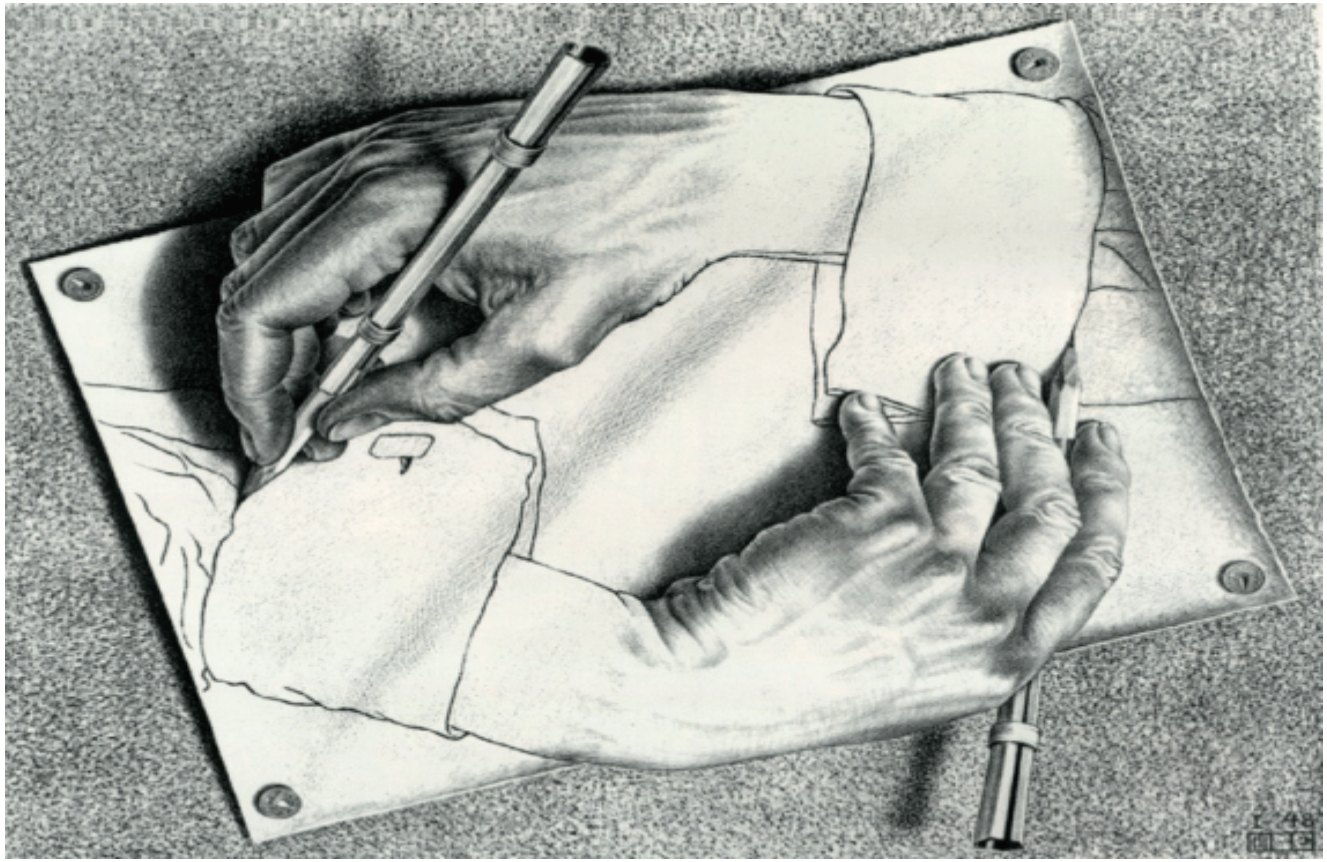
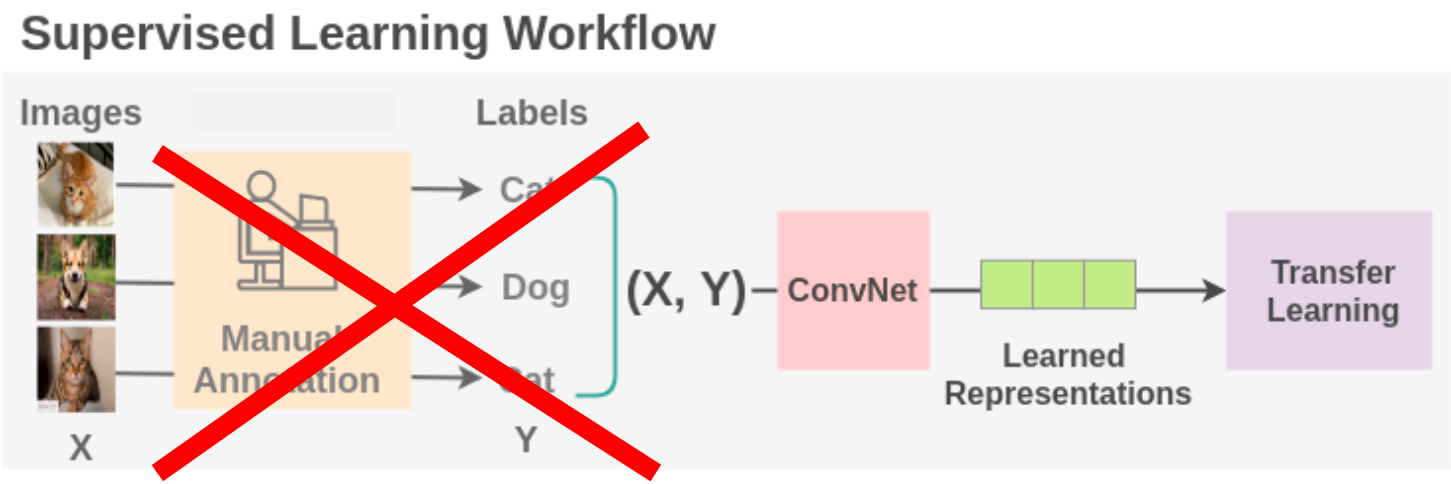
# Self-supervised learning



M.C. Escher, *Drawing Hands* (1948) – via A. Efros

# Motivation

- Recall the idea of *transfer learning*: start with general-purpose feature representation pre-trained on a large, diverse dataset and adapt it to specialized tasks
- Challenge: overcoming reliance on *supervised* pre-training



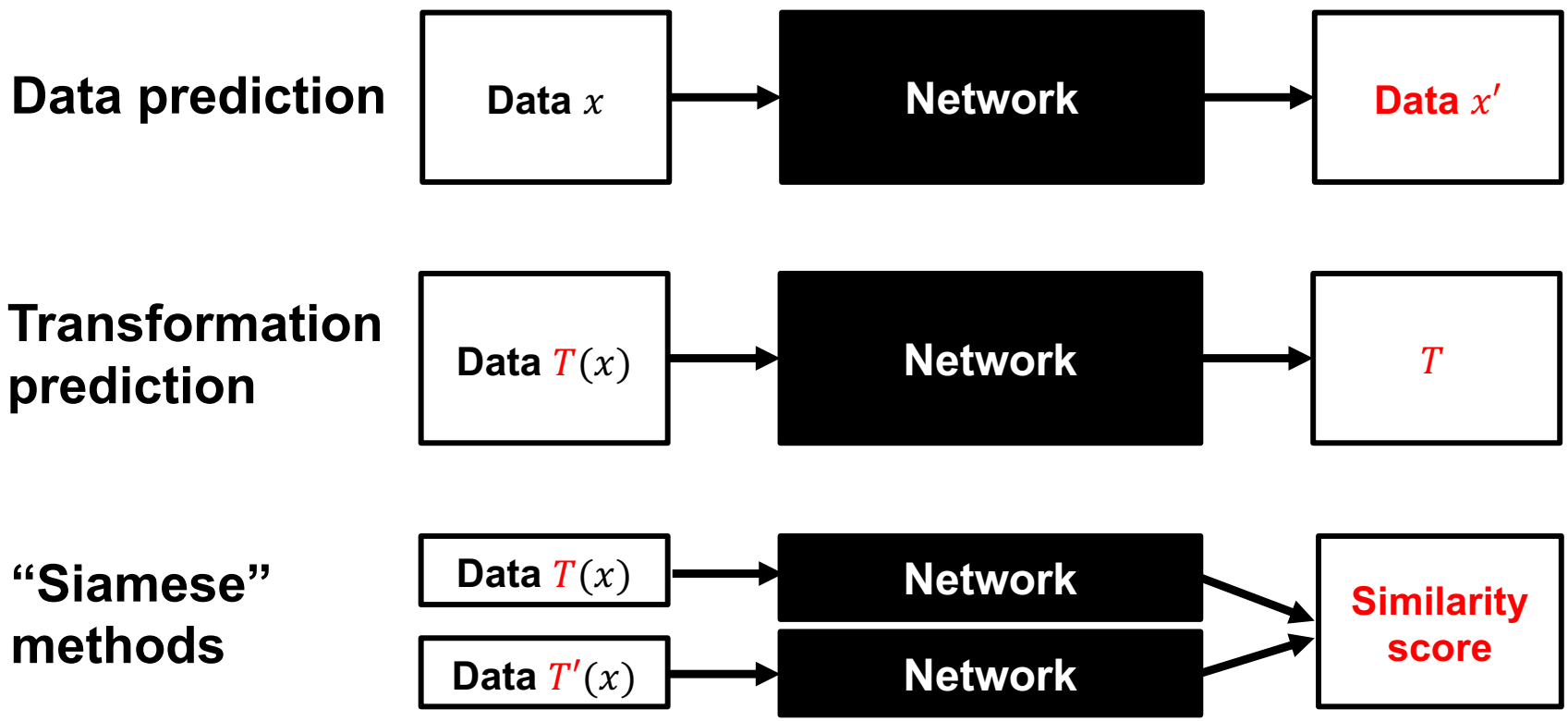

**Pretext task**

Figure source

## *Unsupervised* vs. *self-supervised* learning

- **Unsupervised learning**: any kind of learning without labels
  - Clustering and quantization
  - Dimensionality reduction, manifold learning
  - Density estimation
  - Learning to sample
- **Self-supervised learning**: a type of unsupervised learning in which the learner “makes up” labels from the data and then solves a supervised task

# Self-supervised learning: Outline




Source: A. Efros

# Data prediction: Inpainting

D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A. Efros. Context Encoders: Feature Learning by Inpainting. CVPR 2016

# Preview: Masked autoencoders

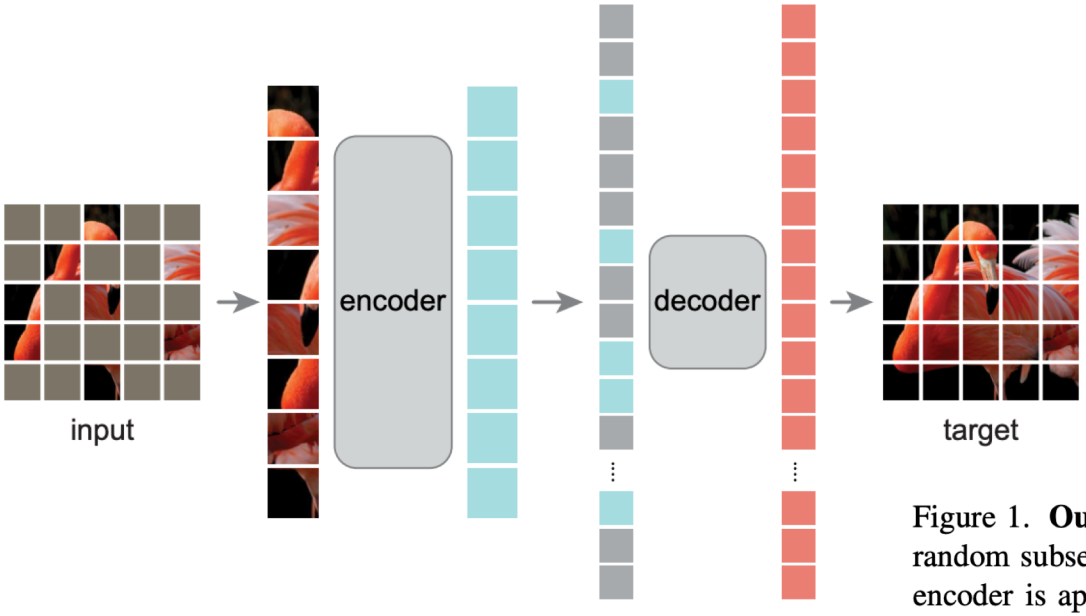



Figure 1. **Our MAE architecture**. During pre-training, a large random subset of image patches (*e.g.*, 75%) is masked out. The encoder is applied to the small subset of *visible patches*. Mask tokens are introduced *after* the encoder, and the full set of encoded patches and mask tokens is processed by a small decoder that reconstructs the original image in pixels. After pre-training, the decoder is discarded and the encoder is applied to uncorrupted images to produce representations for recognition tasks.

K. He et al. Masked autoencoders are scalable vision learners. CVPR 2022

# Preview: Masked autoencoders

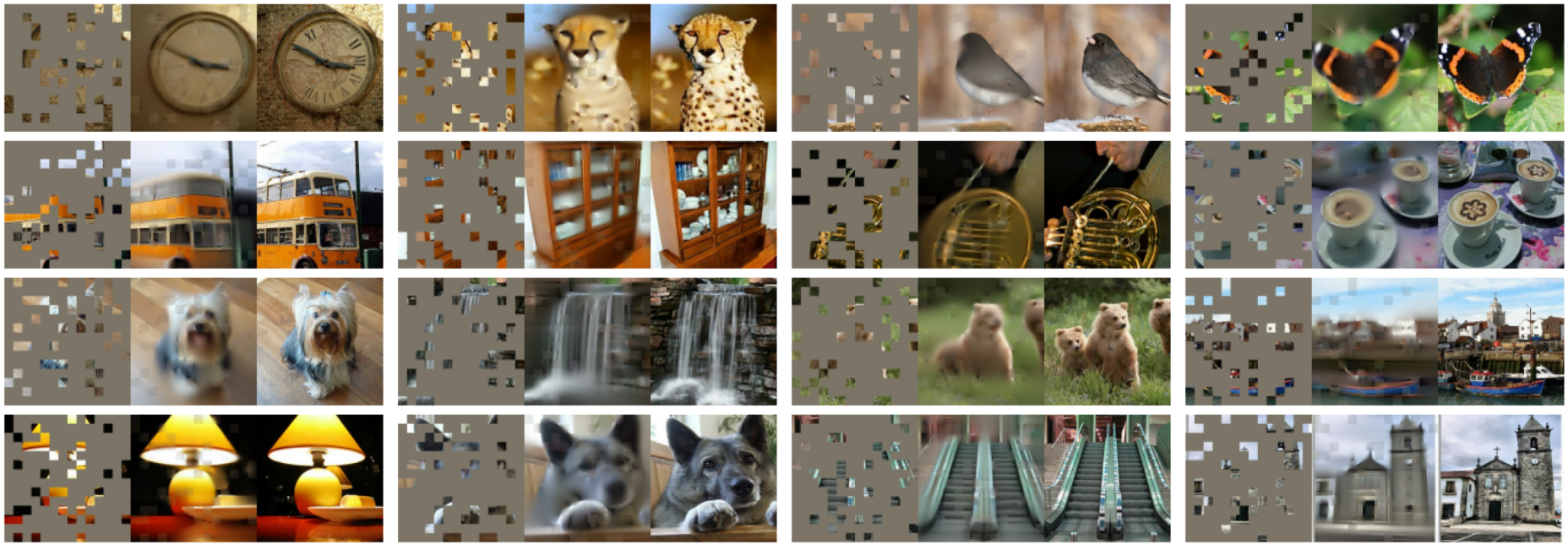
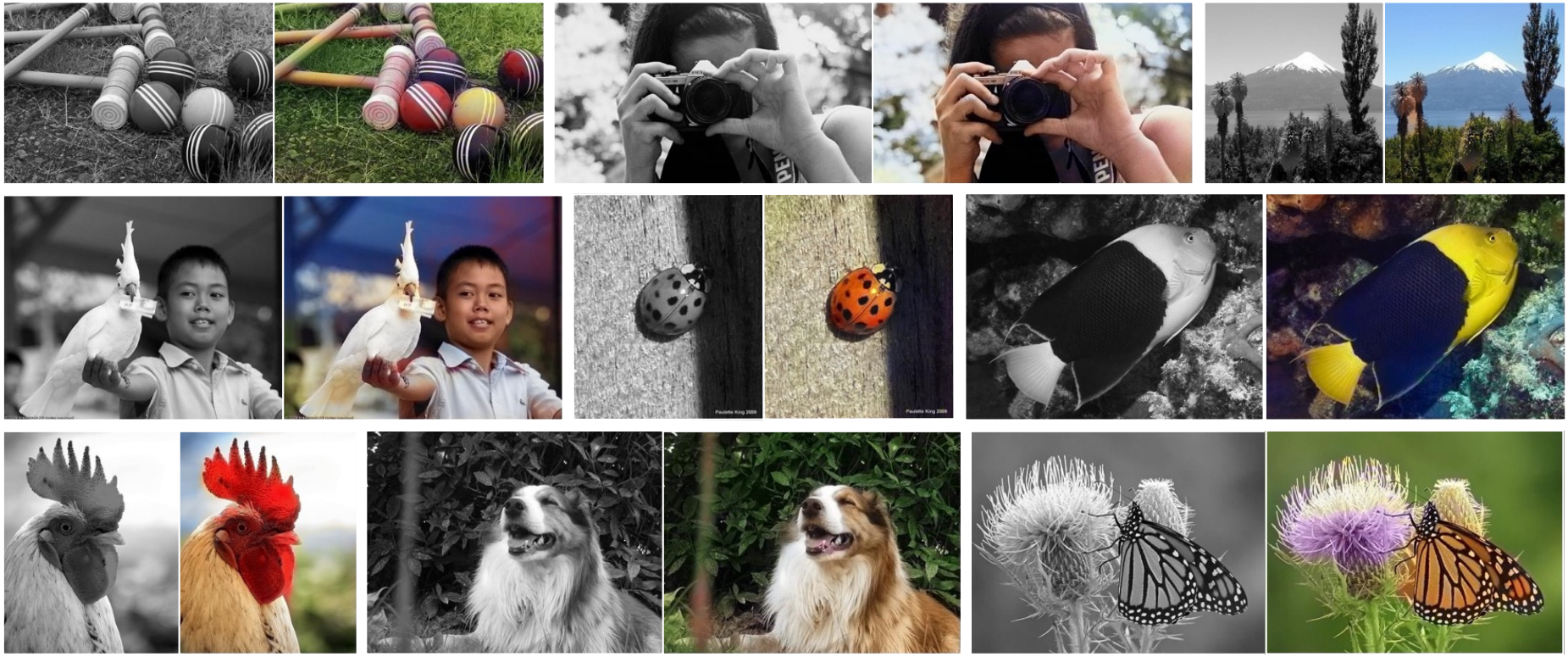


