UCLA

University of Catonsville, Left of Arbutus

Supervised Learning is Expensive ...

CMSC 475/675 Neural Networks

Lecture 9: Self-Supervised Learning

UMBC

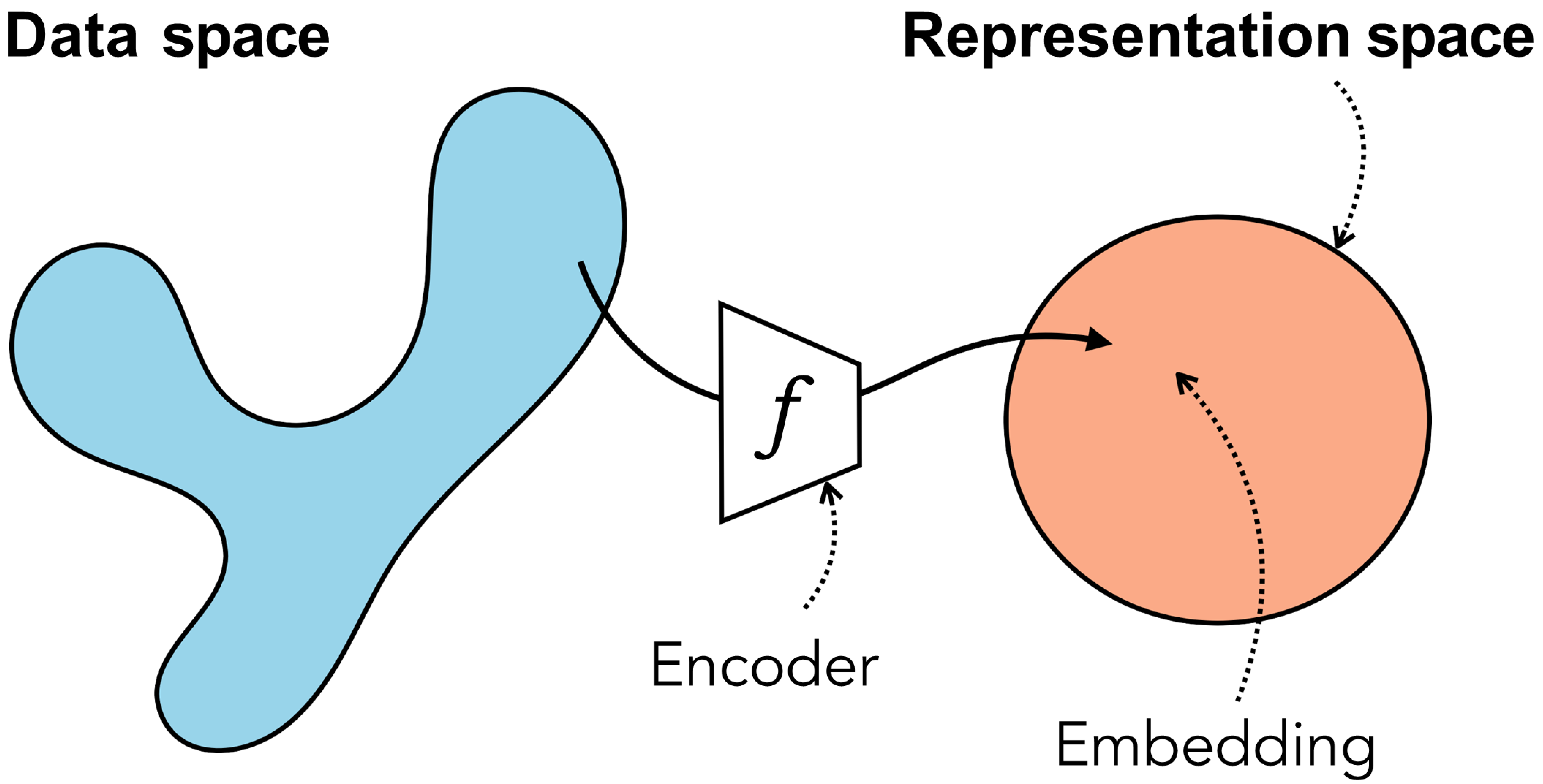
- Train a model on 1 million images ➔ label 1 million images
- Labels aren't magically given to you ➔ need human effort
- How much will it cost?

