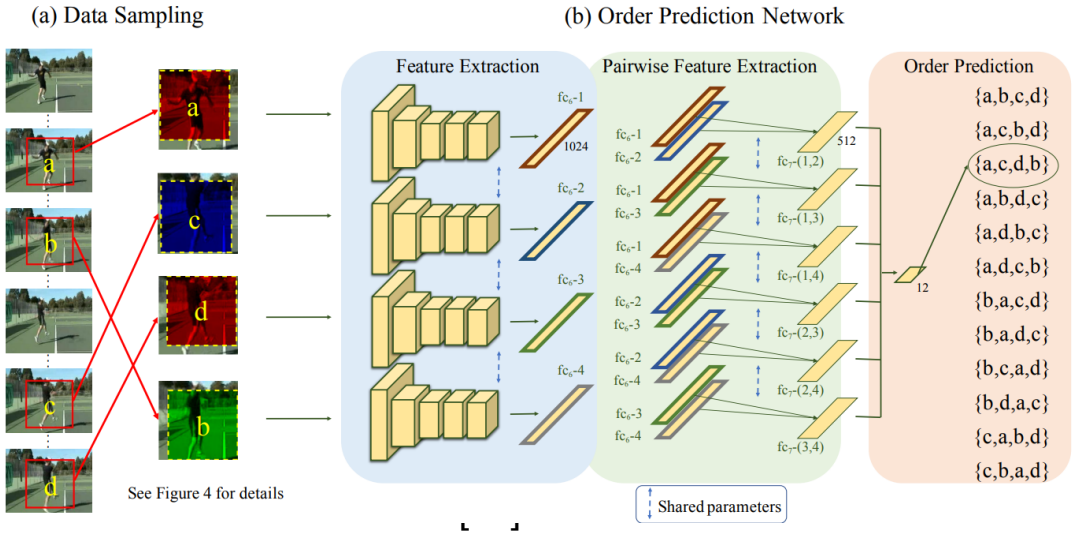
Solving a Jigsaw Puzzle
Context-based pretext tasks



M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," *ECCV 2016*.

Video
Context-based pretext tasks

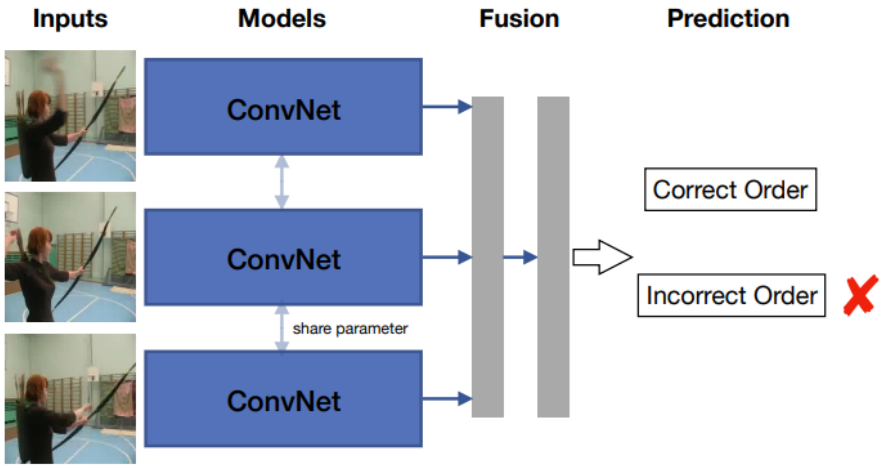
- Example 1: Detecting frames order in sequences
- Frame order recognition [16]



[16] H.-Y. Lee, J.-B. Huang, M. Singh, M.-H. Yang, "Unsupervised representation learning by sorting sequences", *ICCV 2017*

Video
Context-based pretext tasks

- Example 1: Detecting frames order in sequences
- Frame order verification [17]



[17]

[17] I. Misra, C. L. Zitnick, M. Hebert, "Shuffle and learn: unsupervised learning using temporal order verification", *ECCV 2016*

[illegible]M[illegible]

Task: given a color video …
Colorize all frames of a gray scale version using a reference frame

Reference Frame

Gray-scale Video

Vondrick, Shrivastava, Fathi, Guadarrama, Murphy. "Tracking Emerges by Colorizing Videos", *ECCV 2018*

Learning and Using the Arrow of Time

Figure 1: Seeing these ordered frames from videos, can you tell whether each video is playing forward or backward? (answer below[1]). Depending on the video, solving the task may require (a) low-level understanding (e.g. physics), (b) high-level reasoning (e.g. semantics), or (c) familiarity with very subtle effects or with (d) camera conventions. In this work, we learn and exploit several types of knowledge to predict the arrow of time automatically with neural network models trained on large-scale video datasets.

D. Wei, J. Lim, W. Freeman, A. Zisserman. "Learning and Using the Arrow of Time" *CVPR 2018*

Learning and Using the Arrow of Time

- First, can we train a reliable arrow of time classifier from large-scale natural videos while avoiding artificial cues (i.e. cues introduced during video production, not from the visual world);

- Second, what does the model learn about the visual world in order to solve this task;

- Third, and last, can we apply such learned commonsense knowledge to other video analysis tasks? (video representation learning and video forensics)

When the network is cheating

Black framing pattern

Cinematic conventions

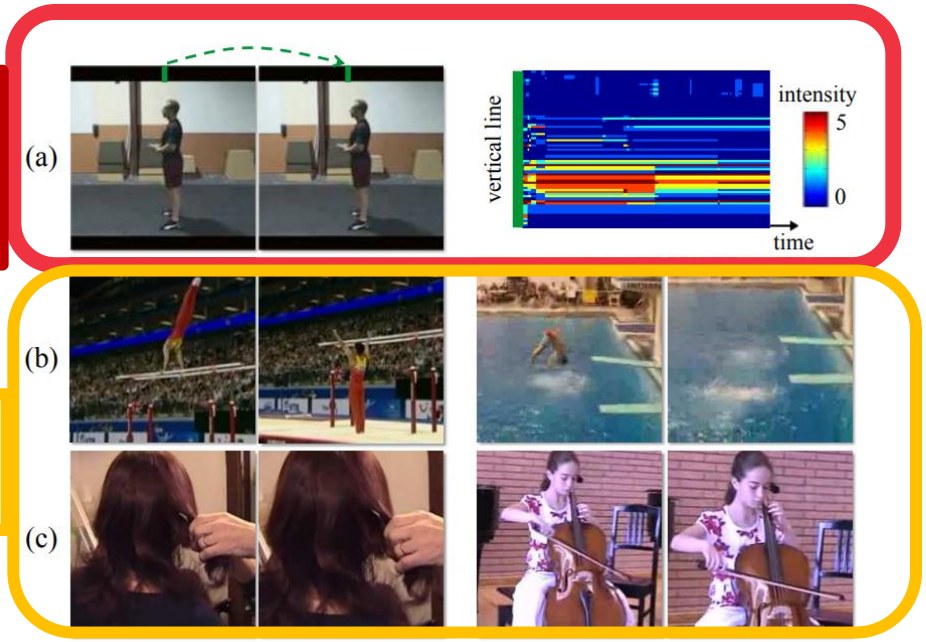


Figure 3: Illustration of artificial signals from videos in UCF101 dataset. (a) The black framing of the clip has non-zero intensity value (left), and a vertical slice over time displays an asymmetric temporal pattern (right). After training, we cluster the learned last-layer feature of top-confident test clips. We find some clusters have consistent (b) tilt-down or (c) zoom-in camera motion. We show two frames from two representative clips for each cluster.