Figure 2. Example results on ImageNet *validation* images. For each triplet, we show the masked image (left), our MAE reconstruction[†] (middle), and the ground-truth (right). The masking ratio is 80%, leaving only 39 out of 196 patches. More examples are in the appendix.
[†]*As no loss is computed on visible patches, the model output on visible patches is qualitatively worse. One can simply overlay the output with the visible patches to improve visual quality. We intentionally opt not to do this, so we can more comprehensively demonstrate the method's behavior.*

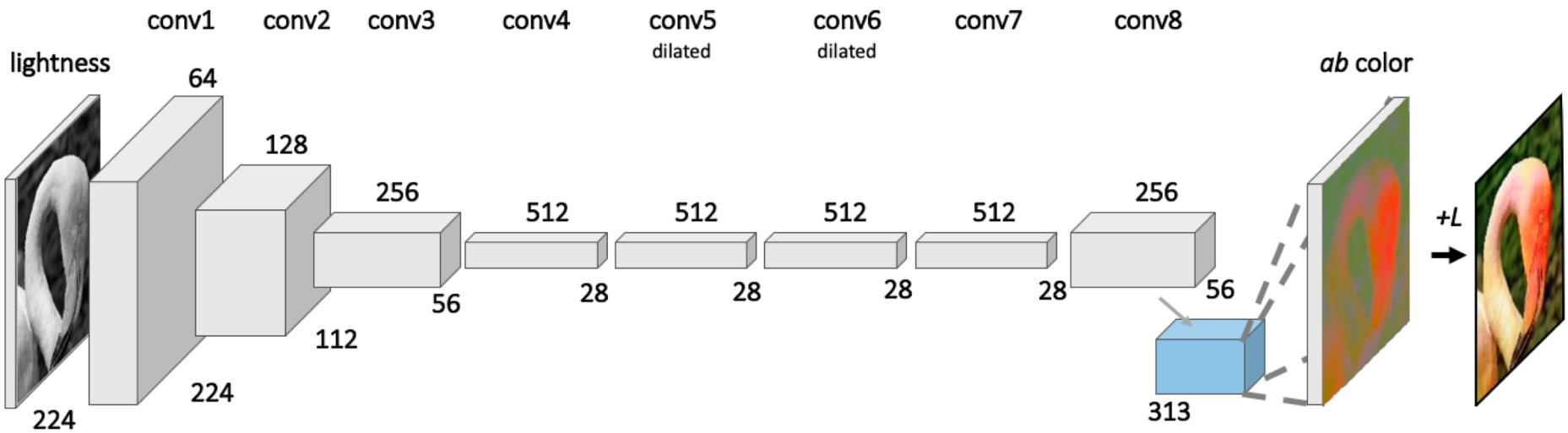
K. He et al. Masked autoencoders are scalable vision learners. CVPR 2022

# Colorization



R. Zhang, P. Isola, and A. Efros, Colorful Image Colorization, ECCV 2016

# Colorization: Architecture




R. Zhang, P. Isola, and A. Efros, [Colorful Image Colorization](), ECCV 2016

# Colorization: Results

R. Zhang, P. Isola, and A. Efros, Colorful Image Colorization, ECCV 2016

# Failure Cases



Source: A. Efros, R. Zhang

# Inherent Ambiguity

Grayscale

Source: A. Efros, R. Zhang

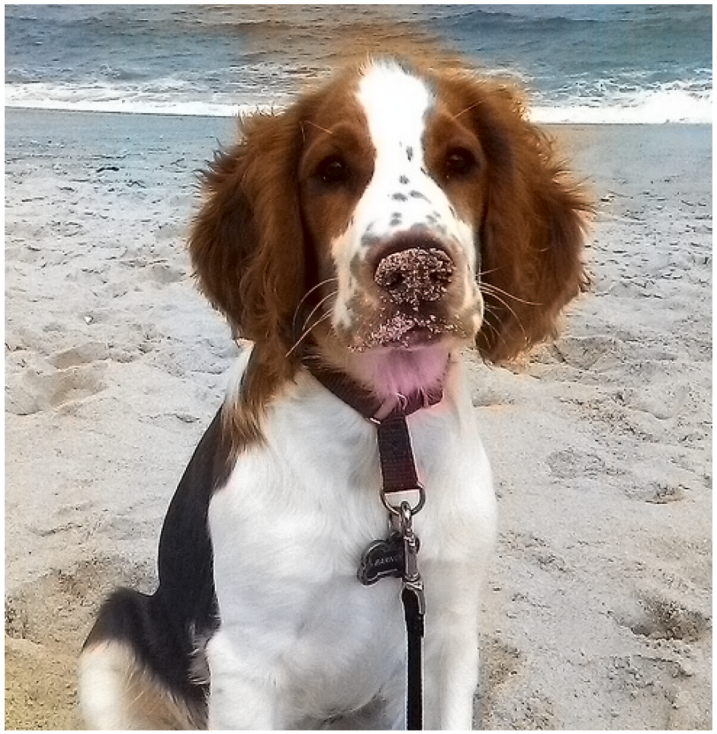
# Inherent Ambiguity

Prediction

Ground Truth

Source: A. Efros, R. Zhang

# Biases

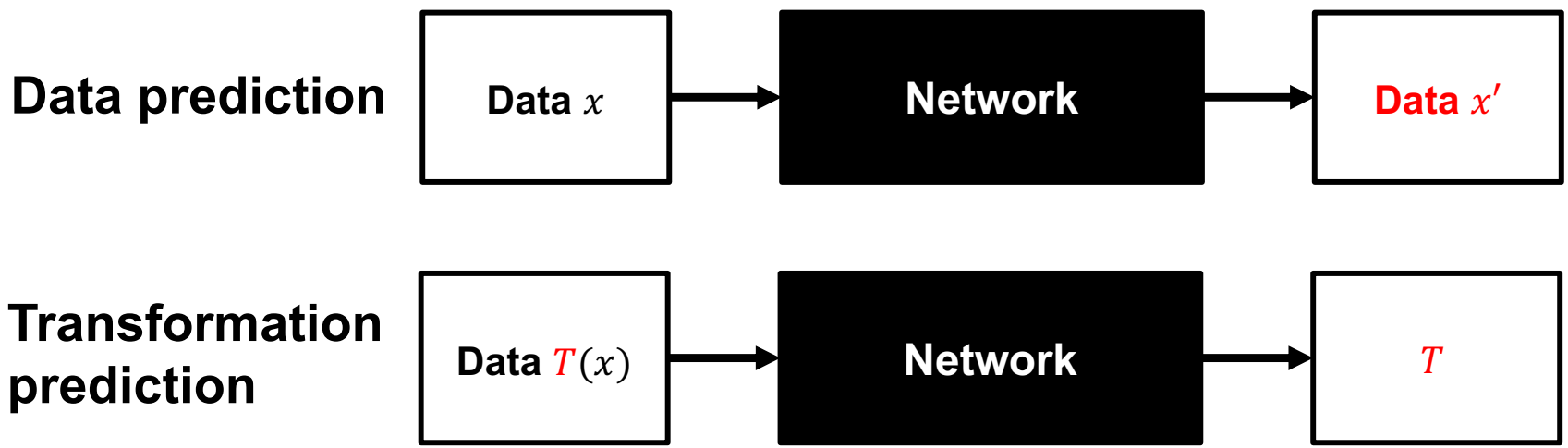


Source: A. Efros, R. Zhang

# Biases

Source: A. Efros, R. Zhang

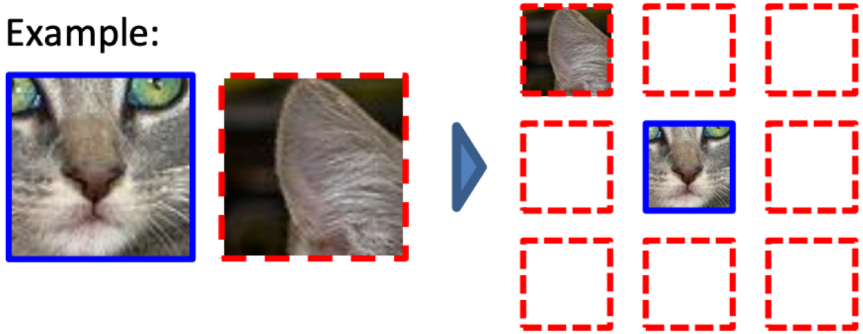
# Self-supervised learning: Outline



# Context prediction

- *Pretext task*: randomly sample a patch and one of 8 neighbors
- Guess the spatial relationship between the patches

Example:



Question 1:

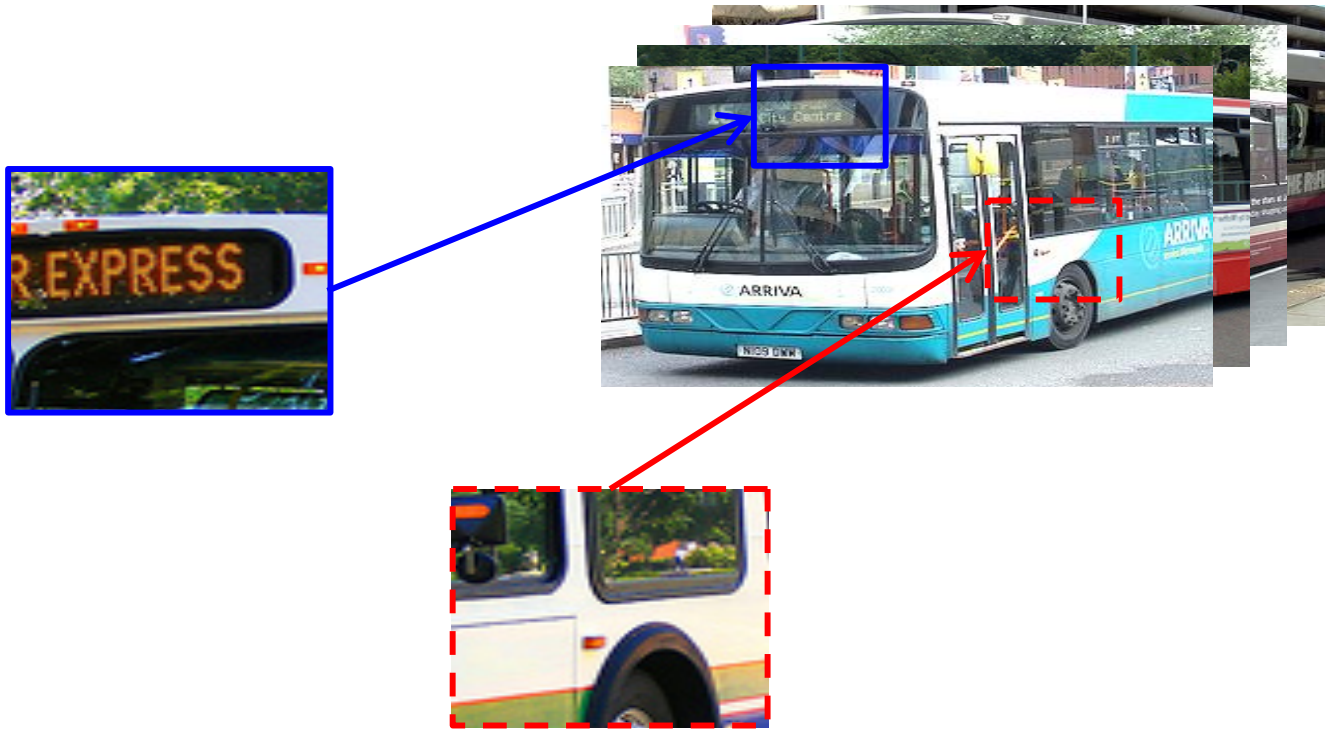

**A: Bottom right**

Question 2:


**A: Top center**

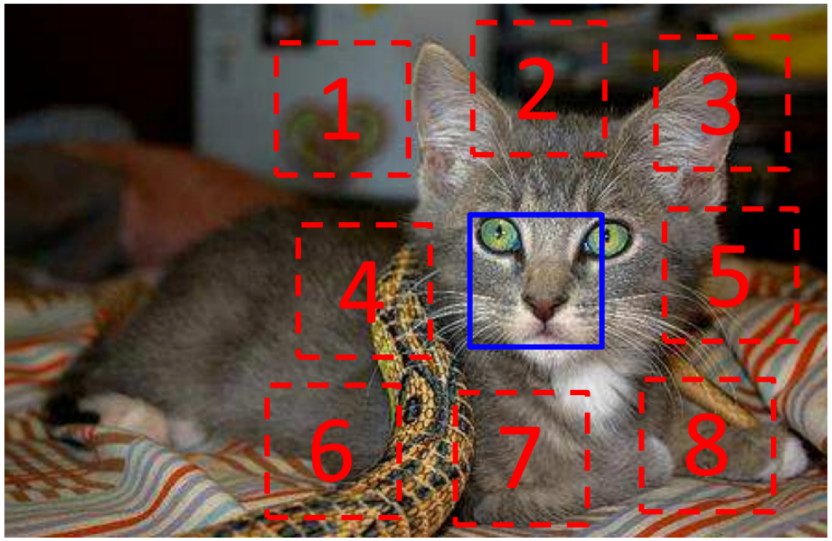
C. Doersch, A. Gupta, A. Efros. Unsupervised Visual Representation Learning by Context Prediction. ICCV 2015