(1,000,000 images)	(Small to medium sized dataset)
× (10 seconds/image)	(Fast annotation)
× (1/3600 hours/second)	
× ($15 / hour)	(Minimum wage)
× (3 annotators / image)	(for consensus / removing noise)

= ~ $125k without considering overhead / admin costs …

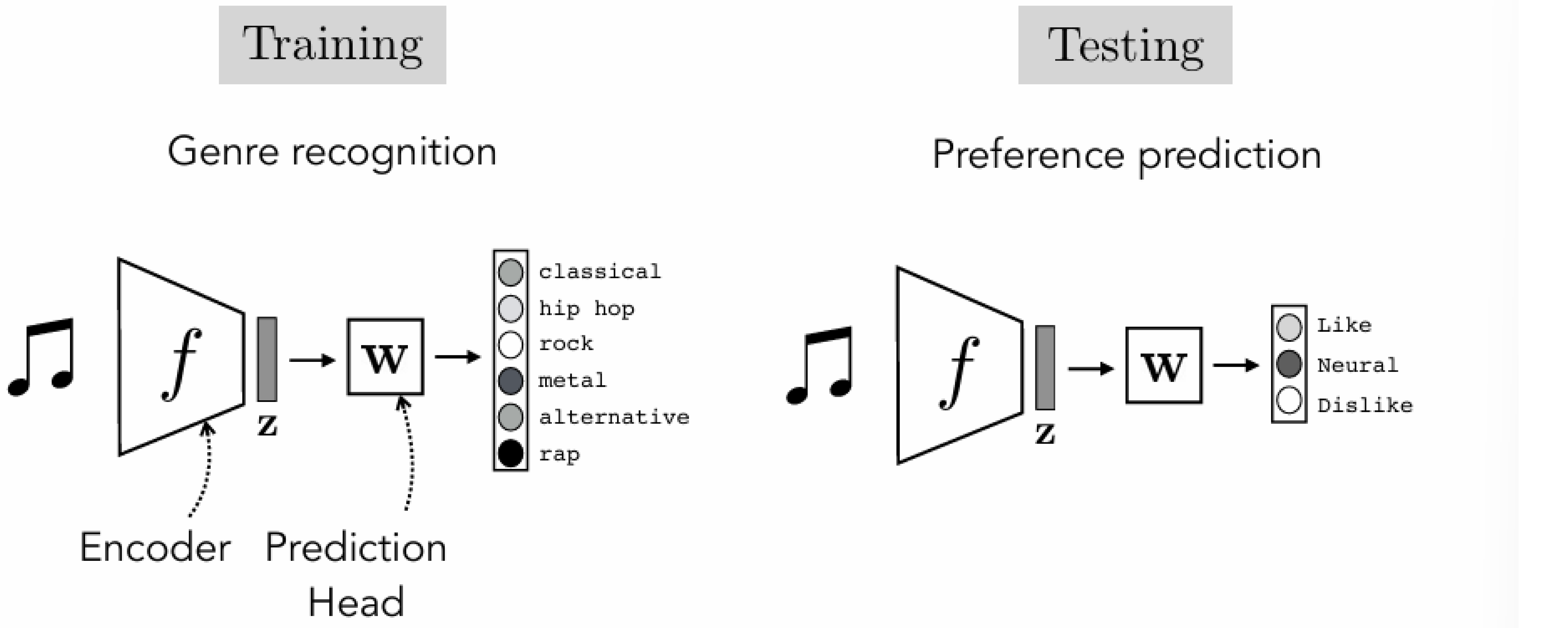
Recap: Representation Learning "x2vec"

- A representation of a data domain $\mathcal{X}$ is a function $f : \mathcal{X} \to \mathbb{R}^d$ (an encoder) that assigns a feature vector to each input in that domain.
- A representation of a datapoint is a vector $z \in \mathbb{R}^d$ with $z = f(x)$.



Why Learn Representations?

"Generally speaking, a good representation is one that makes a subsequent learning task easier."
- Goodfellow et al. "Deep Learning". 2016



Often, what we will be "tested" on is not what we were trained on.