Context-based pretext tasks (tasks from image transformations)

- Pretext tasks focus on “visual common sense”, e.g., predict rotations, inpainting, rearrangement, and colorization.
- The models are forced learn good features about natural images, e.g., semantic representation of an object category, in order to solve the pretext tasks.
- We don’t care about the performance of these pretext tasks, but rather how useful the learned features are for downstream tasks (classification, detection, segmentation).
- Problems: 1) coming up with individual pretext tasks is tedious, and 2) the learned representations may not be general.

UniGe | MaLGa

Self-supervised contrastive learning

Pretext tasks from image transformations

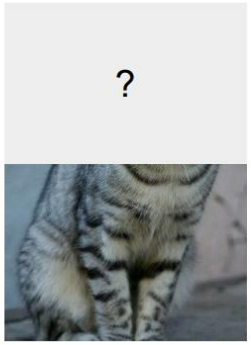
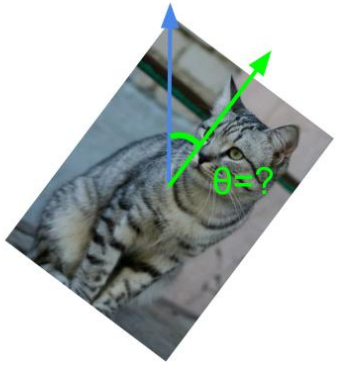


image completion



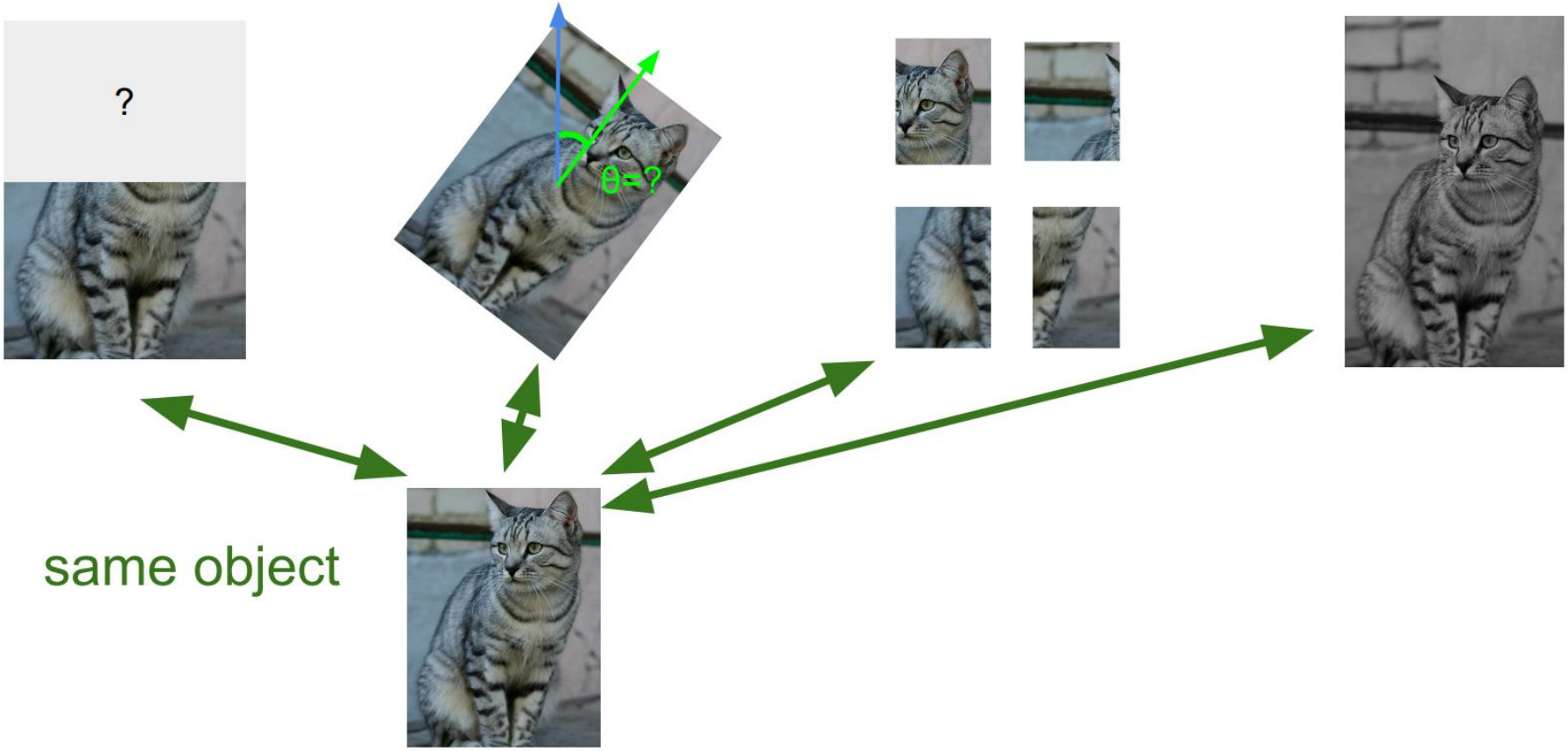
rotation prediction

“jigsaw puzzle”