# Context prediction: Semantics from a non-semantic task



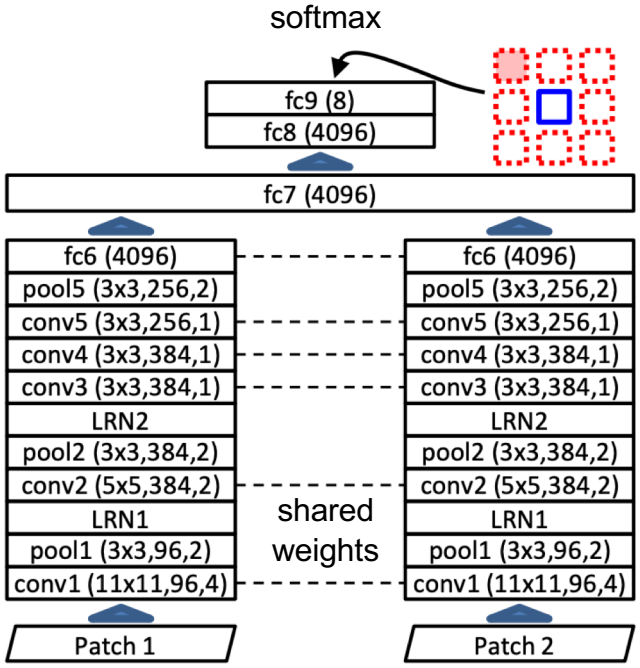

Source: A. Efros

# Context prediction: Details




Prevent “cheating”: sample patches with gaps, pre-process to overcome chromatic aberration

AlexNet-like architecture




C. Doersch, A. Gupta, A. Efros. Unsupervised Visual Representation Learning by Context Prediction. ICCV 2015

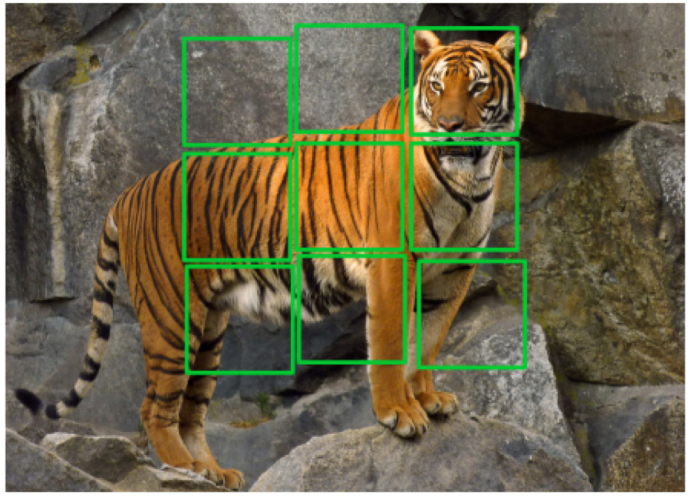
# Context prediction: Results

- Combine learned feature extractor with R-CNN to perform detection on PASCAL VOC 2007
- Unsupervised pre-training is 5% mAP better than training from scratch, but still 8% below pre-training with ImageNet label supervision

C. Doersch, A. Gupta, A. Efros. Unsupervised Visual Representation Learning by Context Prediction. ICCV 2015

# Jigsaw puzzle solving

Crop out tiles



Shuffle

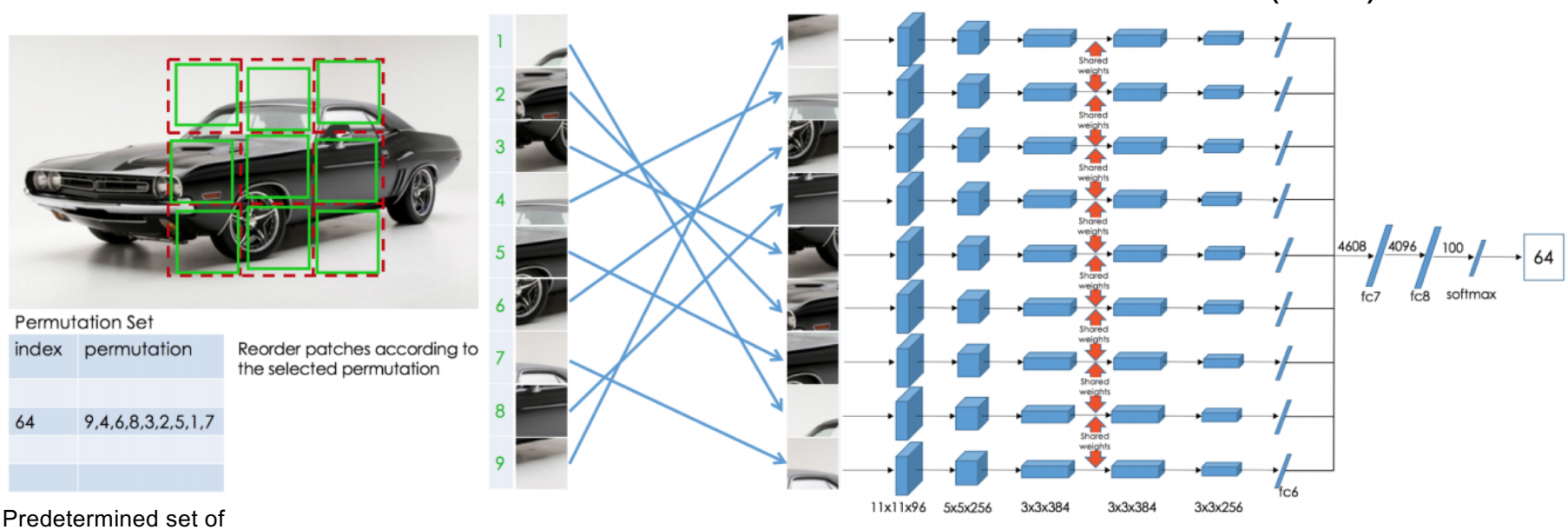
Pretext task: reassemble

Claim: jigsaw solving is easier than context prediction, trains faster, transfers better

M. Noroozi and P. Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. ECCV 2016

# Jigsaw puzzle solving: Details

Context free network (CFN)

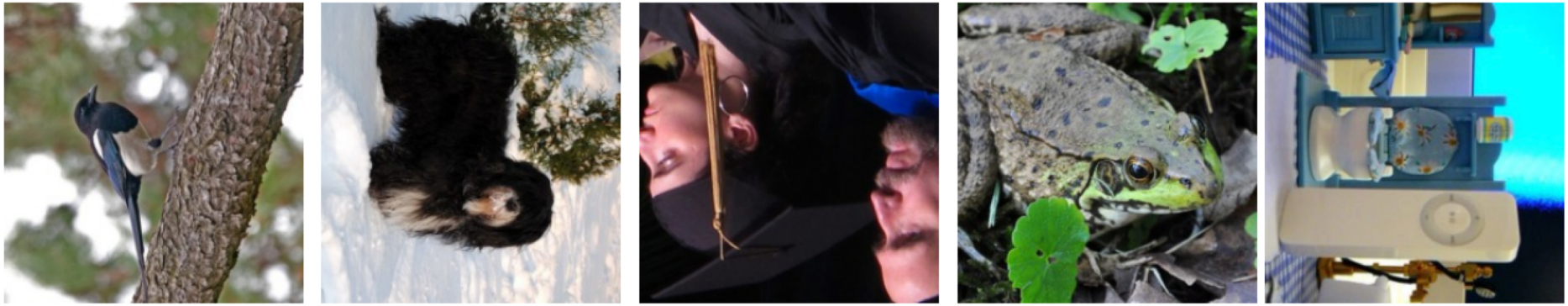



Predetermined set of 1000 permutations (out of 362,880 possible)

M. Noroozi and P. Favaro. [Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles]. ECCV 2016

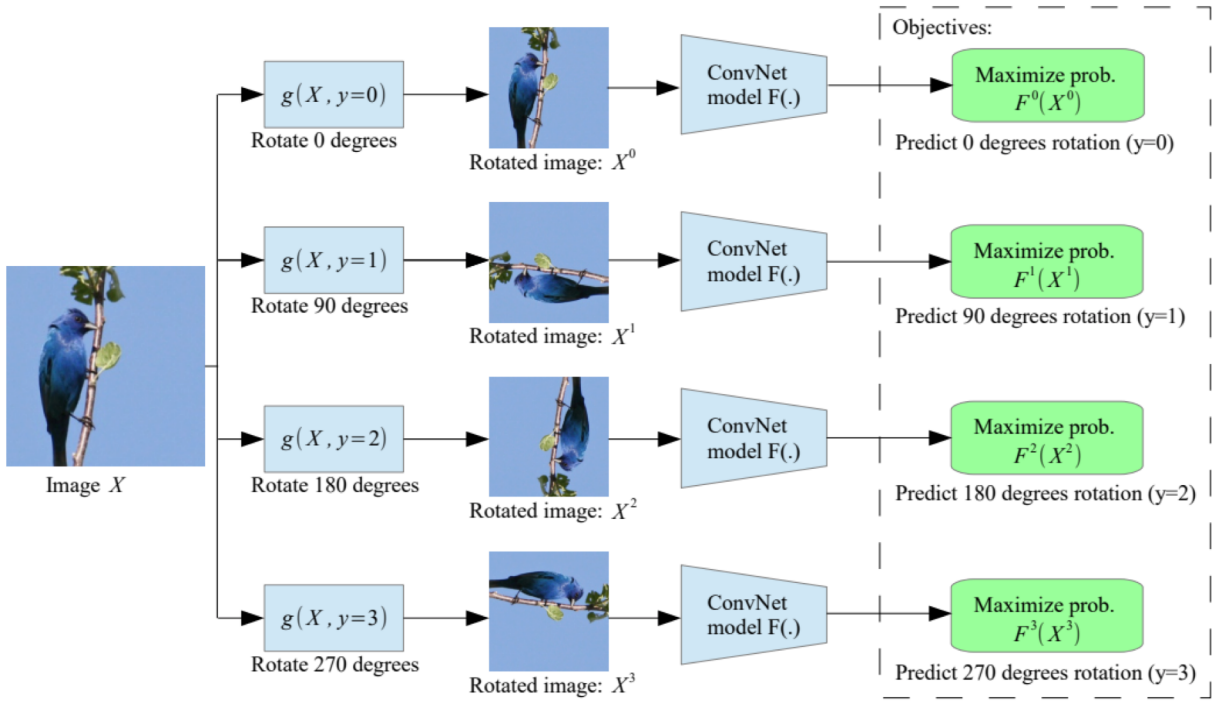
# Rotation prediction

- Pretext task: recognize image rotation (0, 90, 180, 270 degrees)



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

# Rotation prediction



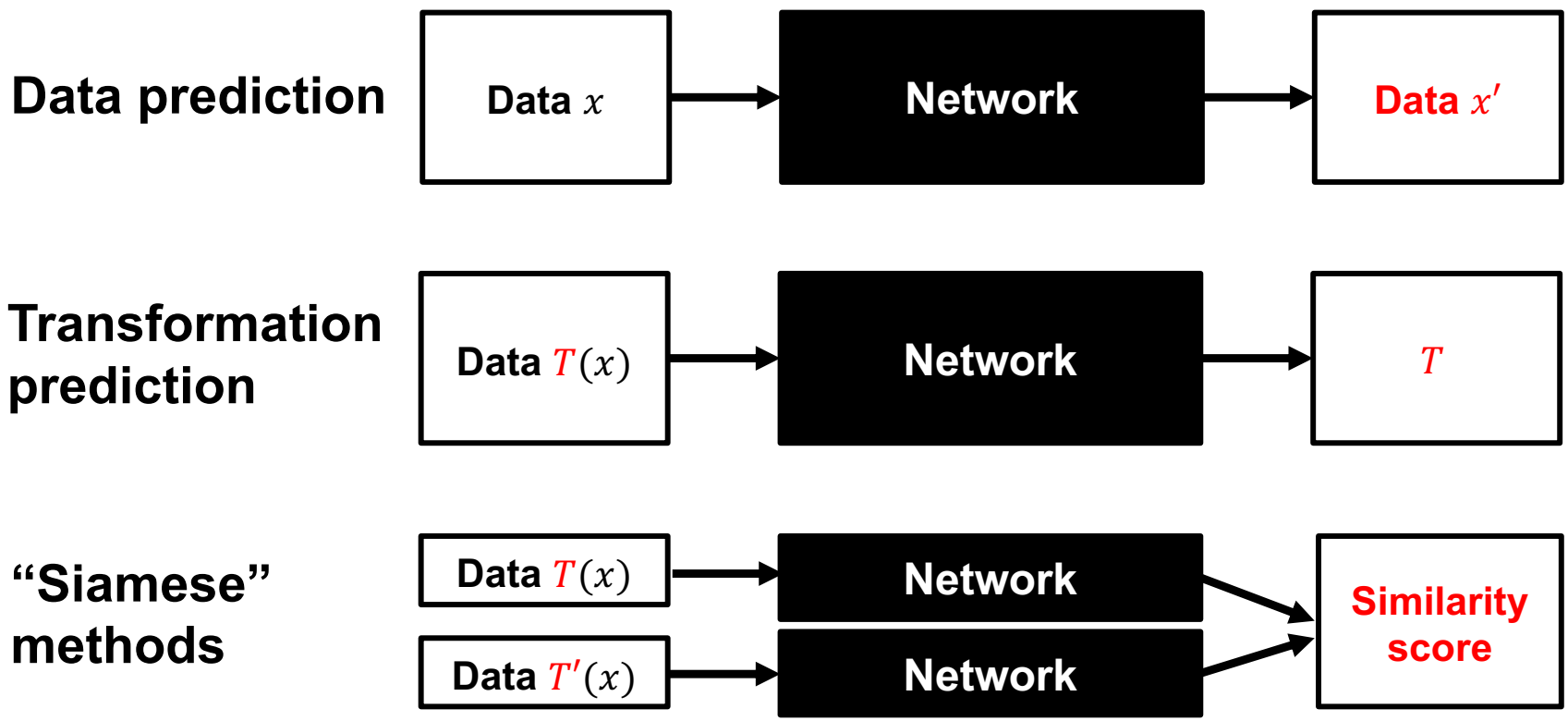

During training, feed in all four rotated versions of an image in the same mini-batch

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

# Rotation prediction: PASCAL VOC Transfer results

| Method | Classification | Detection (mAP) | Segmentation (mIoU) |
|---|---|---|---|
| Supervised (ImageNet) | **79.9** | **56.8** | **48.0** |
| Colorization | 65.6 | 46.9 | 35.6 |
| Context | 65.3 | 51.1 | |
| Jigsaw | 67.6 | 53.2 | 37.6 |
| Rotation | 73.0 | 54.4 | 39.1 |

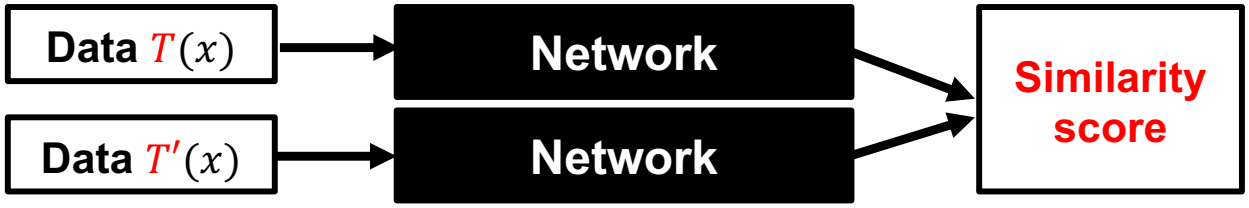
# Last time: Self-supervised learning




Source: A. Efros

# “Siamese” methods

- Extract representations from two transformed versions of a data point, encourage these representations to be similar
  - **Contrastive methods**: train using both positive (similar) and negative (dissimilar) pairs
  - **Non-contrastive methods**: train with only positive examples




Source: A. Efros

# Contrastive methods

- Encourage representations of transformed versions of the same image to be the same and different images to be different