Why Learn Representations?

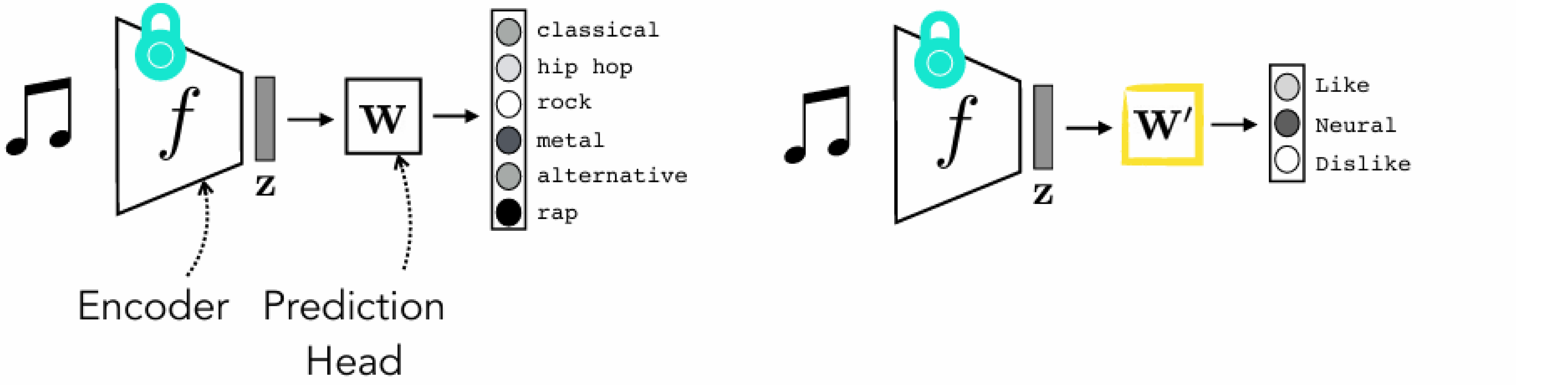
"Generally speaking, a good representation is one that makes a subsequent learning task easier."
- Goodfellow et al. "Deep Learning". 2016