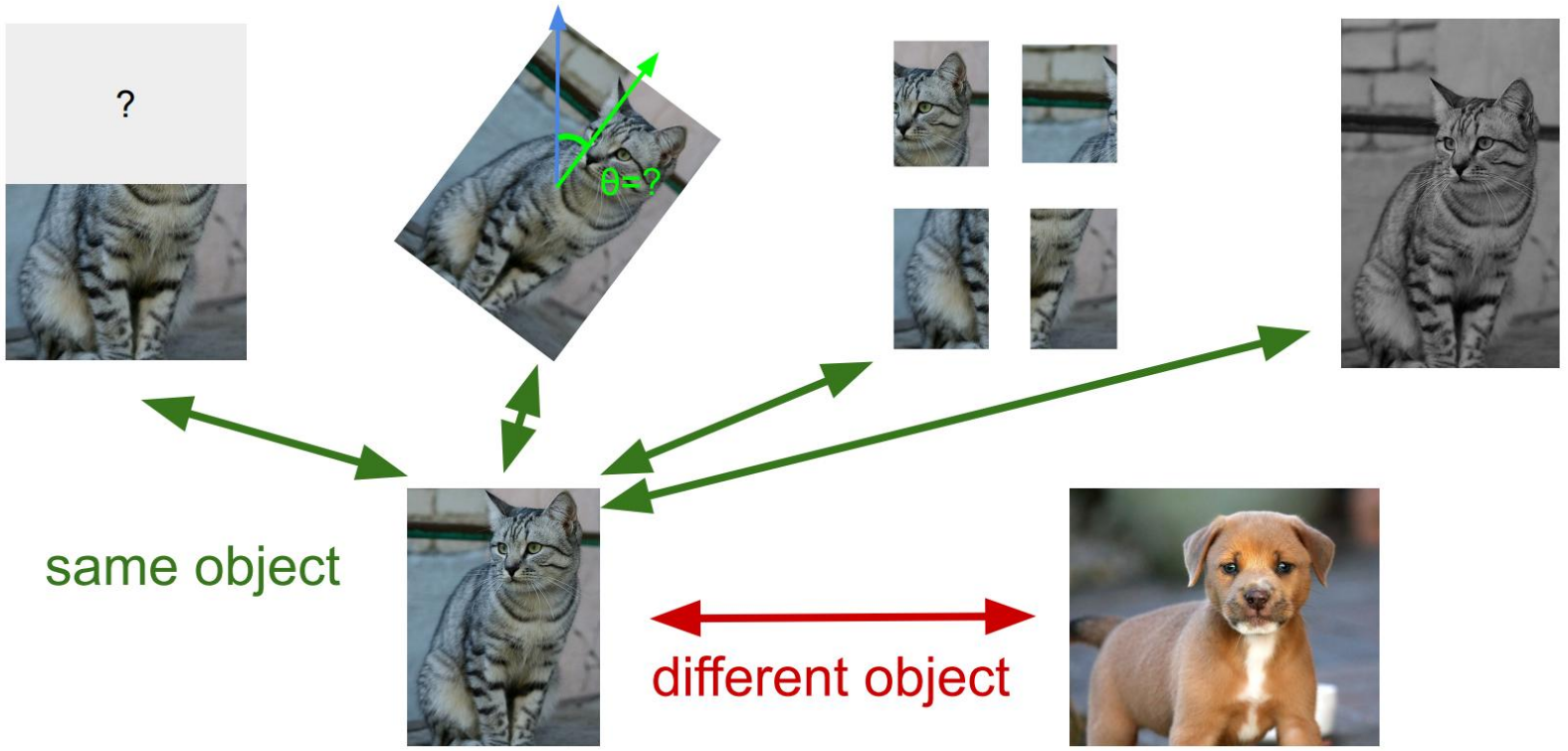
colorization

- Learned representations may be tied to a specific pretext task!
- Can we come up with a more general pretext task?

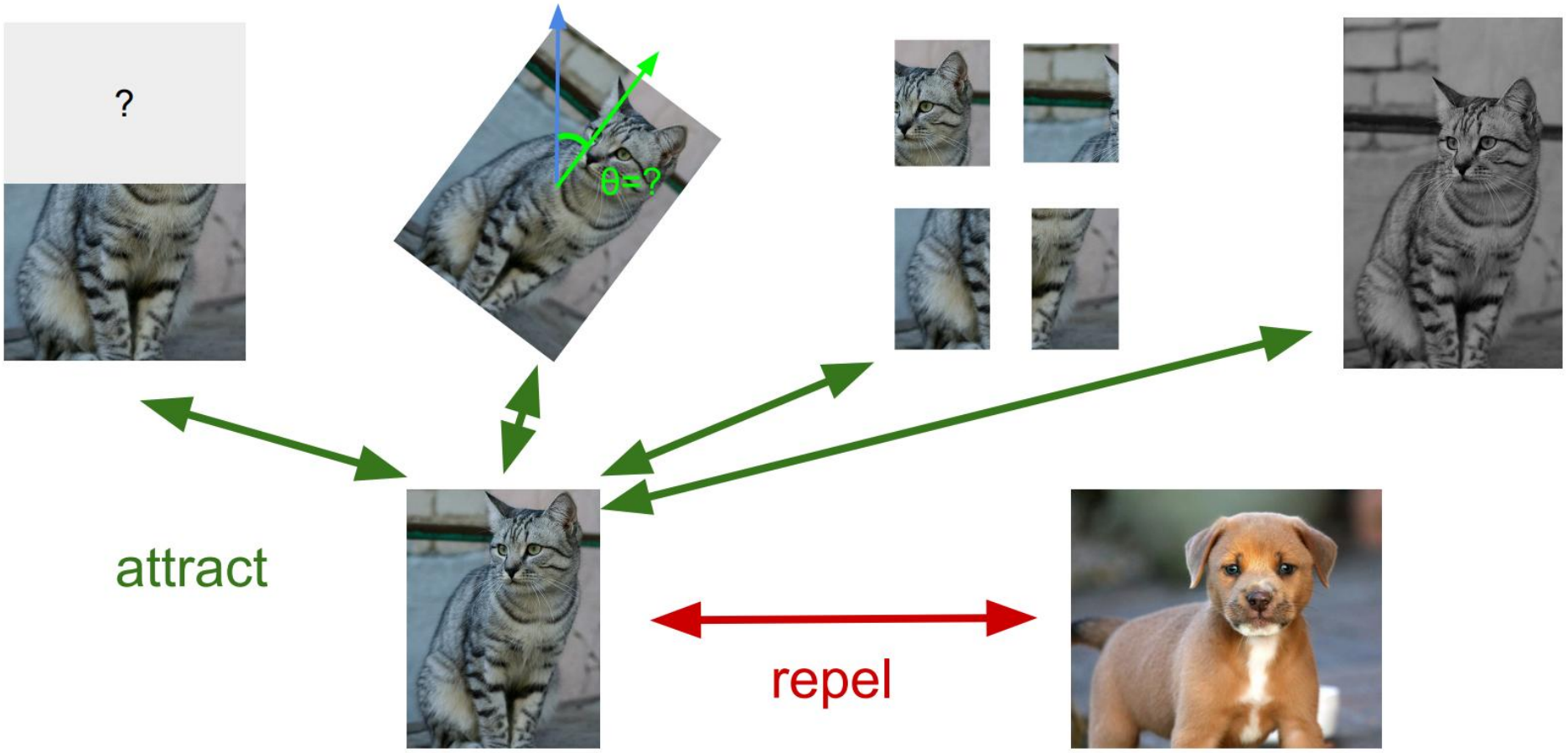
What is the potential problem with this task?