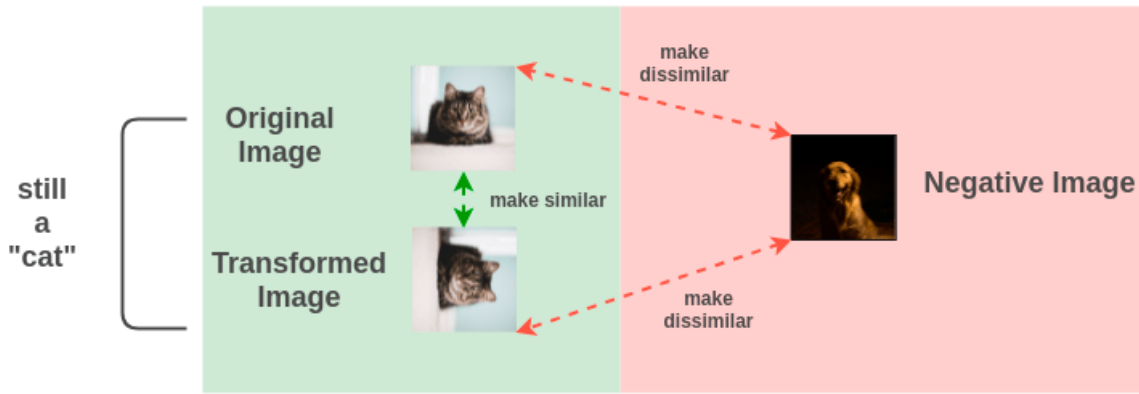



Figure source

# Contrastive loss formulation

- Given:
  - *Query* point $x$
  - *Positive* sample $x^+$: version of $x$ subjected to a random transformation or augmentation (cropping, rotation, color change, etc.)
  - *Negative* samples $x^-$

$x$ $x^+$ $x^-$

# Contrastive loss formulation

- Given: query $x$, positive sample $x^+$, negative samples $x^-$
- Measure similarity by dot product of L2-normalized feature representations:

$$\mathrm{sim}(x, y) = \frac{f(x)}{\|f(x)\|_2} \cdot \frac{f(y)}{\|f(y)\|_2}$$

- **Contrastive loss**: make $x$ similar to $x^+$, dissimilar from $x^-$:

$$l(x, x^+) = -\log \frac{\exp(\mathrm{sim}(x, x^+)/\tau)}{\exp(\mathrm{sim}(x, x^+)/\tau) + \sum_{j=1}^{N} \exp\left(\mathrm{sim}(x, x_j^-)/\tau\right)}$$

- Intuitively, this is the loss of a softmax classifier that tries to classify $x$ as $x^+$

# Recall: Softmax with temperature

Low temperature:
More concentrated
distribution

Higher temperature:
More uniform
distribution

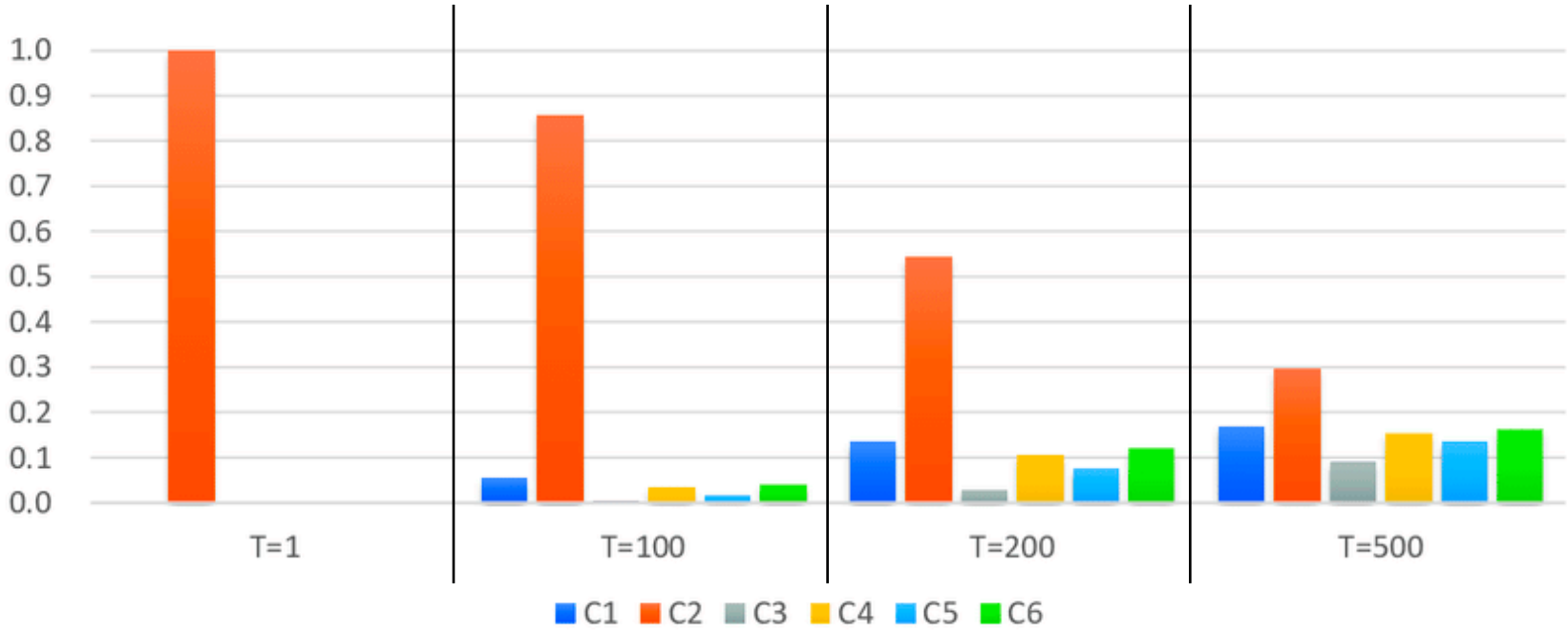
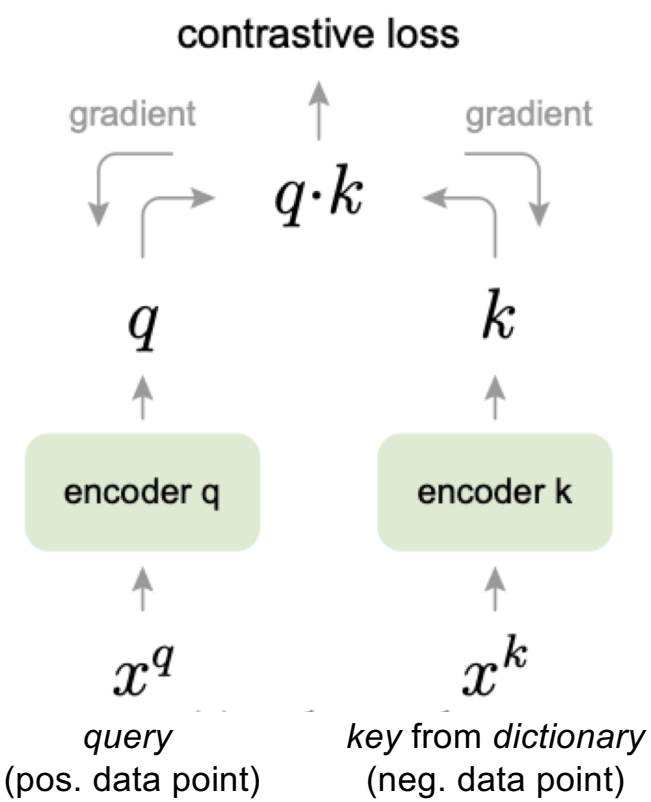



Figure source

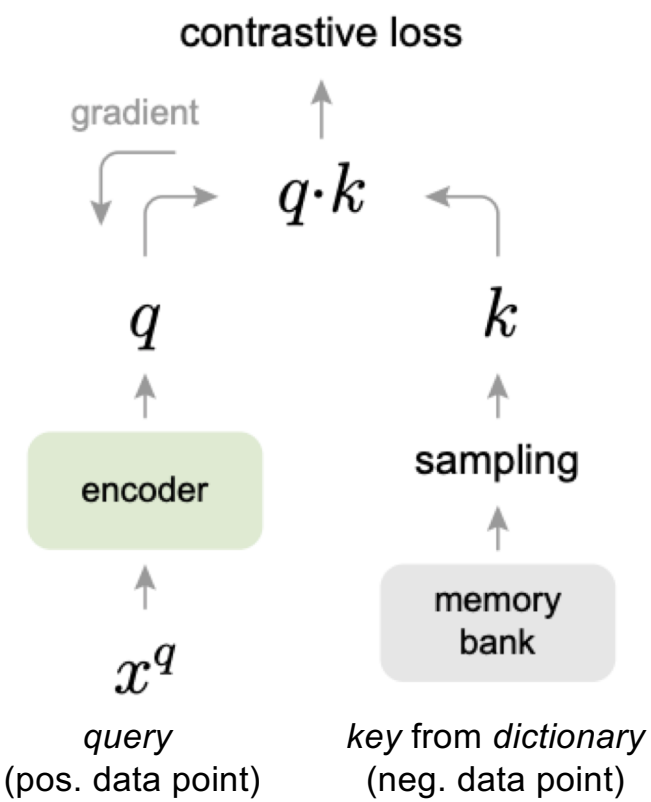
# Mechanisms for obtaining negative samples




- **End-to-end**: dictionary of negative examples is the current minibatch
  - Pro: dictionary is encoded using latest encoder parameters
  - Con: needs large minibatches to work well, but back-propagating through negative samples is compute- and memory-intensive

K. He et al. Momentum Contrast for Unsupervised Visual Representation Learning. CVPR 2020

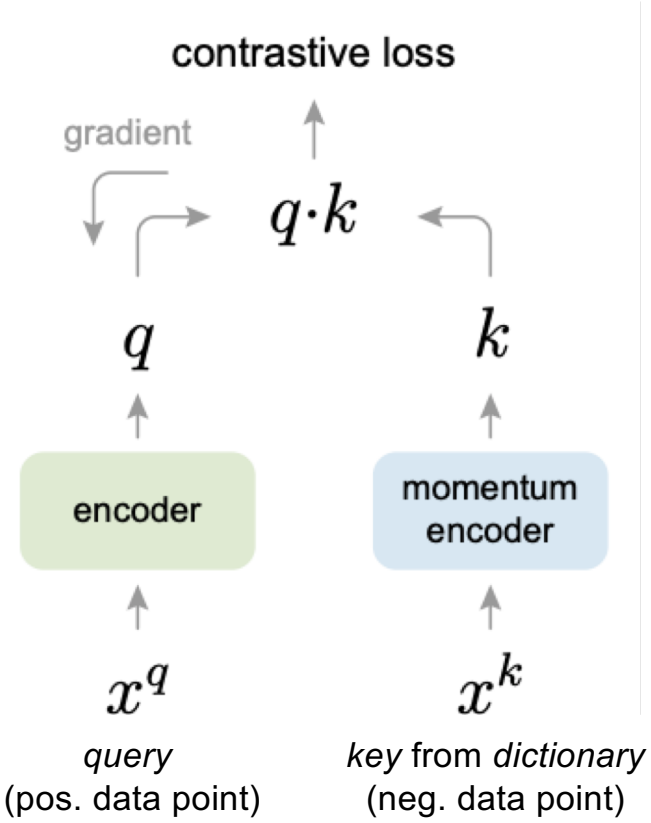
# Mechanisms for obtaining negative samples




- **Memory bank** of encodings of many points in the dataset – each point is encoded when last seen
  - Pros: no need to back-propagate through or re-encode memory bank entries, not limited by minibatch size
  - Con: memory bank entries are inconsistently encoded with different parameters, many become stale

K. He et al. Momentum Contrast for Unsupervised Visual Representation Learning. CVPR 2020

# Mechanisms for obtaining negative samples




- **Momentum encoder:** queue in which each example is encoded using slowly changing parameters $w_k$ updated as $w_k \leftarrow m\, w_k + (1 - m) w_q$, where $m = 0.999$ is the *momentum coefficient*
  - Pros: no need to re-encode dictionary examples, queue removes the most stale examples, slowly changing momentum encoder parameters allow for good learning

K. He et al. Momentum Contrast for Unsupervised Visual Representation Learning. CVPR 2020

# Mechanisms for obtaining negative samples

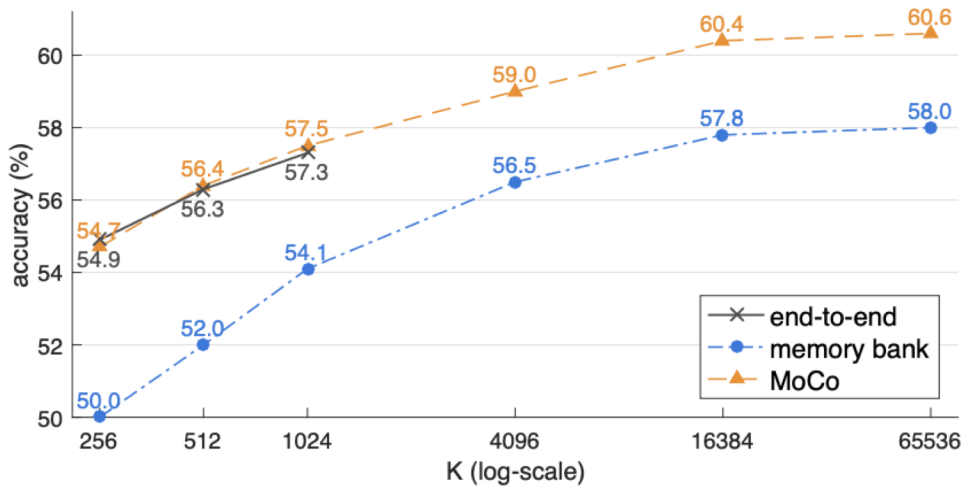
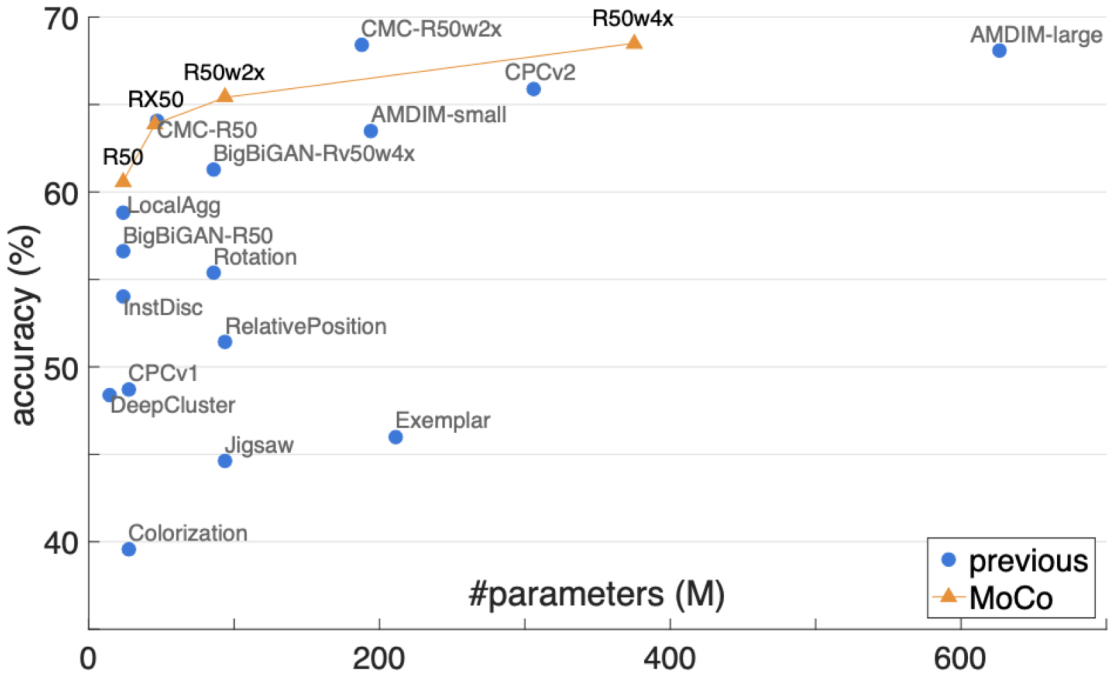