Training

Genre recognition

Adapting

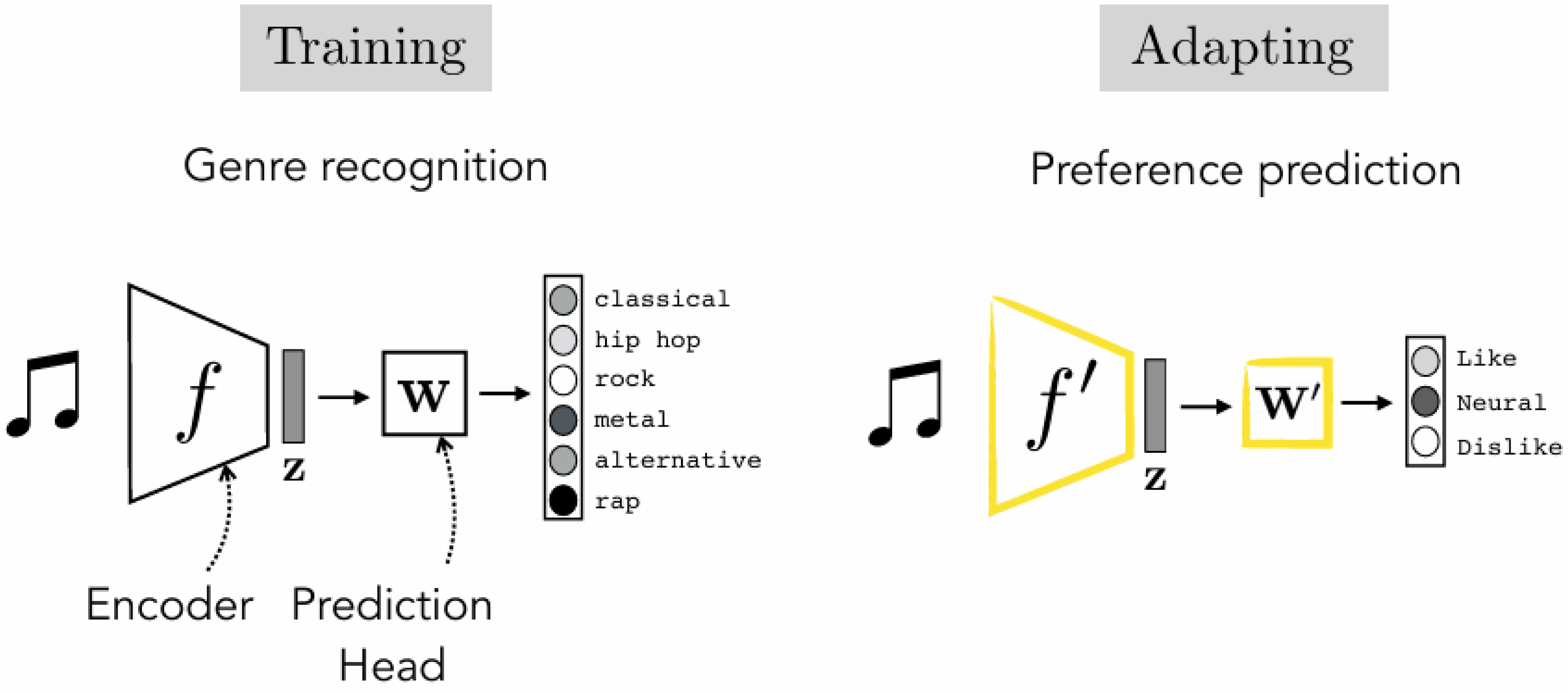
Preference prediction



Linear adaptation: freeze f, train a new linear map to new target data

Why Learn Representations?

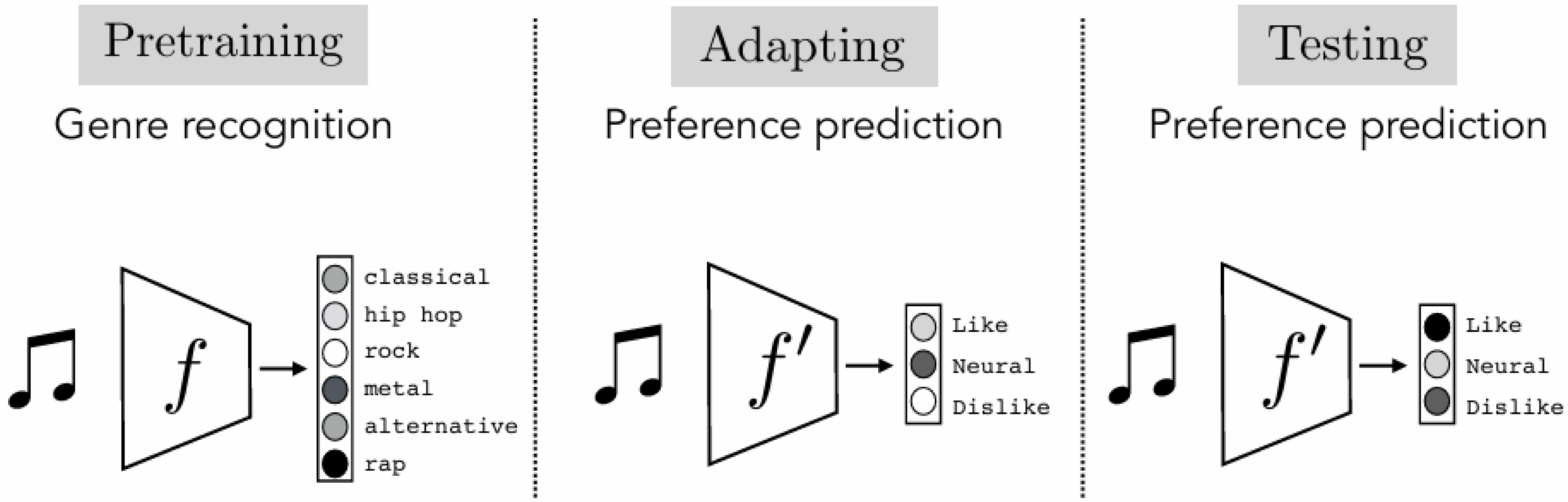
"Generally speaking, a good representation is one that makes a subsequent learning task easier."
- Goodfellow et al. "Deep Learning". 2016



Finetuning: initialize f′ as f, then continue training on new target data

Why Learn Representations?