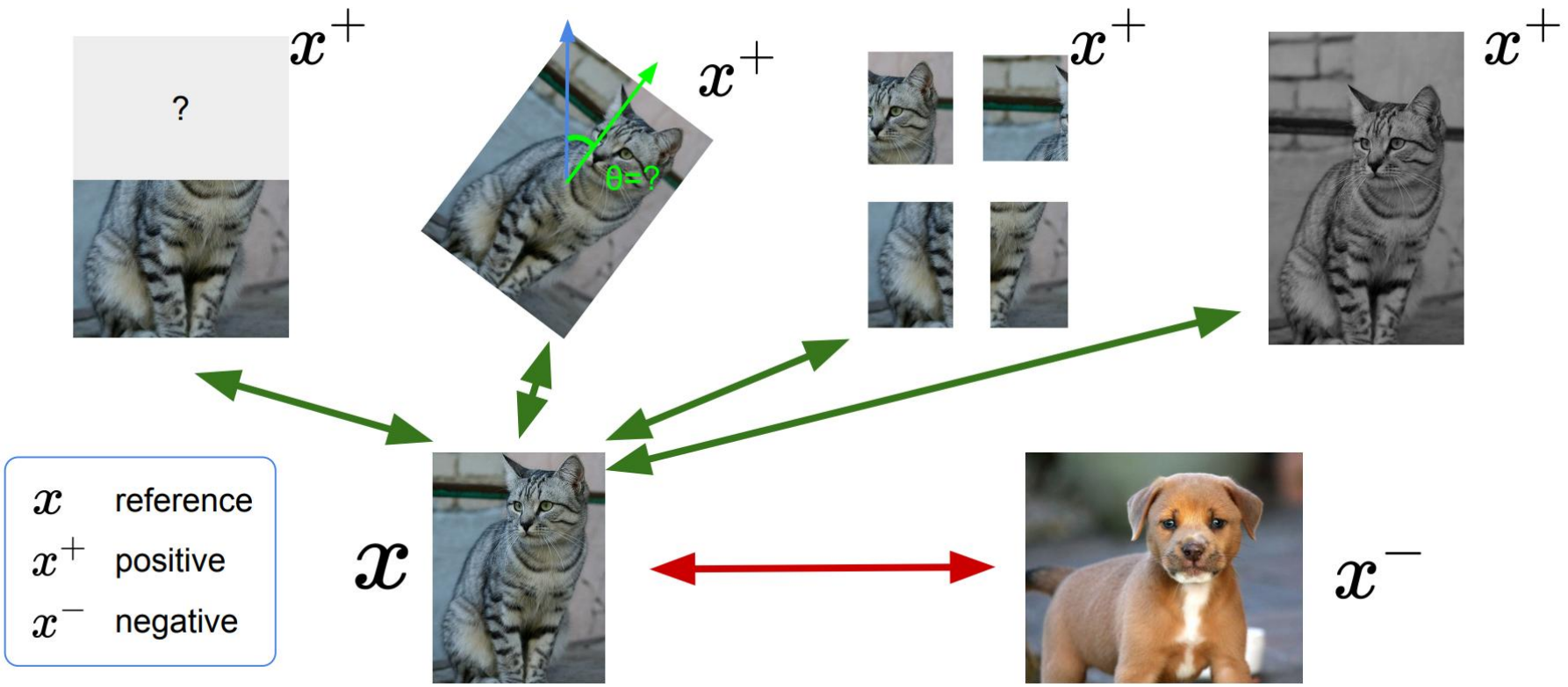
A more general pretext task



Self-supervised contrastive learning



Some notations



A formulation on contrastive learning (1)

$$\text{score}(f(x), f(x^+)) >> \text{score}(f(x), f(x^-))$$

Given a chosen score function, we aim to learn an encoder function f that yields high score for positive pairs (x, x+) and low scores for negative pairs (x, x-)

A formulation on contrastive learning (2)

- Given one positive and N-1 Negative samples

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+))}{\exp(s(f(x), f(x^+)) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))} \right]$$

A formulation on contrastive learning (3)

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+))}{\exp(s(f(x), f(x^+)) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))} \right]$$

x

x^+

x

x_1^-

x_2^-

x_3^-

...

A formulation on contrastive learning (4)

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+))}{\exp(s(f(x), f(x^+)) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))} \right]$$

score for the positive pair

score for the N-1 negative pairs

- Seems familiar?

A formulation on contrastive learning (5)

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+))}{\exp(s(f(x), f(x^+)) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))} \right]$$

score for the positive pair

score for the N-1 negative pairs

- Seems familiar? Yes!
- N-way cross-entropy -> learn to find the positive sample from the N samples

Softmax

$$p_i = \frac{e^{a_i}}{\sum_{k=1}^{N} e^a_k}$$

Cross-entropy loss

$$H(y, p) = -\sum_i y_i log(p_i)$$

A formulation on contrastive learning (6)

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+))}{\exp(s(f(x), f(x^+)) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))} \right]$$

- Commonly known as the InfoNCE loss (van den Oord et al., 2018)
- InfoNCE is actually a lower bound on the mutual information between f(x) and f(x+):

$$MI[f(x), f(x^+)] - \log(N) \geq -L$$

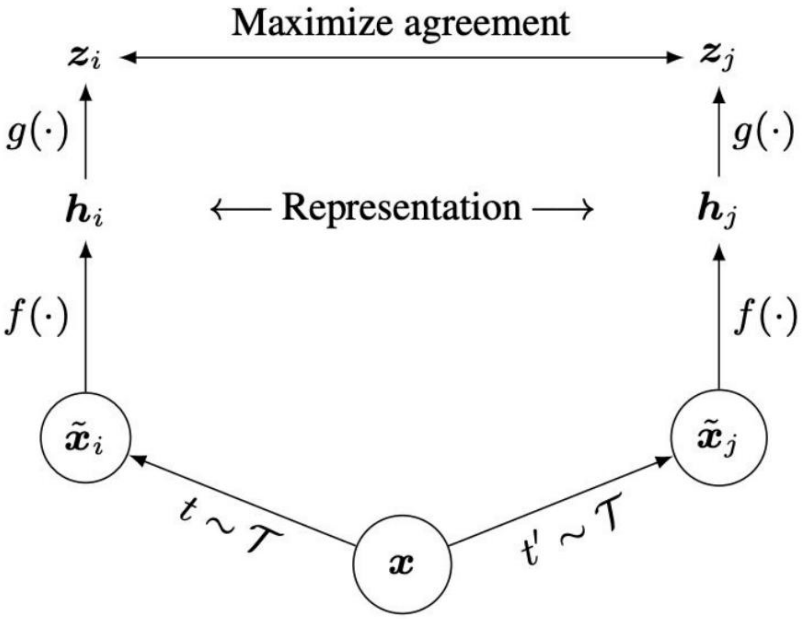
- **The larger the negative sample size** (N), the **tighter** the bound

SimCLR (1)

- Cosine similarity as the score function:

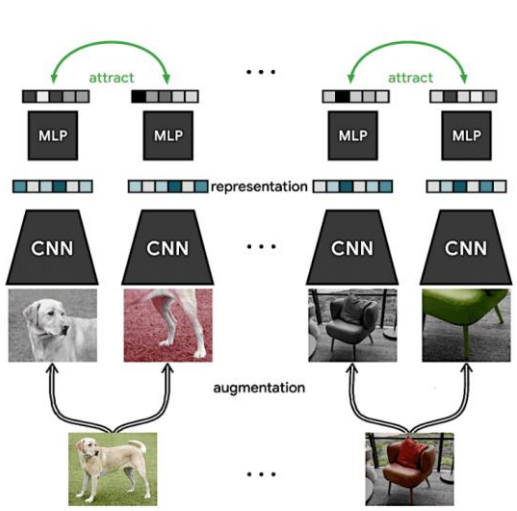
$$s(u, v) = \frac{u^T v}{||u||||v||}$$

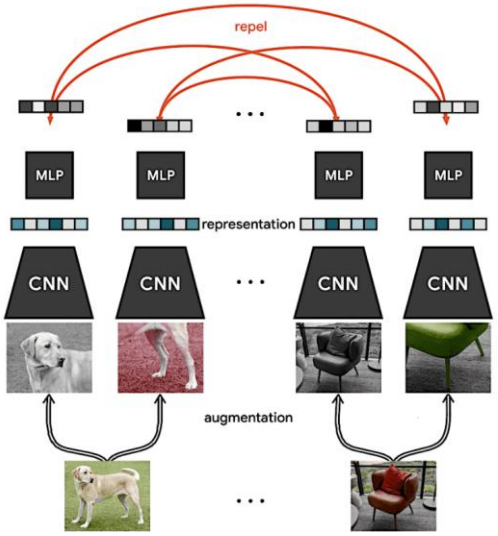
- Use a **projection network g(·)** to project features to a space where contrastive learning is applied
- Generate **positive samples through data augmentation**: random cropping, random color distortion, and random blur.



SimCLR (2)

SimCLR learns representations by **maximizing agreement between differently augmented views** of the same data example via a **contrastive loss** in the latent space.