Figure 3. **Comparison of three contrastive loss mechanisms** under the ImageNet linear classification protocol. We adopt the same pretext task (Sec. 3.3) and only vary the contrastive loss mechanism (Figure 2). The number of negatives is $K$ in memory bank and MoCo, and is $K-1$ in end-to-end (offset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

K. He et al. Momentum Contrast for Unsupervised Visual Representation Learning. CVPR 2020

# MoCo results

Comparison on linear ImageNet classification
(supervised accuracy above 75%)




K. He et al. [Momentum Contrast for Unsupervised Visual Representation Learning](). CVPR 2020

# MoCo results

Comparison on PASCAL
object detection

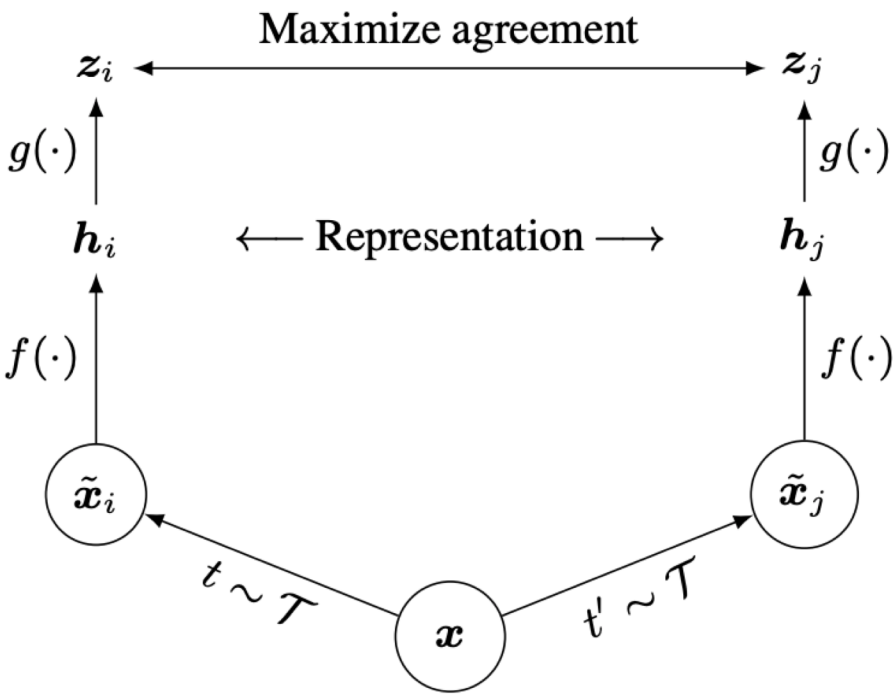
| pre-train | $AP_{50}$ | AP | $AP_{75}$ |
|---|---|---|---|
| random init. | 60.2 | 33.8 | 33.1 |
| super. IN-1M | 81.3 | 53.5 | 58.8 |
| **MoCo** IN-1M | 81.5 (+0.2) | 55.9 (+**2.4**) | 62.6 (+**3.8**) |
| **MoCo** IG-1B | 82.2 (+**0.9**) | 57.2 (+**3.7**) | 63.7 (+**4.9**) |

(b) Faster R-CNN, R50-**C4**

Table 2. **Object detection fine-tuned on PASCAL VOC** `trainval07+12`. Evaluation is on `test2007`: $AP_{50}$ (default VOC metric), AP (COCO-style), and $AP_{75}$, averaged over 5 trials. All are fine-tuned for 24k iterations (~23 epochs). In the brackets are the gaps to the ImageNet supervised pre-training counterpart. In green are the gaps of at least +**0.5** point.

K. He et al. Momentum Contrast for Unsupervised Visual Representation Learning. CVPR 2020

# SimCLR




- Instead of memory bank or queue, use large mini-batch size (on cloud TPU)
- Introduce nonlinear *projection* ($g$) between representation ($h$) and feature used for computing contrastive loss ($z$)

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. ICML 2020

# SimCLR

- Performed extensive ablation study of data augmentations
- Found that composing multiple augmentations gives the best results

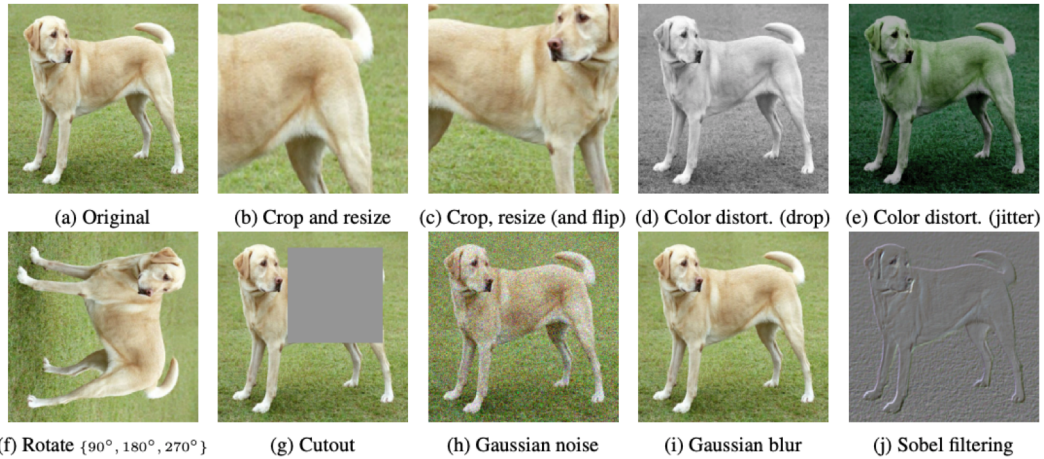



*Figure 4.* Illustrations of the studied data augmentation operators. Each augmentation can transform data stochastically with some internal parameters (e.g. rotation degree, noise level). Note that we *only* test these operators in ablation, the *augmentation policy used to train our models* only includes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

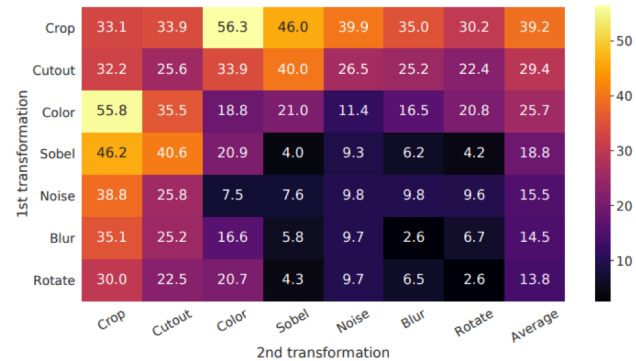
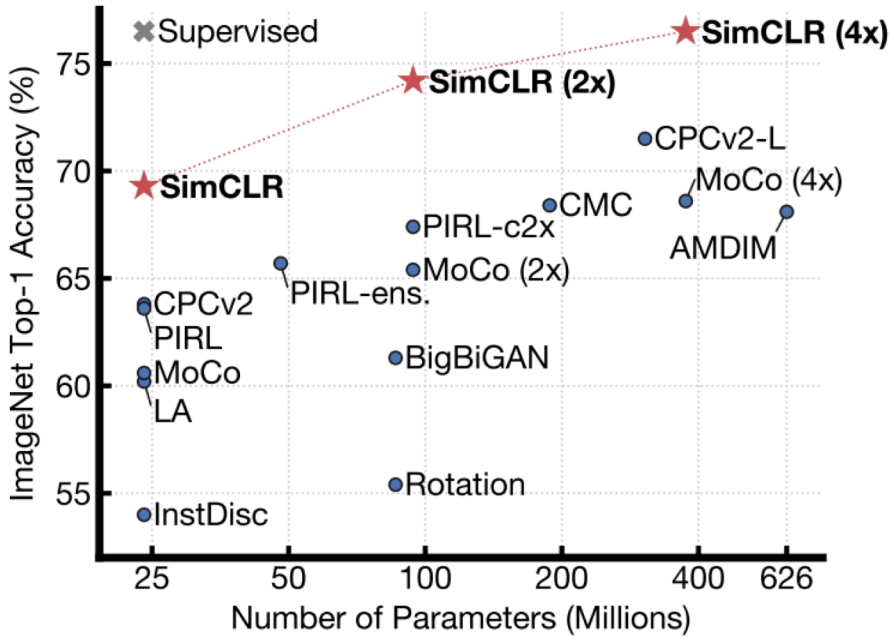



*Figure 5.* Linear evaluation (ImageNet top-1 accuracy) under individual or composition of data augmentations, applied only to one branch. For all columns but the last, diagonal entries correspond to single transformation, and off-diagonals correspond to composition of two transformations (applied sequentially). The last column reflects the average over the row.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. ICML 2020

# SimCLR: Evaluation

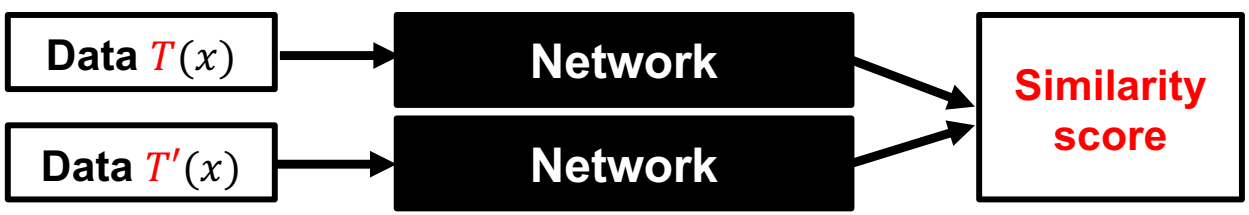



No detection evaluation

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. ICML 2020

# Non-contrastive methods

- Extract representations from two transformed versions of a data point, encourage these representations to be similar
  - **Contrastive methods**: train using both positive (similar) and negative (dissimilar) pairs
    - Key challenge: sampling of negative pairs
  - **Non-contrastive methods**: train with only positive examples
    - Key challenge: avoiding degenerate solutions (all representations collapsing to constant output value)




Source: A. Efros

# BYOL

- *Online branch* uses projection like SimCLR, plus an additional prediction head to predict the target representation
- *Target branch* uses momentum encoder, but there is no need for queue of negative examples

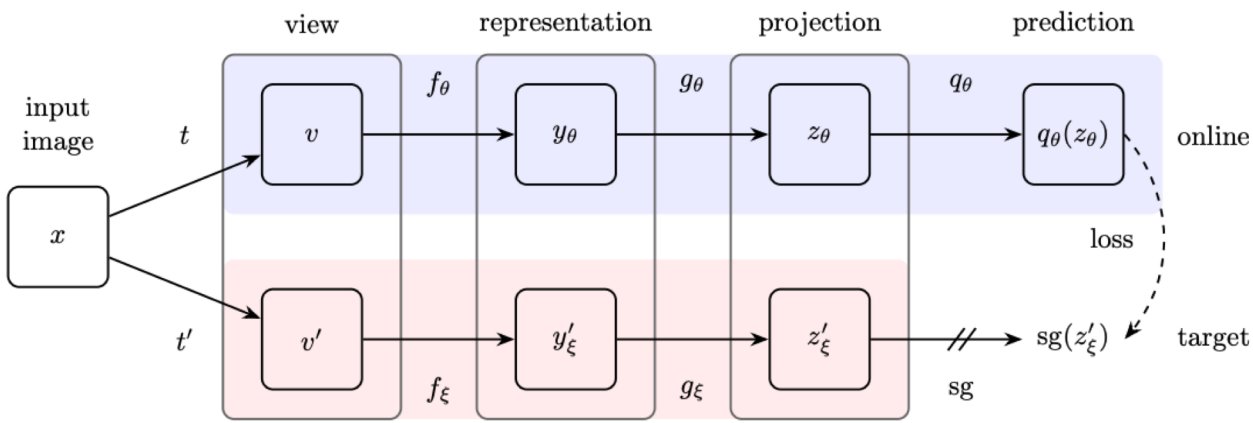



Figure 2: BYOL's architecture. BYOL minimizes a similarity loss between $q_\theta(z_\theta)$ and $\mathrm{sg}(z'_\xi)$, where $\theta$ are the trained weights, $\xi$ are an exponential moving average of $\theta$ and sg means stop-gradient. At the end of training, everything but $f_\theta$ is discarded,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

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

# BYOL: Evaluation

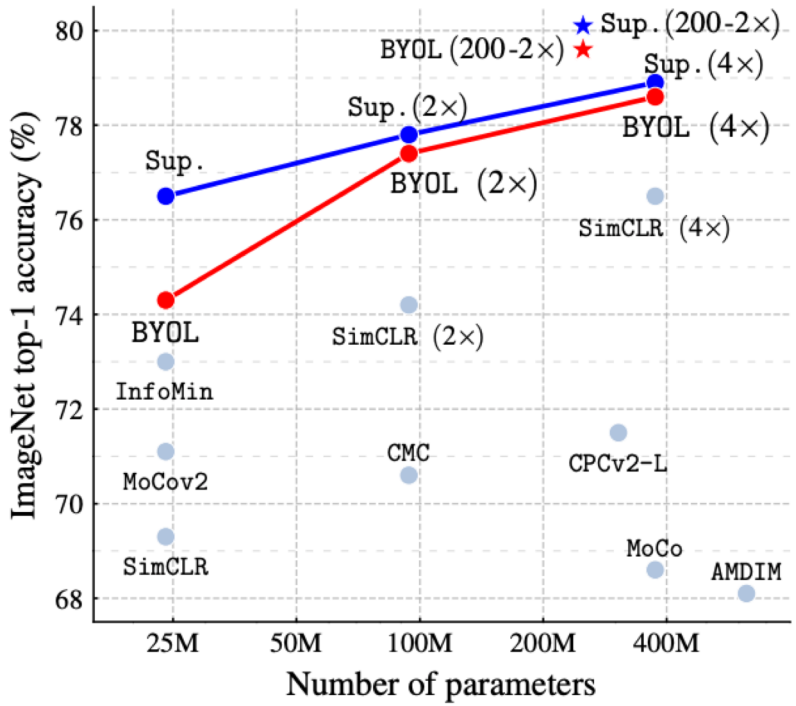



Figure 1: Performance of BYOL on ImageNet (linear evaluation) using ResNet-50 and our best architecture ResNet-200 (2×), compared to other unsupervised and supervised (Sup.) baselines [8].

# SimSiam

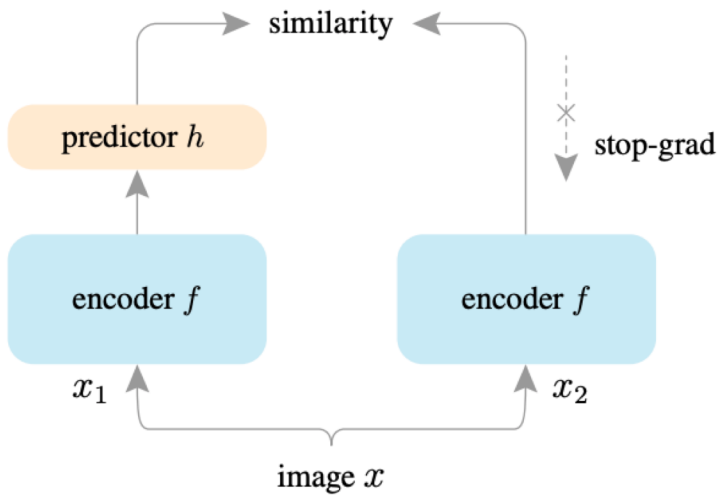



Figure 1. **SimSiam architecture**. Two augmented views of one image are processed by the same encoder network $f$ (a backbone plus a projection MLP). Then a prediction MLP $h$ is applied on one side, and a stop-gradient operation is applied on the other side. The model maximizes the similarity between both sides. It uses neither negative pairs nor a momentum encoder.