"Generally speaking, a good representation is one that makes a subsequent learning task easier."
- Goodfellow et al. "Deep Learning". 2016



A lot of data

A little data

Learning from examples

(aka **supervised learning**)

Training data

$$\{x^{(1)}, y^{(1)}\}$$
$$\{x^{(2)}, y^{(2)}\} \quad \rightarrow \quad \boxed{\text{Learner}} \quad \rightarrow \quad f : X \rightarrow Y$$
$$\{x^{(3)}, y^{(3)}\}$$
$$\ldots$$

$$f^* = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{N} \mathcal{L}(f(\mathbf{x}^{(i)}), \mathbf{y}^{(i)})$$

Learning without examples

(includes **unsupervised learning** / **self-supervised learning**)

Data

$\{x^{(1)}\}$

$\{x^{(2)}\}$ $\rightarrow$ Learner $\rightarrow$?

$\{x^{(3)}\}$

$\cdots$

Learning without examples

(includes **unsupervised learning** / **self-supervised learning**)

Data

$\{x^{(1)}\}$

$\{x^{(2)}\}$

$\{x^{(3)}\}$

$\ldots$

$\rightarrow$ Learner $\rightarrow$

Embeddings

Clusters

Metrics

...

Two Basic Approaches:
(1) Compression (2) Prediction

Learning Method	Learning Principle	Short Summary
Autoencoding	Compression	Remove redundant information
Contrastive	Compression	Achieve invariance to viewing transformations
Clustering	Compression	Quantize continuous data into discrete categories
Future prediction	Prediction	Predict the future
Imputation	Prediction	Predict missing data
Pretext tasks	Prediction	Predict abstract properties of your data

Some examples of the “Compression” Approach:

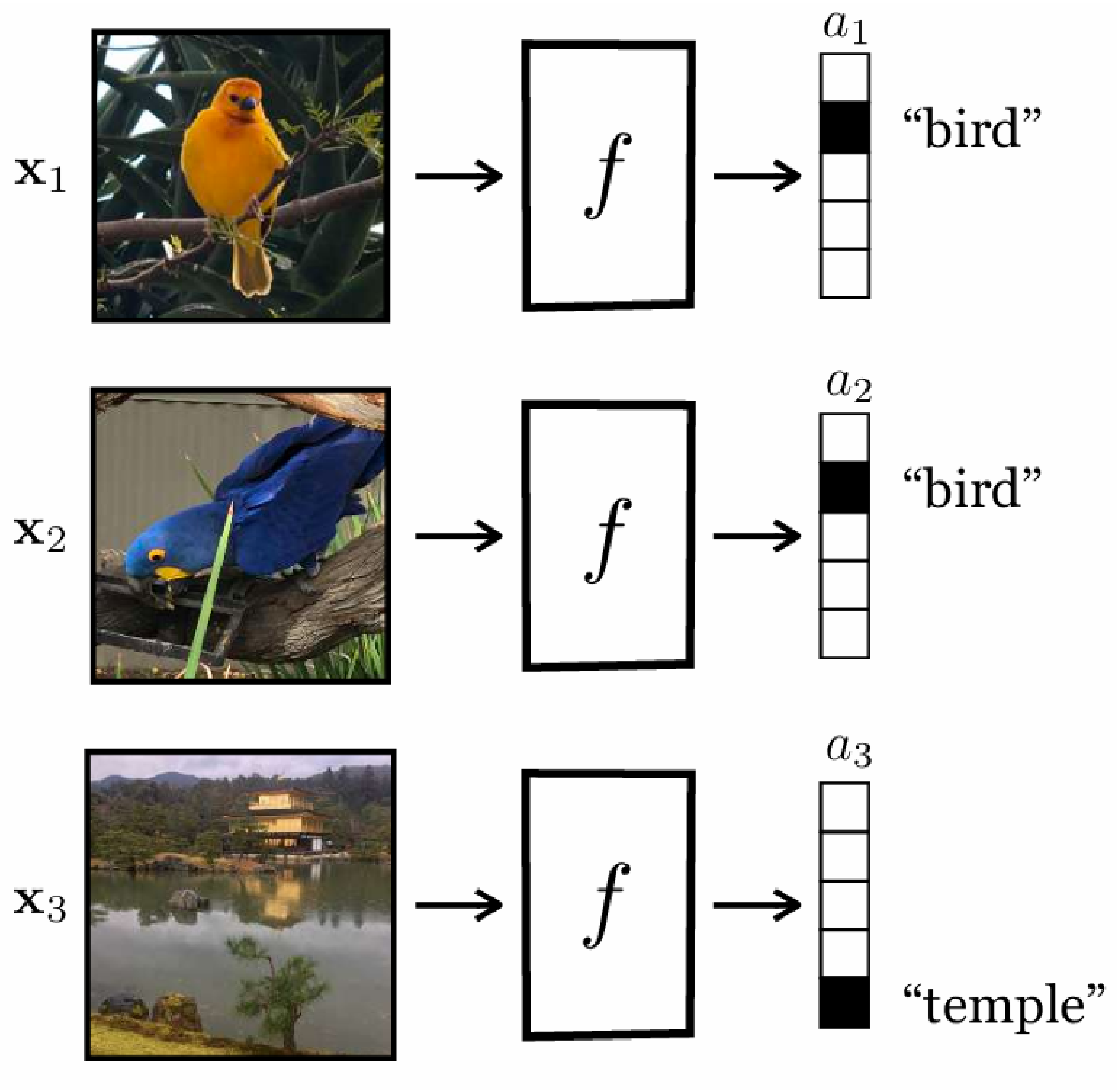
Recap: Autoencoder

$$f^*, g^* = \arg\min_{f,g} \mathbb{E}_{\mathbf{x}} \left\| \mathbf{x} - g(f(\mathbf{x})) \right\|_2^2$$

Clustering

Training
Data
Learner
Encoder
$f : \mathcal{X} \to \{1, \ldots, k\}$
Inference
Data
$f : \mathcal{X} \to \{1, \ldots, k\}$
Clusters