A mini-batch of N examples is randomly sampled and the contrastive prediction task on pairs of augmented examples derived from the mini-batch is defined, resulting in 2N data points. Given a positive pair, the other 2(N-1) augmented examples within a mini-batch are treated as negative examples.
Then the loss function for a positive pair of examples (i,j) is defined as:

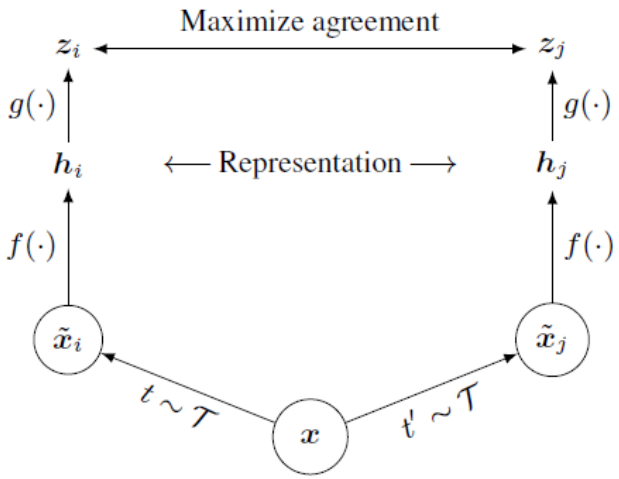
$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)}$$

The final loss is computed across all positive pairs, both (i,j) and (j,i), in a mini-batch. **The training requires a big batch size.**

https://github.com/google-research/simclr

Ting Chen et al. "A simple framework for contrastive learning of visual representations." *ICML* 2020.

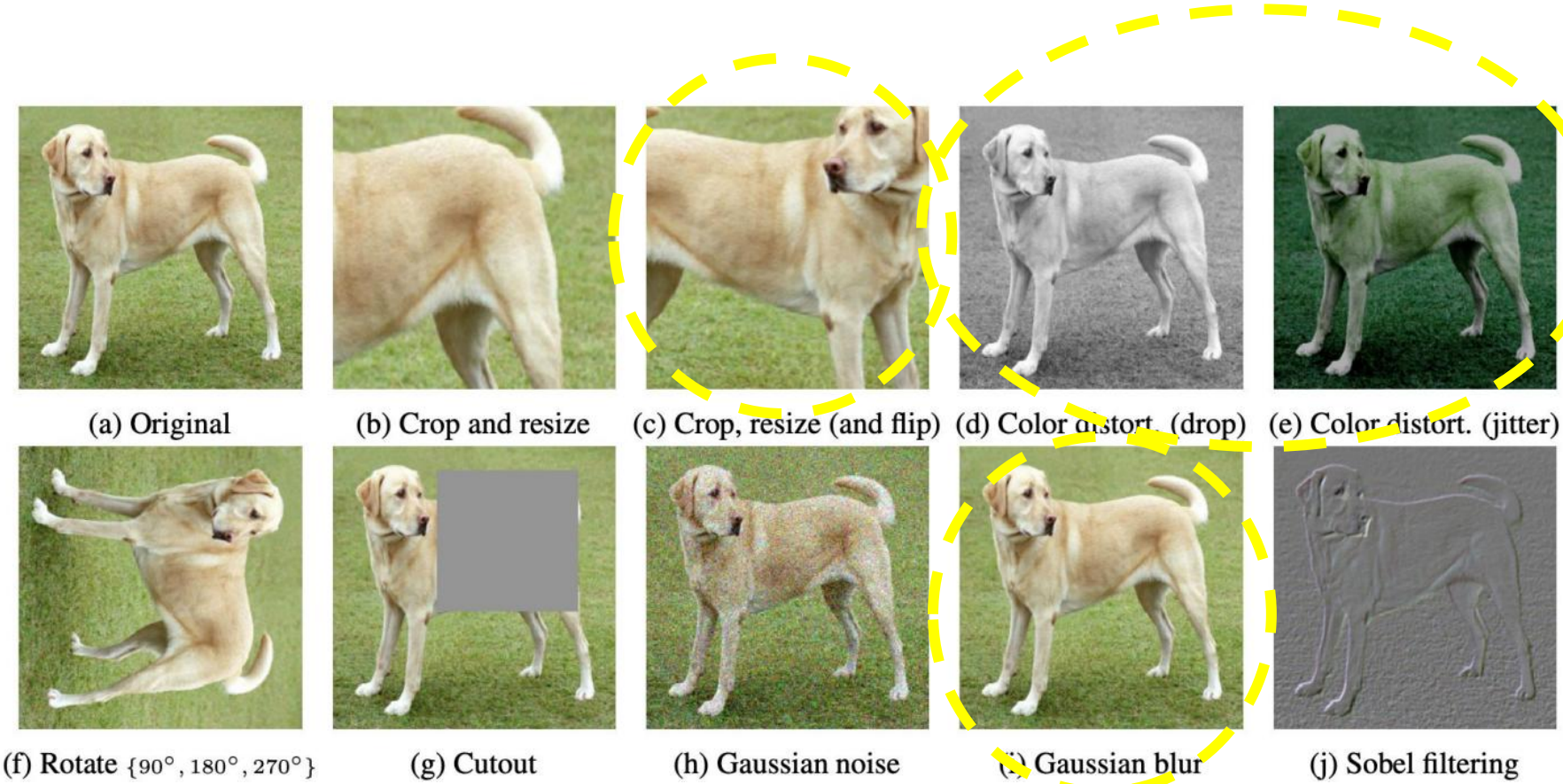
SimCLR (3)



Two separate data augmentation operators are **sampled from the same family of augmentations** (t ~ $\mathcal{T}$ and t' ~ $\mathcal{T}$) and applied to each data example to obtain **two correlated views**. A base encoder network *f*(.) and a projection head *g(.)* are trained to maximize agreement using a contrastive loss.
After training is completed, we throw away the projection head *g(.)* and use encoder *f(.)* and representation *h* for downstream tasks.
In fact the paper shows ***h* is a more discriminant feature** than *z* for downstream tasks.

Ting Chen et al. "A simple framework for contrastive learning of visual representations." *ICML* 2020.

SimCLR (4): Data augmentation



Ting Chen et al. "A simple framework for contrastive learning of visual representations." *ICML* 2020.

SimCLR (5)

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 $g(\cdot)$

Positive pair

InfoNCE loss
Use the other (N-1) augmented samples in the mini-batch as negatives

SimCLR (6)

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 $g(\cdot)$

Positive pair

InfoNCE loss
Use the other (N-1) augmented samples in the mini-batch as negatives

SimCLR (7)

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 $g(\cdot)$

Positive pair

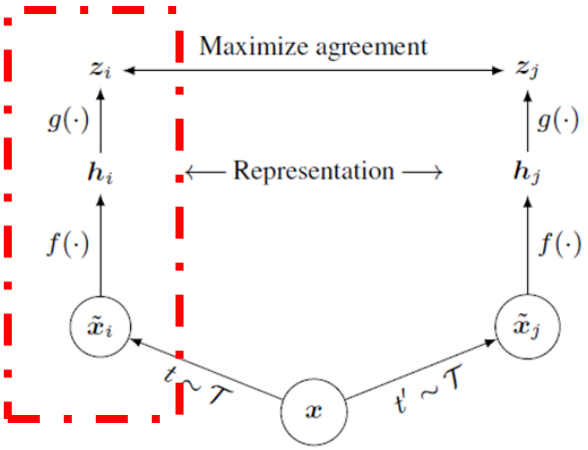
InfoNCE loss
Use the other (N-1) augmented samples in the mini-batch as negatives