- Like BYOL, but without the momentum encoder
- Like SimCLR, but without the negative pairs

X. Chen and K. He. Exploring Simple Siamese Representation Learning. CVPR 2021

# SimSiam: Evaluation

| method | batch size | negative pairs | momentum encoder | 100 ep | 200 ep | 400 ep | 800 ep |
|---|---|---|---|---|---|---|---|
| SimCLR (repro.+) | 4096 | ✓ | | 66.5 | 68.3 | 69.8 | 70.4 |
| MoCo v2 (repro.+) | **256** | ✓ | ✓ | 67.4 | 69.9 | 71.0 | 72.2 |
| BYOL (repro.) | 4096 | | ✓ | 66.5 | **70.6** | **73.2** | **74.3** |
| SwAV (repro.+) | 4096 | | | 66.5 | 69.1 | 70.7 | 71.8 |
| **SimSiam** | **256** | | | **68.1** | 70.0 | 70.8 | 71.3 |

Table 4. **Comparisons on ImageNet linear classification**. All are based on **ResNet-50** pre-trained with **two 224×224 views**. Evaluation is on a single crop. All competitors are from our reproduction, and "+" denotes *improved* reproduction *vs*. original papers (see supplement).

X. Chen and K. He. Exploring Simple Siamese Representation Learning. CVPR 2021

# SimSiam: Evaluation

| pre-train | VOC 07 detection | | | VOC 07+12 detection | | | COCO detection | | | COCO instance seg. | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $AP_{50}$ | AP | $AP_{75}$ | $AP_{50}$ | AP | $AP_{75}$ | $AP_{50}$ | AP | $AP_{75}$ | $AP^{mask}_{50}$ | $AP^{mask}$ | $AP^{mask}_{75}$ |
| scratch | 35.9 | 16.8 | 13.0 | 60.2 | 33.8 | 33.1 | 44.0 | 26.4 | 27.8 | 46.9 | 29.3 | 30.8 |
| ImageNet supervised | 74.4 | 42.4 | 42.7 | 81.3 | 53.5 | 58.8 | 58.2 | 38.2 | 41.2 | 54.7 | 33.3 | 35.2 |
| SimCLR (repro.+) | 75.9 | 46.8 | 50.1 | 81.8 | 55.5 | 61.4 | 57.7 | 37.9 | 40.9 | 54.6 | 33.3 | 35.3 |
| MoCo v2 (repro.+) | **77.1** | **48.5** | **52.5** | **82.3** | **57.0** | **63.3** | **58.8** | **39.2** | **42.5** | **55.5** | **34.3** | **36.6** |
| BYOL (repro.) | **77.1** | 47.0 | 49.9 | 81.4 | 55.3 | 61.1 | 57.8 | 37.9 | 40.9 | 54.3 | 33.2 | 35.0 |
| SwAV (repro.+) | 75.5 | 46.5 | 49.6 | 81.5 | 55.4 | 61.4 | 57.6 | 37.6 | 40.3 | 54.2 | 33.1 | 35.1 |
| **SimSiam**, base | 75.5 | 47.0 | 50.2 | **82.0** | 56.4 | 62.8 | 57.5 | 37.9 | 40.9 | 54.2 | 33.2 | 35.2 |
| **SimSiam**, optimal | **77.3** | **48.5** | **52.5** | **82.4** | **57.0** | **63.7** | **59.3** | **39.2** | **42.1** | **56.0** | **34.4** | **36.7** |

Table 5. **Transfer Learning**. All unsupervised methods are based on 200-epoch pre-training in ImageNet. *VOC 07 detection*: Faster R-CNN [30] fine-tuned in VOC 2007 trainval, evaluated in VOC 2007 test; *VOC 07+12 detection*: Faster R-CNN fine-tuned in VOC 2007 trainval + 2012 train, evaluated in VOC 2007 test; *COCO detection* and *COCO instance segmentation*: Mask R-CNN [18] (1× schedule) fine-tuned in COCO 2017 train, evaluated in COCO 2017 val. All Faster/Mask R-CNN models are with the C4-backbone [13]. All VOC results are the average over 5 trials. **Bold entries** are within 0.5 below the best.

X. Chen and K. He. Exploring Simple Siamese Representation Learning. CVPR 2021

# DINO: Self-distillation with no labels

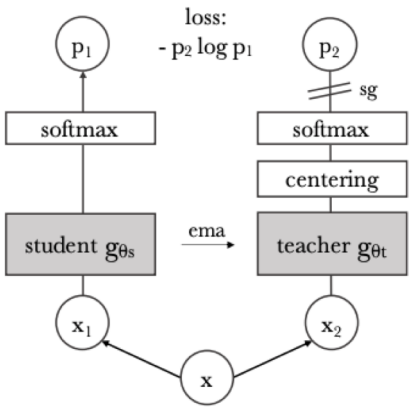


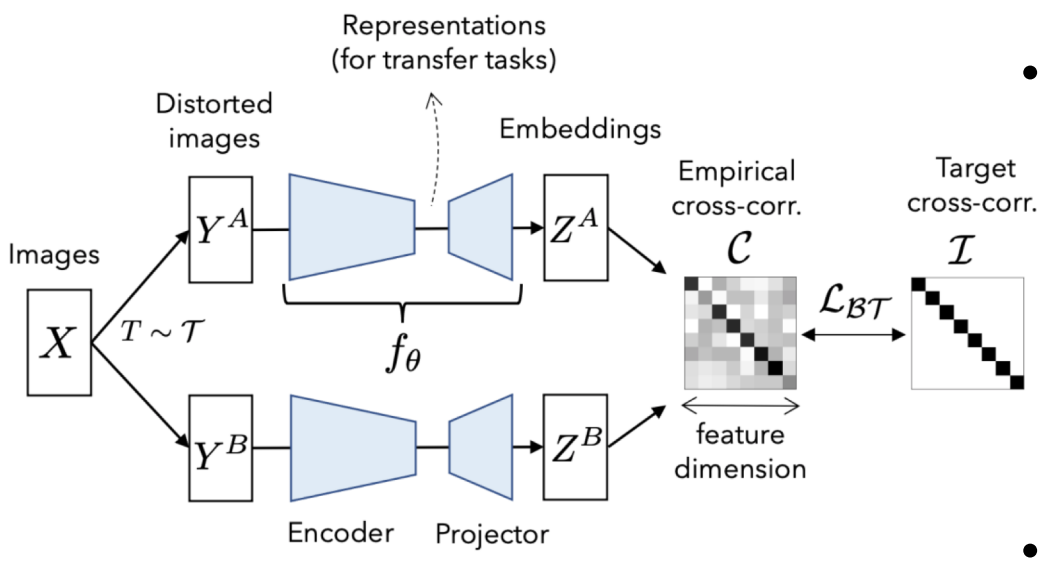

Figure 2: **Self-distillation with no labels.** We illustrate DINO in the case of one single pair of views ($x_1$, $x_2$) for simplicity. The model passes two different random transformations of an input image to the student and teacher networks. Both networks have the same architecture but different parameters. The output of the teacher network is centered with a mean computed over the batch. Each networks outputs a $K$ dimensional feature that is normalized with a temperature softmax over the feature dimension. Their similarity is then measured with a cross-entropy loss. We apply a stop-gradient (sg) operator on the teacher to propagate gradients only through the student. The teacher parameters are updated with an exponential moving average (ema) of the student parameters.

**Algorithm 1** DINO PyTorch pseudocode w/o multi-crop.

```
# gs, gt: student and teacher networks
# C: center (K)
# tps, tpt: student and teacher temperatures
# l, m: network and center momentum rates
gt.params = gs.params
for x in loader: # load a minibatch x with n samples
    x1, x2 = augment(x), augment(x) # random views

    s1, s2 = gs(x1), gs(x2) # student output n-by-K
    t1, t2 = gt(x1), gt(x2) # teacher output n-by-K

    loss = H(t1, s2)/2 + H(t2, s1)/2
    loss.backward() # back-propagate

    # student, teacher and center updates
    update(gs) # SGD
    gt.params = l*gt.params + (1-l)*gs.params
    C = m*C + (1-m)*cat([t1, t2]).mean(dim=0)

def H(t, s):
    t = t.detach() # stop gradient
    s = softmax(s / tps, dim=1)
    t = softmax((t - C) / tpt, dim=1) # center + sharpen
    return - (t * log(s)).sum(dim=1).mean()
```

M. Caron et al. [Emerging Properties in Self-Supervised Vision Transformers](). ICCV 2021

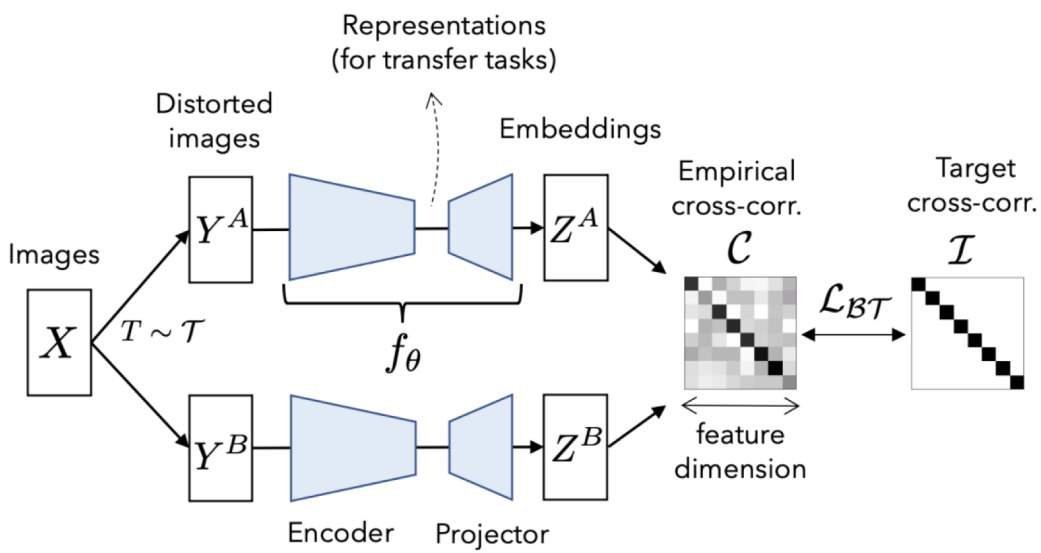
# Barlow Twins




- Get rid of asymmetric encoders, stop-gradients
- To avoid degeneracy, force cross-correlation matrix of "twin" embeddings to be close to identity – this tries to make individual dimensions of embeddings non-redundant
- Use projection head that *expands* dimensions

J. Zbontar et al. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. ICML 2021

# Barlow Twins




$$\mathcal{L}_{\mathcal{BT}} \triangleq \underbrace{\sum_i (1-\mathcal{C}_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j \neq i} \mathcal{C}_{ij}{}^2}_{\text{redundancy reduction term}}$$

$$\mathcal{C}_{ij} \triangleq \frac{\sum_b z^A_{b,i} z^B_{b,j}}{\sqrt{\sum_b \left(z^A_{b,i}\right)^2} \sqrt{\sum_b \left(z^B_{b,j}\right)^2}}$$

*Figure 1.* BARLOW TWINS's objective function measures the cross-correlation matrix between the embeddings of two identical networks fed with distorted versions of a batch of samples, and tries to make this matrix close to the identity. This causes the embedding vectors of distorted versions of a sample to be similar, while minimizing the redundancy between the components of these vectors. BARLOW TWINS is competitive with state-of-the-art methods for self-supervised learning while being conceptually simpler, naturally avoiding trivial constant (i.e. collapsed) embeddings, and being robust to the training batch size.

J. Zbontar et al. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. ICML 2021

# Barlow Twins: Evaluation

*Table 1.* **Top-1 and top-5 accuracies (in %) under linear evaluation on ImageNet**. All models use a ResNet-50 encoder. Top-3 best self-supervised methods are underlined.

| Method | Top-1 | Top-5 |
|---|---|---|
| Supervised | 76.5 | |
| MoCo | 60.6 | |
| PIRL | 63.6 | - |
| SimCLR | 69.3 | 89.0 |
| MoCo v2 | 71.1 | 90.1 |
| SimSiam | 71.3 | - |
| SwAV (w/o multi-crop) | 71.8 | - |
| BYOL | 74.3 | 91.6 |
| SwAV | 75.3 | - |
| Barlow Twins (ours) | 73.2 | 91.0 |

J. Zbontar et al. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. ICML 2021

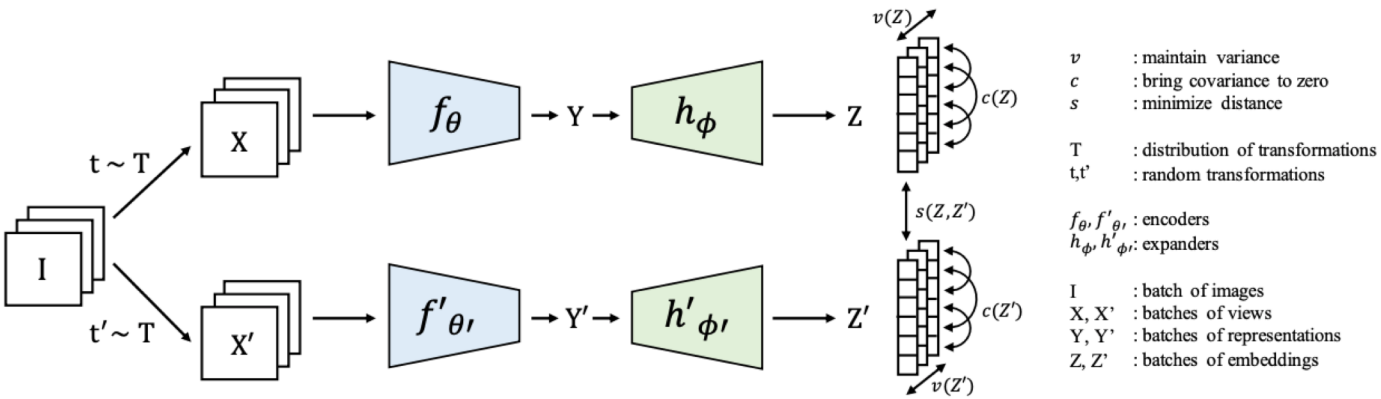
# Barlow Twins: Evaluation

*Table 4.* **Transfer learning: object detection and instance segmentation.** We benchmark learned representations on the object detection task on VOC07+12 using Faster R-CNN (Ren et al., 2015) and on the detection and instance segmentation task on COCO using Mask R-CNN (He et al., 2017). All methods use the C4 backbone variant (Wu et al., 2019) and models on COCO are finetuned using the 1× schedule. Best results are in **bold**.

| Method | VOC07+12 det | | | COCO det | | | COCO instance seg | | |
|---|---|---|---|---|---|---|---|---|---|
| | $AP_{all}$ | $AP_{50}$ | $AP_{75}$ | $AP^{bb}$ | $AP^{bb}_{50}$ | $AP^{bb}_{75}$ | $AP^{mk}$ | $AP^{mk}_{50}$ | $AP^{mk}_{75}$ |
| Sup. | 53.5 | 81.3 | 58.8 | 38.2 | 58.2 | 41.2 | 33.3 | 54.7 | 35.2 |
| MoCo-v2 | **57.4** | 82.5 | **64.0** | **39.3** | 58.9 | **42.5** | **34.4** | 55.8 | 36.5 |
| SwAV | 56.1 | **82.6** | 62.7 | 38.4 | 58.6 | 41.3 | 33.8 | 55.2 | 35.9 |
| SimSiam | 57 | 82.4 | 63.7 | 39.2 | **59.3** | 42.1 | **34.4** | **56.0** | **36.7** |
| BT (ours) | 56.8 | **82.6** | 63.4 | 39.2 | 59.0 | **42.5** | 34.3 | **56.0** | 36.5 |

J. Zbontar et al. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. ICML 2021

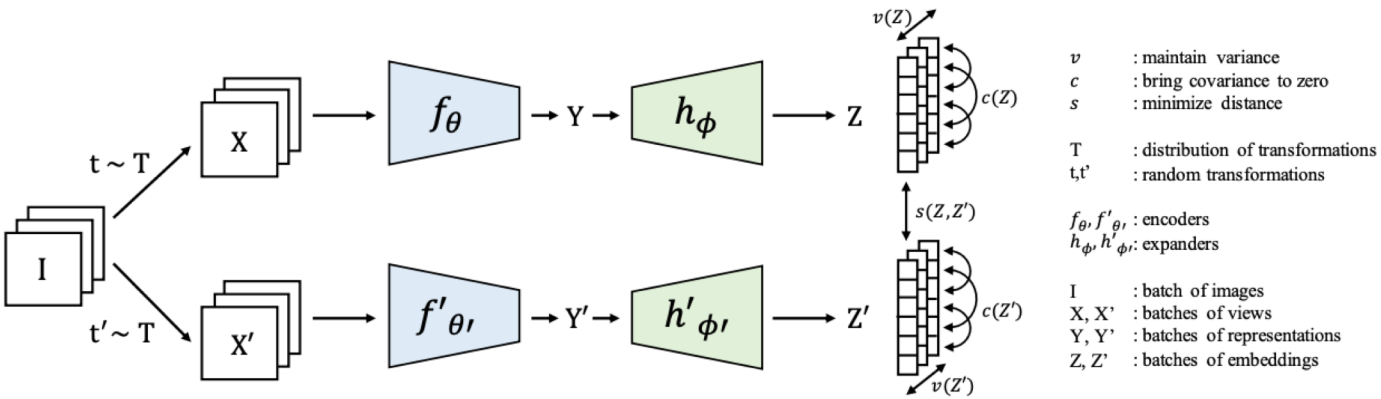
# VICReg




- **Invariance**: the mean square distance between the embedding vectors.
- **Variance**: a hinge loss to maintain the standard deviation (over a batch) of each variable of the embedding above a given threshold. This term forces the embedding vectors of samples within a batch to be different.
- **Covariance**: a term that attracts the covariances (over a batch) between every pair of (centered) embedding variables towards zero. This term decorrelates the variables of each embedding and prevents an *informational collapse* in which the variables would vary together or be highly correlated.