Clustering

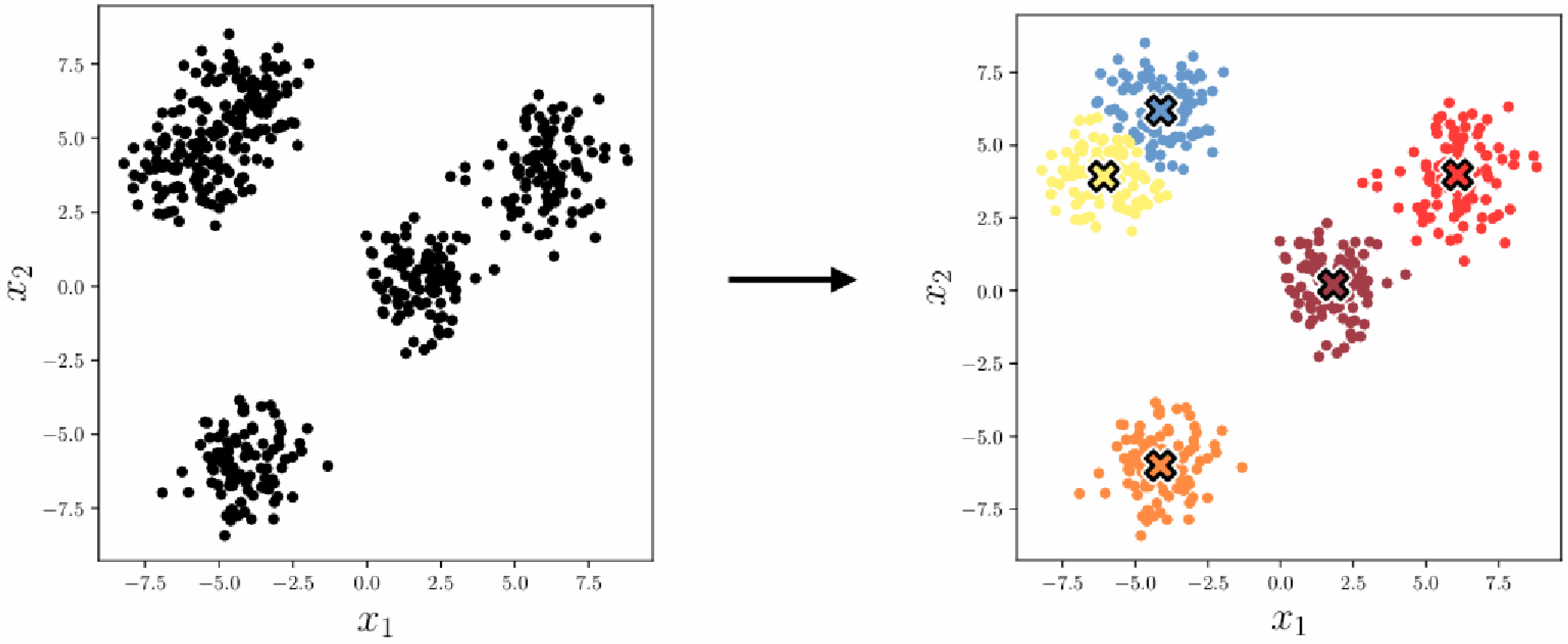


- What's the best representation that humans have come up with so far?
- Language!
- Words are the atoms of language
- Clustering is the problem of making up new words for things

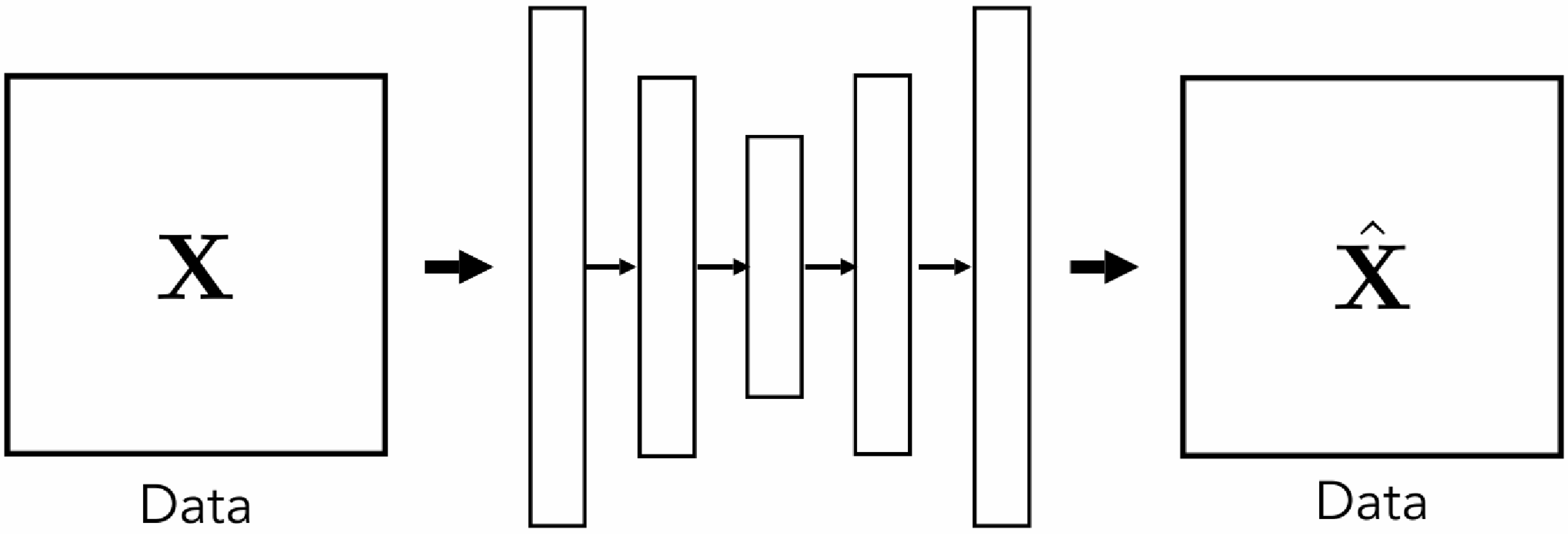
Clustering Algorithm: k-means

- Map datapoints to integers (i.e. cluster)