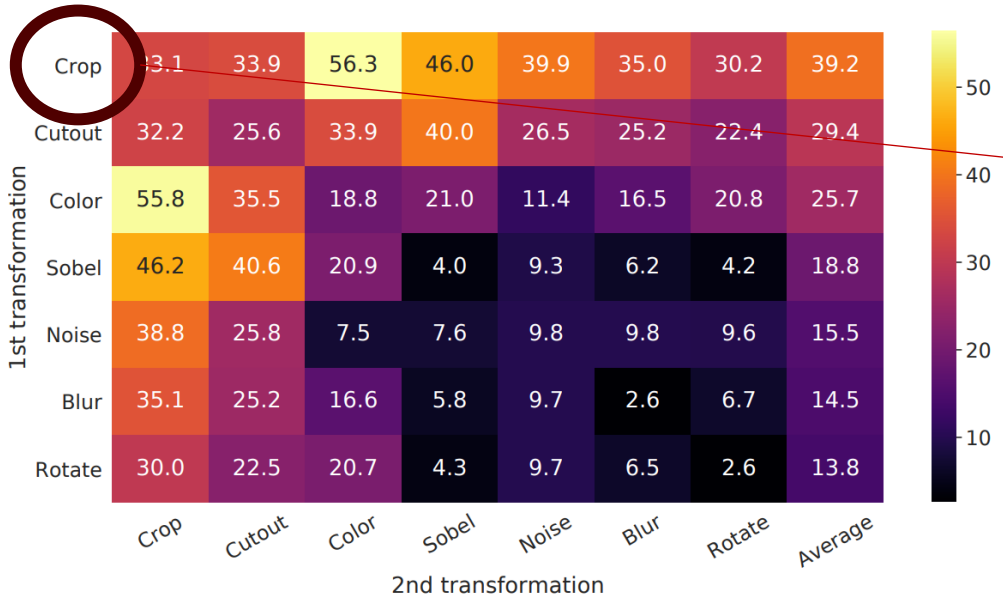
So you have two positive pairs, and you can each one of them treated as the reference

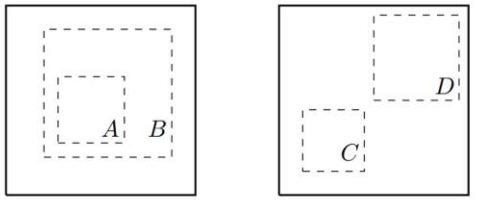
For each pair, the loss is computed both considering (i,j) and (j,i)

SimCLR (8)

- Applied single or combined transform to one branch;
- Identity on the other;
- no single transformation suffices to learn good representations;
- When composing augmentations, the contrastive prediction task becomes harder, but the quality of representation improves dramatically



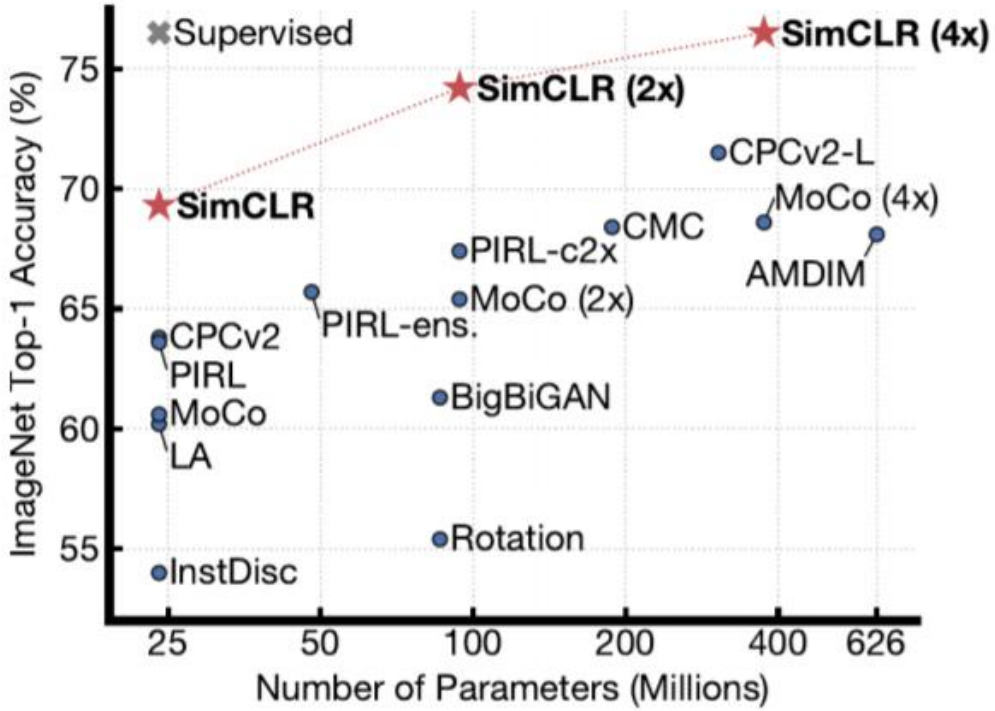




(a) Global and local views. (b) Adjacent views.

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.

SimCLR (9)



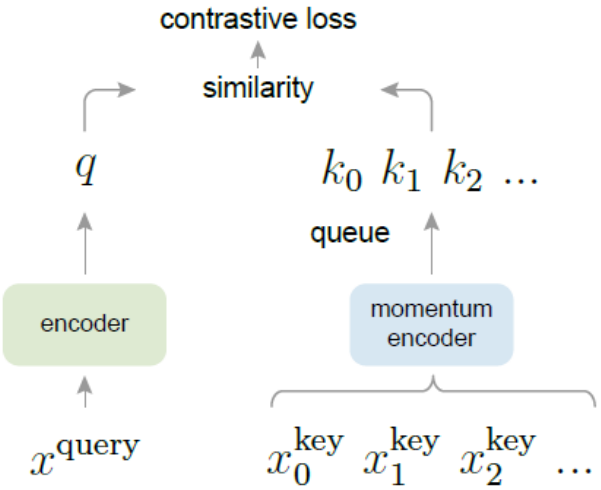
ImageNet Top-1 accuracy of linear classifiers trained on representations learned with different self-supervised methods (pretrained on ImageNet). Gray cross indicates supervised ResNet-50. SimCLR, is shown in bold. ResNet-50 architecture is used in 3 different hidden layer widths (width multipliers of 1×, 2×, and 4×)

MoCo (Introduction and intuition)

- Contrastive learning can be framed as training an encoder for a dictionary look-up task;
- Encoded query *q* and a set of encoded samples {*k0, k1, k2, ...*};
- A single key (denoted as *k+*) in the dictionary that *q* matches;
- Contrastive loss is low when the query is similar to k+ and dissimilar to other keys

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

Self-Supervised Learning: MoCo – Momentum Contrast



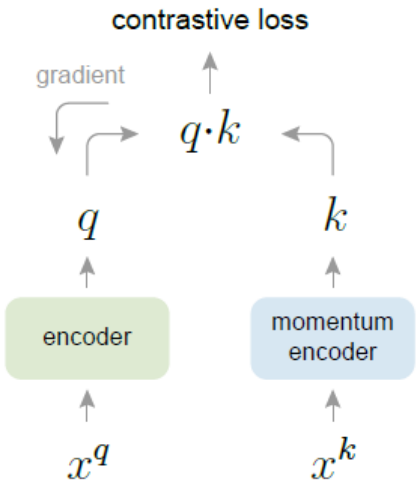
$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i / \tau)}$$

Momentum Contrast (MoCo) trains a visual representation encoder by **matching an encoded query q** to a **dictionary of encoded keys using a contrastive loss**. The dictionary keys are defined on-the-fly by a set of data samples.