A. Bardes et al. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. ICLR 2022

# VICReg




- Advantages over Barlow Twins:
  - More general and applicable to cross-modal data: “does not require that the weights of the two branches be shared, not that the architectures be identical, nor that the inputs be of the same nature”
  - Including the variance preservation term removes the need to use normalized correlation matrix

A. Bardes et al. [VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning](). ICLR 2022

# VICReg: Evaluation

Table 1: **Evaluation on ImageNet.** Evaluation of the representations obtained with a ResNet-50 backbone pretrained with VICReg on: (1) linear classification on top of the frozen representations from ImageNet; (2) semi-supervised classification on top of the fine-tuned representations from 1% and 10% of ImageNet samples. We report Top-1 and Top-5 accuracies (in %). Top-3 best self-supervised methods are underlined.

| Method | Linear | | Semi-supervised | | | |
|---|---|---|---|---|---|---|
| | Top-1 | Top-5 | Top-1 | | Top-5 | |
| | | | 1% | 10% | 1% | 10% |
| Supervised | 76.5 | - | 25.4 | 56.4 | 48.4 | 80.4 |
| MoCo He et al. (2020) | 60.6 | - | - | - | - | - |
| PIRL Misra & Maaten (2020) | 63.6 | - | - | - | 57.2 | 83.8 |
| CPC v2 Hénaff et al. (2019) | 63.8 | - | - | - | - | - |
| CMC Tian et al. (2019) | 66.2 | - | - | - | - | - |
| SimCLR Chen et al. (2020a) | 69.3 | 89.0 | 48.3 | 65.6 | 75.5 | 87.8 |
| MoCo v2 Chen et al. (2020c) | 71.1 | - | - | - | - | - |
| SimSiam Chen & He (2020) | 71.3 | - | - | - | - | - |
| SwAV Caron et al. (2020) | 71.8 | - | - | - | - | - |
| InfoMin Aug Tian et al. (2020) | 73.0 | 91.1 | - | - | - | - |
| OBoW Gidaris et al. (2021) | 73.8 | - | - | - | 82.9 | 90.7 |
| BYOL Grill et al. (2020) | 74.3 | 91.6 | 53.2 | 68.8 | 78.4 | 89.0 |
| SwAV (w/ multi-crop) Caron et al. (2020) | 75.3 | - | 53.9 | 70.2 | 78.5 | 89.9 |
| Barlow Twins Zbontar et al. (2021) | 73.2 | 91.0 | 55.0 | 69.7 | 79.2 | 89.3 |
| VICReg (ours) | 73.2 | 91.1 | 54.8 | 69.5 | 79.4 | 89.5 |

A. Bardes et al. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. ICLR 2022

# VICReg: Evaluation

Table 2: **Transfer learning on downstream tasks.** Evaluation of the representations from a ResNet-50 backbone pretrained with VICReg on: (1) linear classification tasks on top of frozen representations, we report Top-1 accuracy (in %) for Places205 Zhou et al. (2014) and iNat18 Horn et al. (2018), and mAP for VOC07 Everingham et al. (2010); (2) object detection with fine-tunning, we report $AP_{50}$ for VOC07+12 using Faster R-CNN with C4 backbone Ren et al. (2015); (3) object detection and instance segmentation, we report AP for COCO Lin et al. (2014) using Mask R-CNN with FPN backbone He et al. (2017). We use † to denote the experiments run by us. Top-3 best self-supervised methods are underlined.

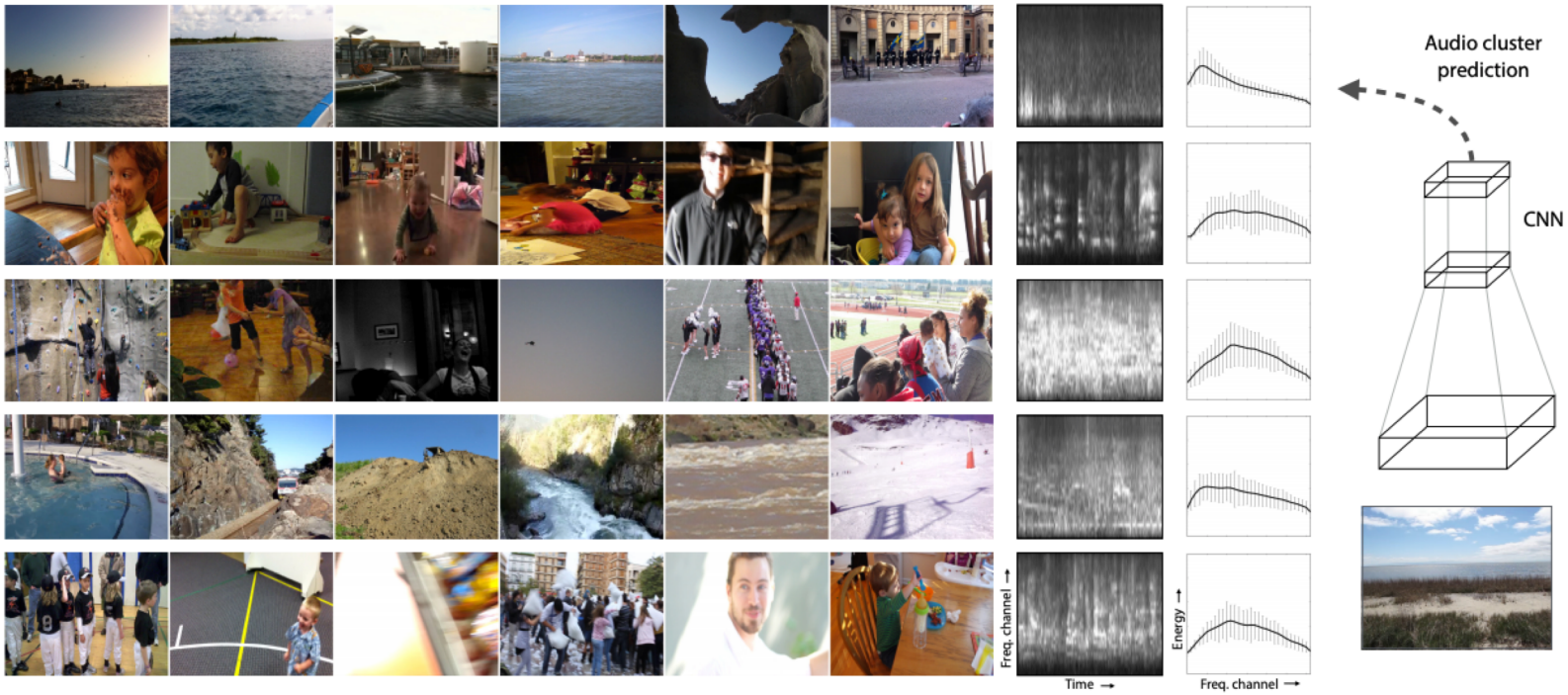
| | Linear Classification | | | Object Detection | | |
|---|---|---|---|---|---|---|
| Method | Places205 | VOC07 | iNat18 | VOC07+12 | COCO det | COCO seg |
| Supervised | 53.2 | 87.5 | 46.7 | 81.3 | 39.0 | 35.4 |
| MoCo He et al. (2020) | 46.9 | 79.8 | 31.5 | - | - | - |
| PIRL Misra & Maaten (2020) | 49.8 | 81.1 | 34.1 | - | - | - |
| SimCLR Chen et al. (2020a) | 52.5 | 85.5 | 37.2 | - | - | - |
| MoCo v2 Chen et al. (2020c) | 51.8 | 86.4 | 38.6 | 82.5 | 39.8 | 36.1 |
| SimSiam Chen & He (2020) | - | - | - | 82.4 | - | - |
| BYOL Grill et al. (2020) | 54.0 | 86.6 | 47.6 | - | 40.4† | 37.0† |
| SwAV (m-c) Caron et al. (2020) | 56.7 | 88.9 | 48.6 | 82.6 | 41.6 | 37.8 |
| OBoW Gidaris et al. (2021) | 56.8 | 89.3 | - | 82.9 | - | - |
| Barlow Twins Grill et al. (2020) | 54.1 | 86.2 | 46.5 | 82.6 | 40.0† | 36.7† |
| VICReg (ours) | 54.3 | 86.6 | 47.0 | 82.4 | 39.4 | 36.4 |

A. Bardes et al. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. ICLR 2022

# Self-supervised learning: Outline

- Data prediction
  - Colorization, inpainting
- Transformation prediction
  - Context prediction, jigsaw puzzle solving, rotation prediction
- “Siamese” methods
  - Contrastive methods
  - Non-contrastive methods
- Self-supervision beyond still images
  - Video, audio, language

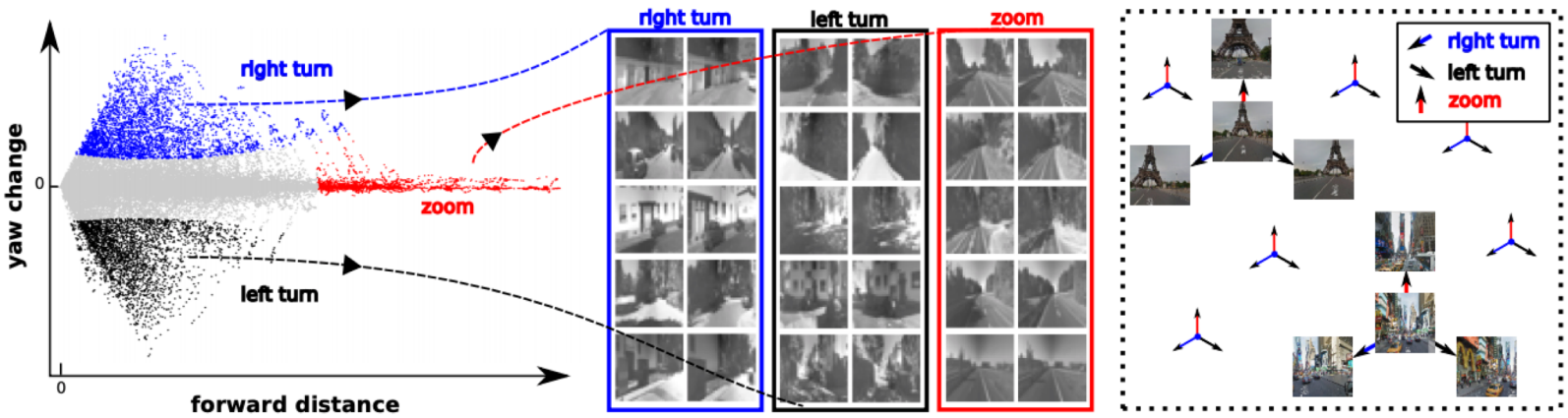
# Learning from audio




(a) Images grouped by audio cluster (b) Clustered audio stats. (c) CNN model

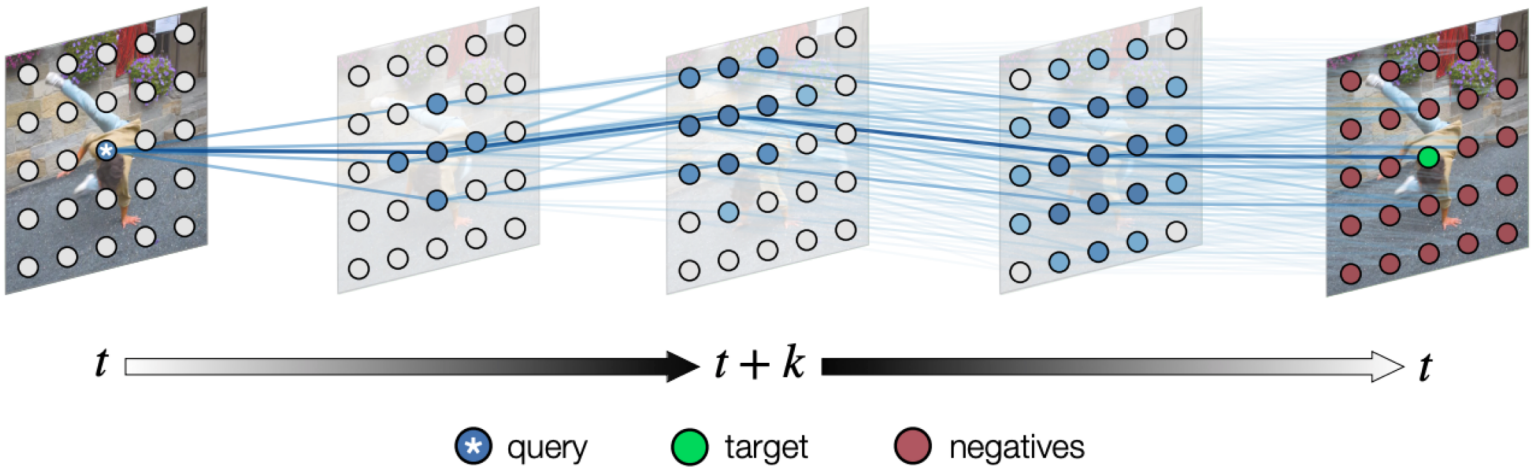
A. Owens et al. Ambient Sound Provides Supervision for Visual Learning. ECCV 2016