The **dictionary is built as a queue**, with the current mini-batch enqueued and the oldest mini-batch dequeued, decoupling it from the mini-batch size. The keys are encoded by a slowly progressing encoder, driven by a momentum update with the query encoder.

Kaiming He et al. "Momentum contrast for unsupervised visual representation learning." *CVPR* 2020.

MoCo



Using a **queue can make the dictionary large**, but it also makes it **intractable to update the key encoder by back-propagation** (the gradient should propagate to all samples in the queue). Formally, denoting the parameters of the key encoder f_k as θ_k and those of the query encoder f_q as θ_q, f_k is updated by:

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

$$m \in [0,1) is\ the\ momentum\ coefficient$$

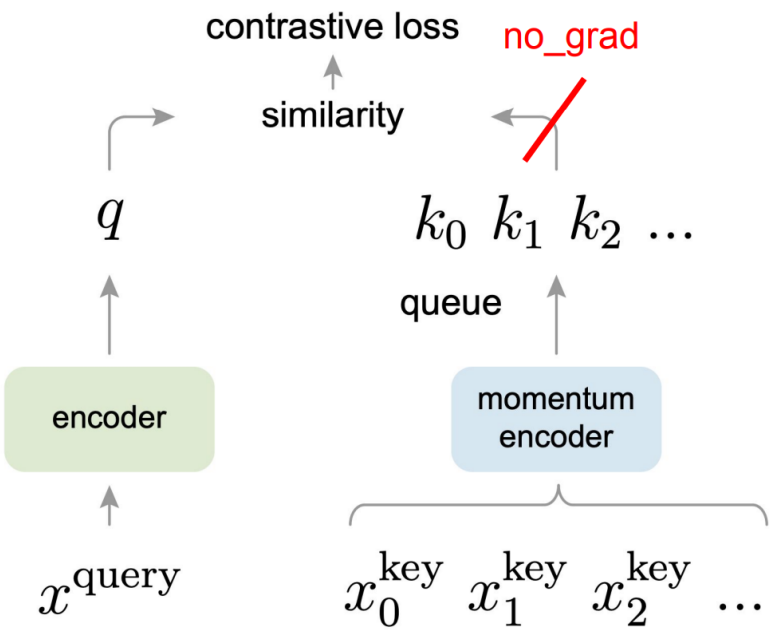
Only the parameters f_q are updated by back-propagation.
The momentum update in eqn. above makes f_k evolve more smoothly than f_q.

As a result, though the keys in the queue are encoded by different encoders (in different mini-batches), the difference among these encoders can be made small.

The main hypothesis is that **good features can be learned by a large dictionary** that covers a **rich set of negative samples**, while the encoder for the dictionary keys is kept as consistent as possible despite its evolution.

Kaiming He et al. "Momentum contrast for unsupervised visual representation learning." *CVPR* 2020.

Self-Supervised Learning: MoCo – Momentum Contrast



$$q = f_q(x^q)$$

Key differences to SimCLR:

- Keep a running queue of keys (negative samples).
- Compute gradients and update the encoder only through queries.
- Decouple min-batch size with the number of keys: can support a large number of negative samples.

Kaiming He et al. "Momentum contrast for unsupervised visual representation learning." *CVPR* 2020.

MoCo pseudocode

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.

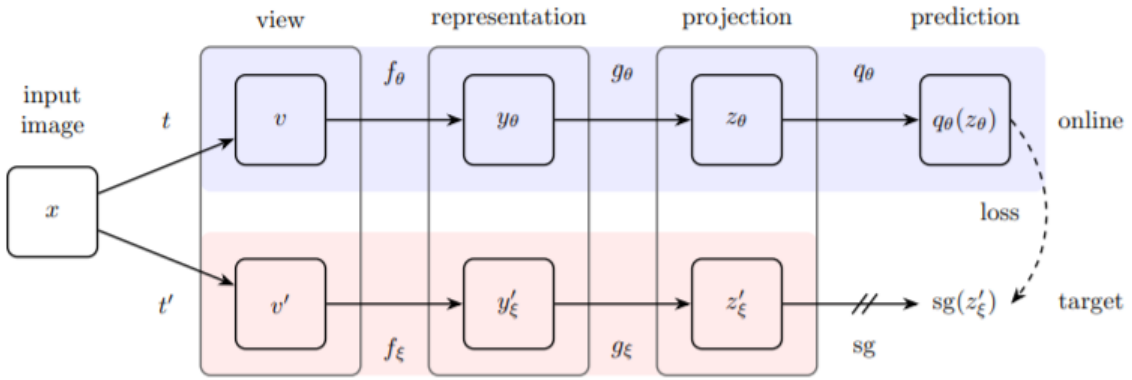
No gradients for the momentum encoder

Queue of negative samples

Momentum update

Update the FIFO negative sample queue

Byol



BYOL is a more recent published method that revisits advancements proposed in previous methods (MoCo and SimCLR) for positive learning. Similarly to SimCLR it makes use of **strong data augmentation** transformations on the image *x* and the **feature representation used at inference time is *f*** (all the other blocks are discarded). Whereas, similarly to MoCo it makes use of a ***online network t* and a *target network t'***.

Moreover, BYOL is **not a contrastive approach**, it **maximizes the similarity** metric between the target and online predictions by means of mean squared error.

Grill et al. "Bootstrap Your Own Latent A New Approach to Self-Supervised Learning." *NeurIPS* 2020.

Byol

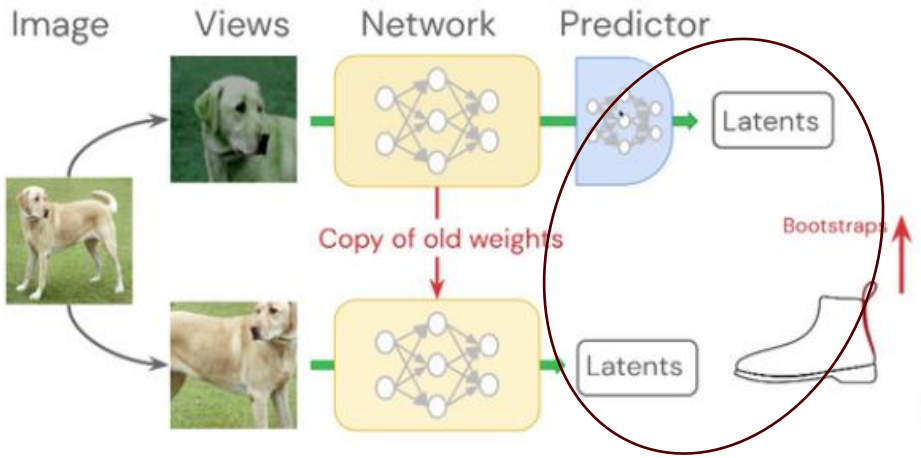
Process overview: We take one unlabelled image and generate two views, which should share the same label. Then Byol uses the top-view image, pass it into a standard network-predictor. Instead of adopting ground truth labels we pass the second image through another network that we called label generator network which is a delayed replica of the upper one except removing the predictor module. Then we use the latent output of the bottom part as the labels for the upper part.
We train then the upper part in supervised way.

Key ingredients:

- Image transformations
- Target Network
- Additional predictor on online network (if we do not
  use it, the representation collapses)

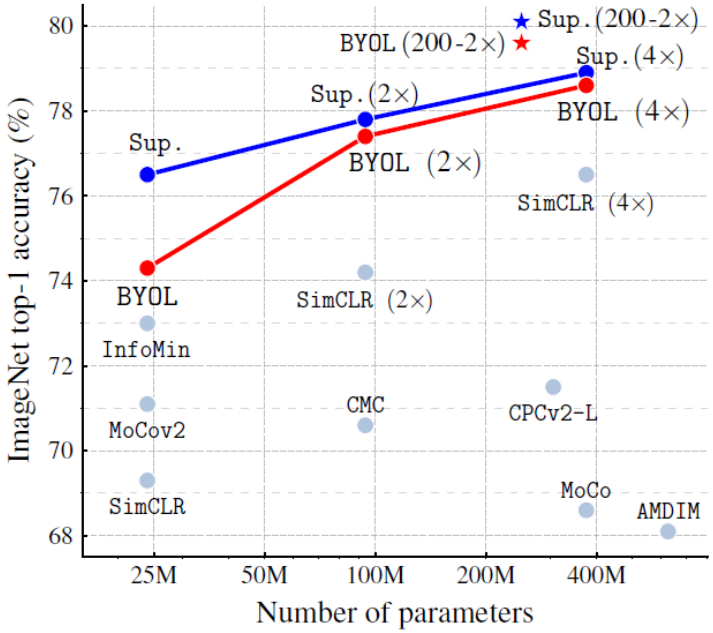
Takeaways:

- Does not use **negative examples**
- **More resilient to transformation and batch size**
  than contrastive methods



Grill et al. "Bootstrap Your Own Latent A New Approach to Self-Supervised Learning." *NeurIPS* 2020.
https://neurips.cc/virtual/2020/public/poster_f3ada80d5c4ee70142b17b8192b2958e.html

Byol results (2)

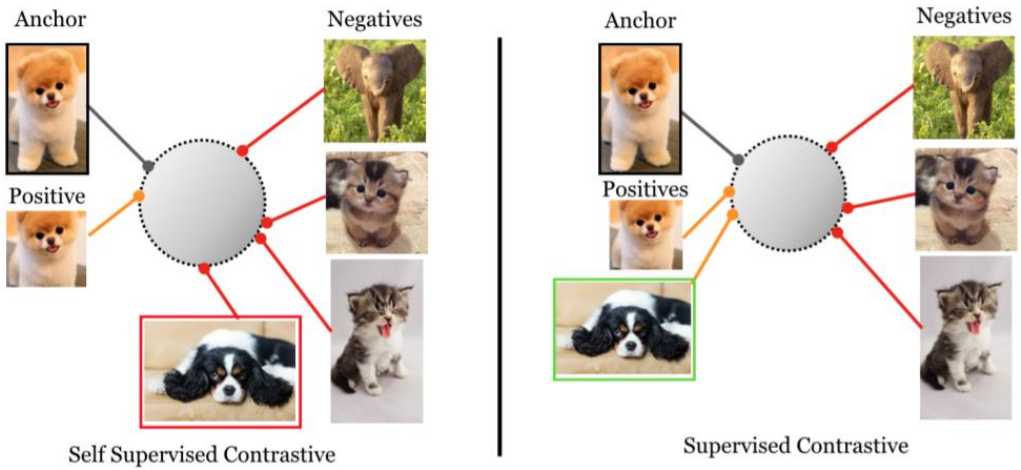


Performance of BYOL on ImageNet (linear evaluation) using ResNet-50 and the best architecture ResNet-200 (4x), compared to other unsupervised and supervised (Sup.) baselines.

BYOL achieves higher performance than state-of-the-art contrastive methods **without using negative pairs**. It iteratively bootstraps the outputs of a network to serve as targets for an enhanced representation.

Grill et al. "Bootstrap Your Own Latent A New Approach to Self-Supervised Learning." *NeurIPS* 2020.

Supervised Contrastive Learning

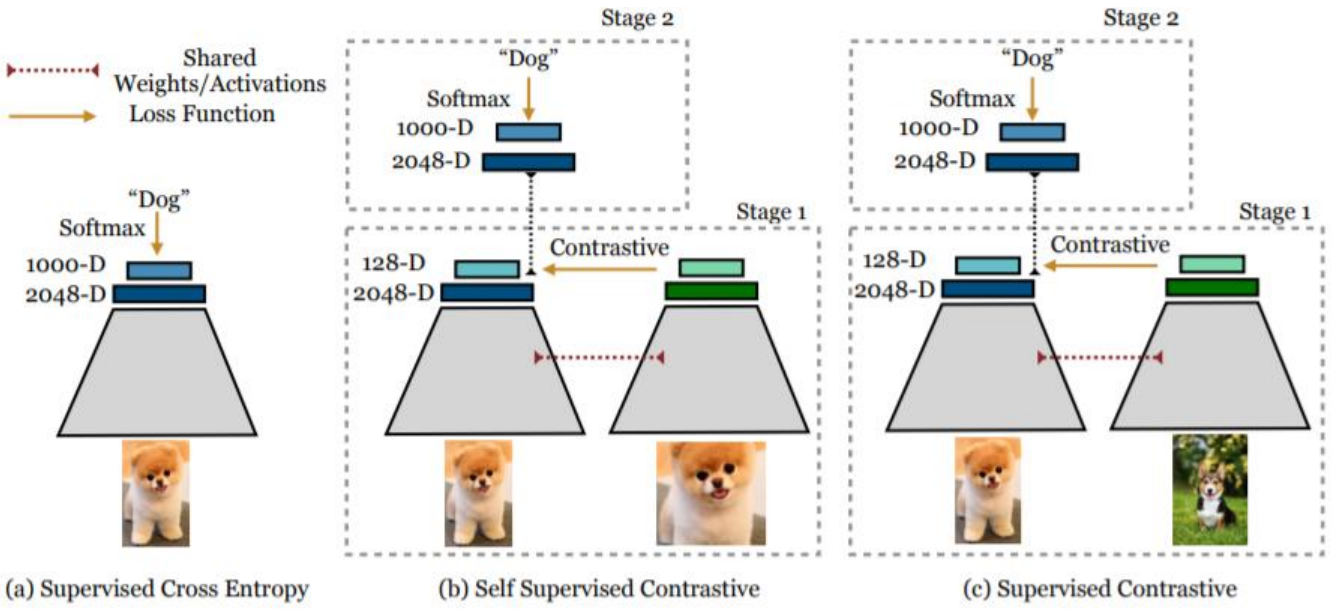


Supervised vs. self-supervised contrastive losses:
The *self-supervised contrastive loss* contrasts a single positive for each anchor (i.e., an augmented version of the same image) against a set of negatives consisting of the entire remainder of the batch.
The *supervised contrastive loss* (right), however, contrasts the set of all samples from the same class as positives against the negatives from the remainder of the batch. As demonstrated by the photo of the black and white puppy, taking class label information into account results in an embedding space where elements of the same class are more closely aligned than in the self-supervised case

Khosla et al. "SupCon - Supervised Contrastive Learning." *NeurIPS* 2020.

Supervised Contrastive Learning

SupCon aims at merging the best of two worlds, supervised learning (a) and recent contrastive based self-supervised learning (b). In fact they propose a supervised contrastive loss (c).



Khosla et al. "SupCon - Supervised Contrastive Learning." *NeurIPS* 2020.