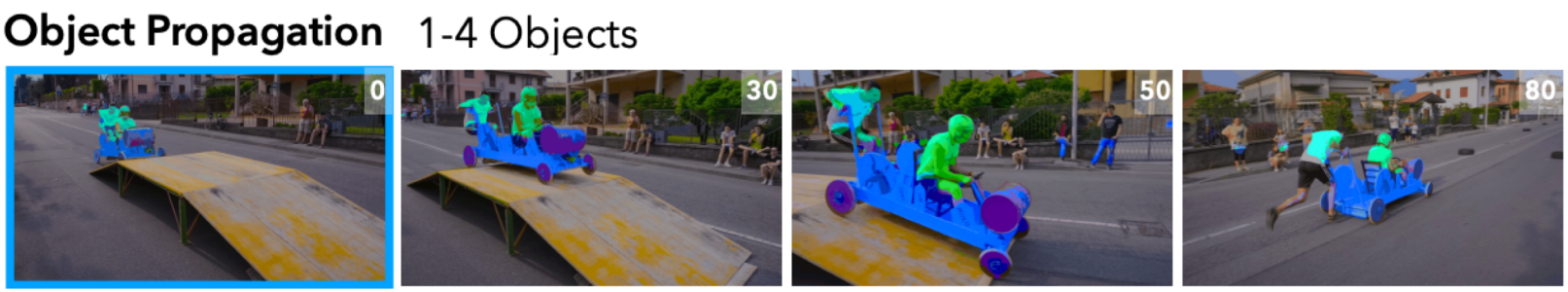
# Ego-motion features




D. Jayaraman and K. Grauman. Learning image representations tied to ego-motion. ICCV 2015

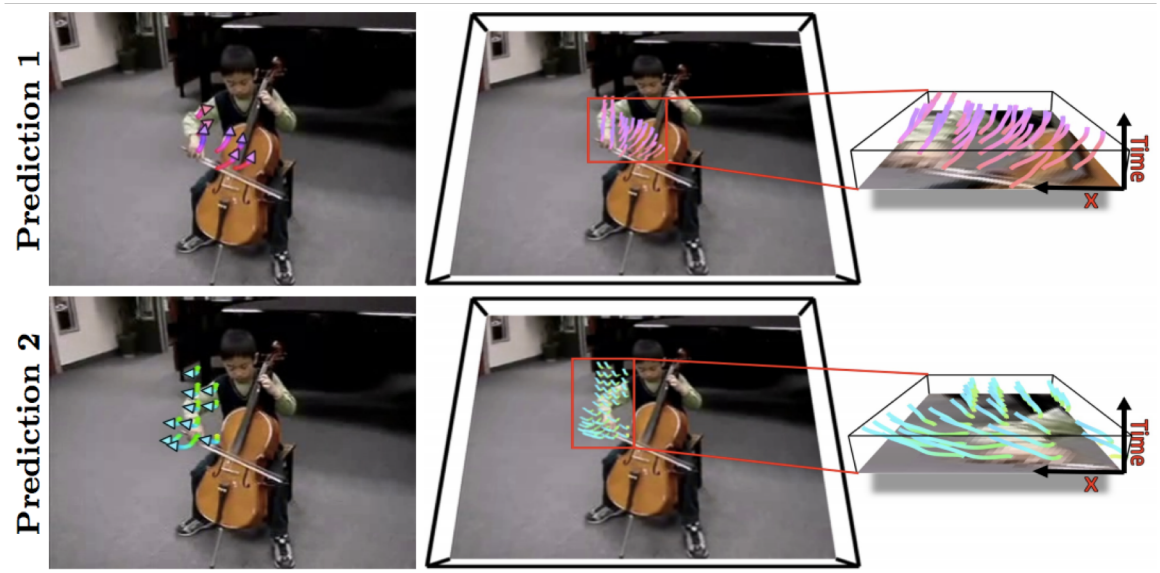
# Video correspondence features



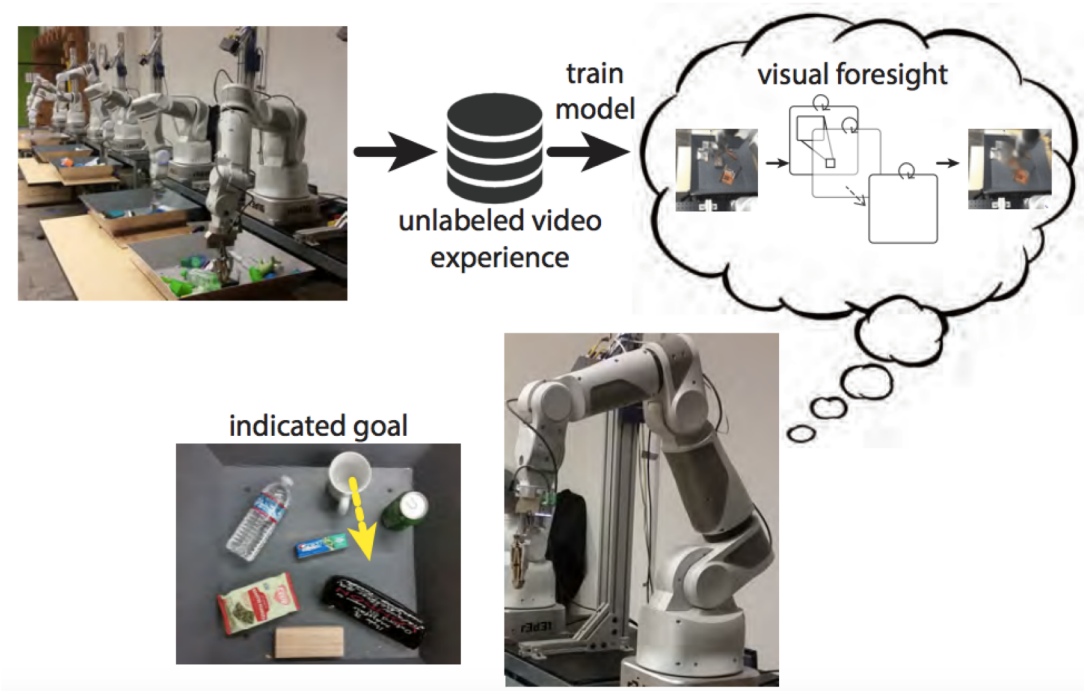

A. Jabri, A. Owens, and A. Efros. Space-time correspondence as a contrastive random walk. NeurIPS 2020

# Future prediction




J. Walker et al. An Uncertain Future: Forecasting from Static Images Using Variational Autoencoders. ECCV 2016

# Future prediction




C. Finn and S. Levine. Deep Visual Foresight for Planning Robot Motion. ICRA 2017

# Self-supervised learning in NLP (coming up)

- word2vec, GloVe, BERT, ELMO, GPT, …

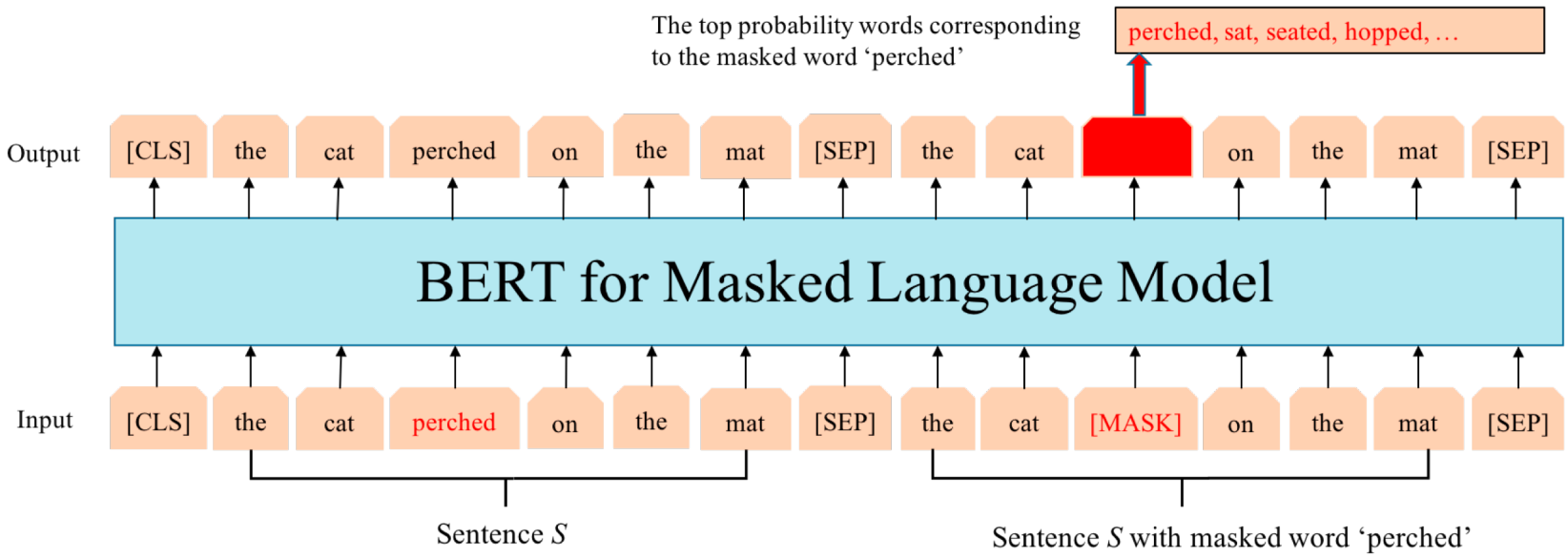



Figure source

# Preview: Self-supervised learning with transformers

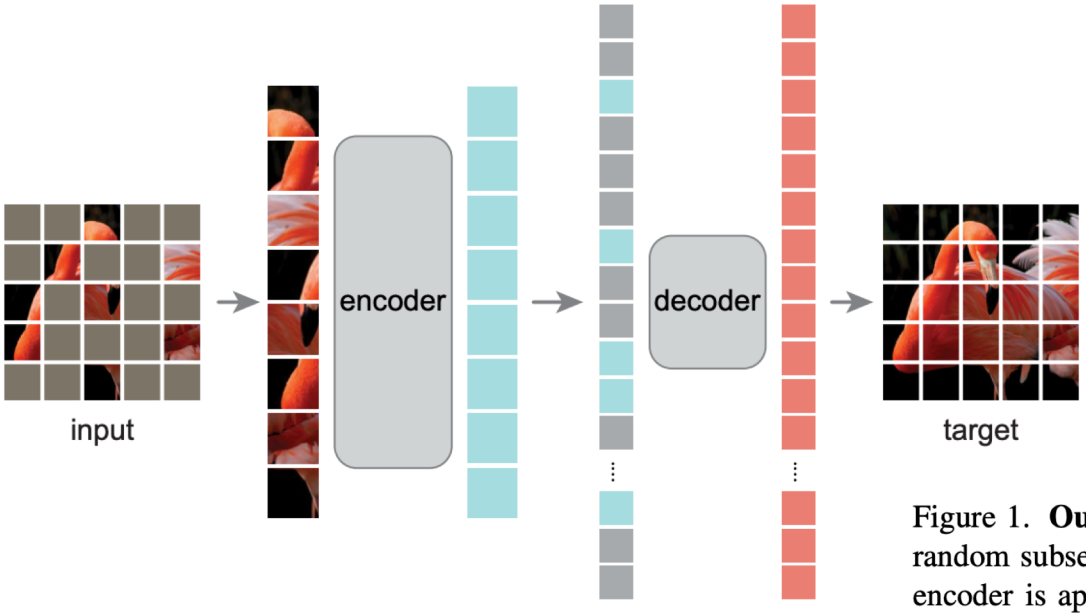



Figure 1. **Our MAE architecture**. During pre-training, a large random subset of image patches (*e.g.*, 75%) is masked out. The encoder is applied to the small subset of *visible patches*. Mask tokens are introduced *after* the encoder, and the full set of encoded patches and mask tokens is processed by a small decoder that reconstructs the original image in pixels. After pre-training, the decoder is discarded and the encoder is applied to uncorrupted images to produce representations for recognition tasks.

K. He et al. Masked autoencoders are scalable vision learners. CVPR 2022

# Preview: Self-supervised learning with transformers

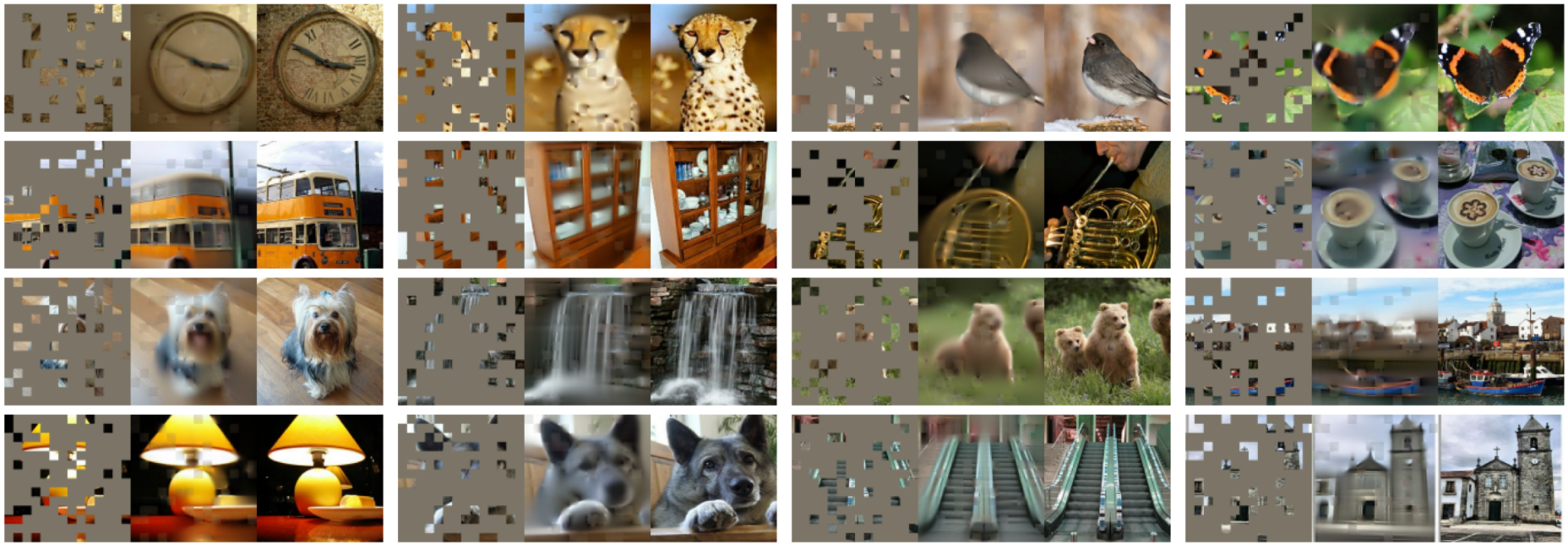


Figure 2. Example results on ImageNet *validation* images. For each triplet, we show the masked image (left), our MAE reconstruction[†] (middle), and the ground-truth (right). The masking ratio is 80%, leaving only 39 out of 196 patches. More examples are in the appendix.
[†]*As no loss is computed on visible patches, the model output on visible patches is qualitatively worse. One can simply overlay the output with the visible patches to improve visual quality. We intentionally opt not to do this, so we can more comprehensively demonstrate the method's behavior.*

K. He et al. Masked autoencoders are scalable vision learners. CVPR 2022

# Preview: Self-supervised learning with transformers

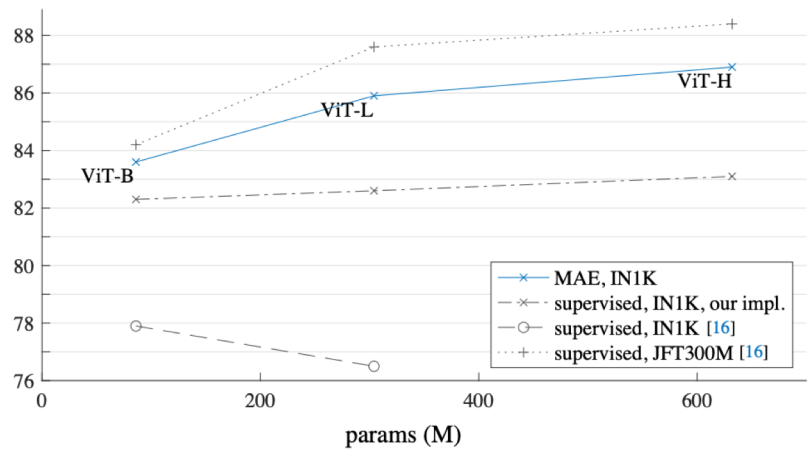



Figure 8. **MAE pre-training *vs*. supervised pre-training**, evaluated by fine-tuning in ImageNet-1K (224 size). We compare with the original ViT results [16] trained in IN1K or JFT300M.

| | | $AP^{box}$ | | $AP^{mask}$ | |
|---|---|---|---|---|---|
| method | pre-train data | ViT-B | ViT-L | ViT-B | ViT-L |
| supervised | IN1K w/ labels | 47.9 | 49.3 | 42.9 | 43.9 |
| MoCo v3 | IN1K | 47.9 | 49.3 | 42.7 | 44.0 |
| BEiT | IN1K+DALLE | 49.8 | **53.3** | 44.4 | 47.1 |
| MAE | IN1K | **50.3** | **53.3** | **44.9** | **47.2** |

Table 4. **COCO object detection and segmentation** using a ViT Mask R-CNN baseline. All entries are based on our implementation. Self-supervised entries use IN1K data *without* labels. Mask AP follows a similar trend as box AP.

K. He et al. Masked autoencoders are scalable vision learners. CVPR 2022

# For further reading

https://github.com/jason718/awesome-self-supervised-learning

https://people.eecs.berkeley.edu/~efros/gelato_bet.html