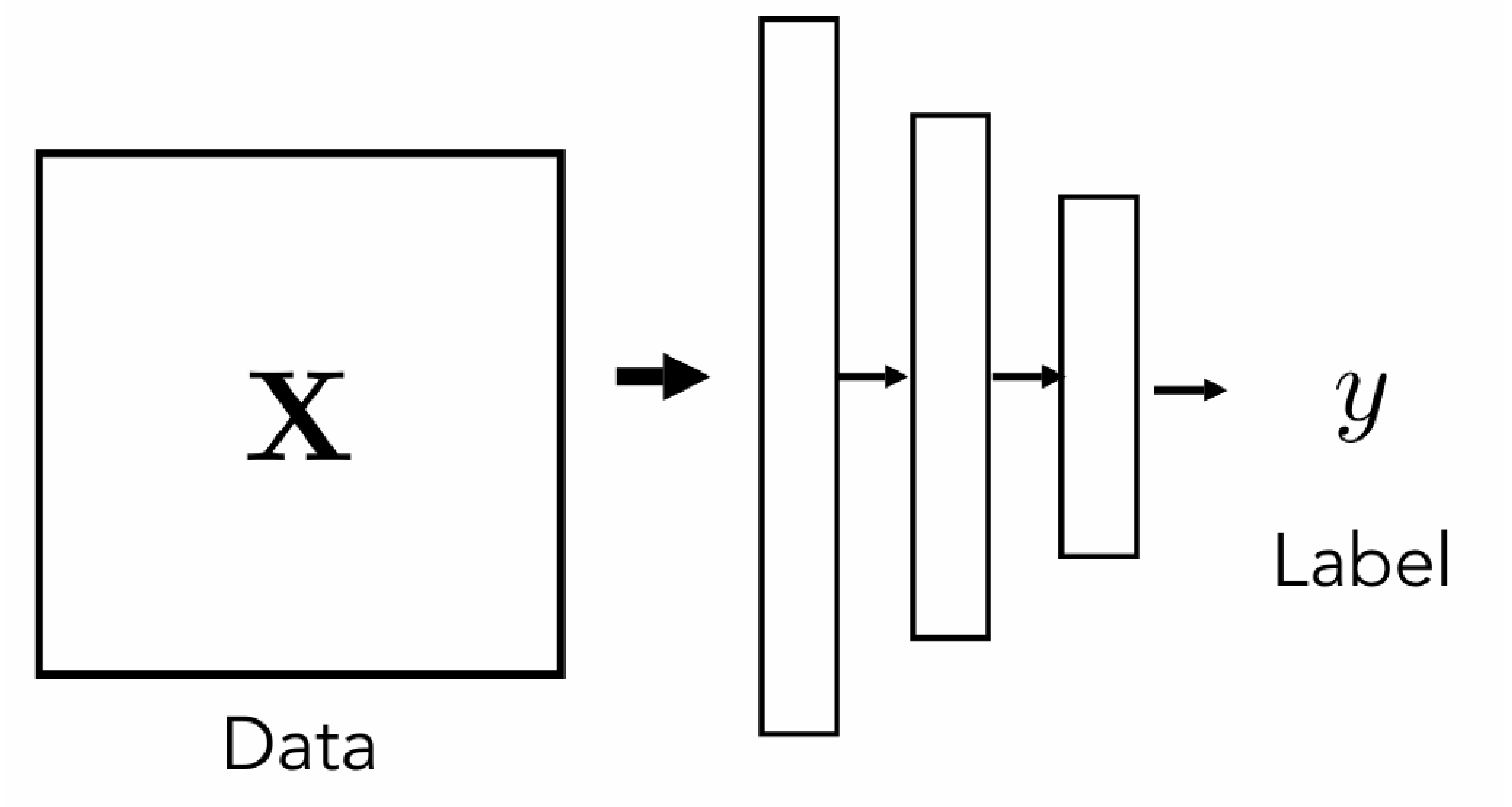
- In such a way that each datapoint is as close as possible to the mean of the cluster it is assigned to



The "Compression" Approach



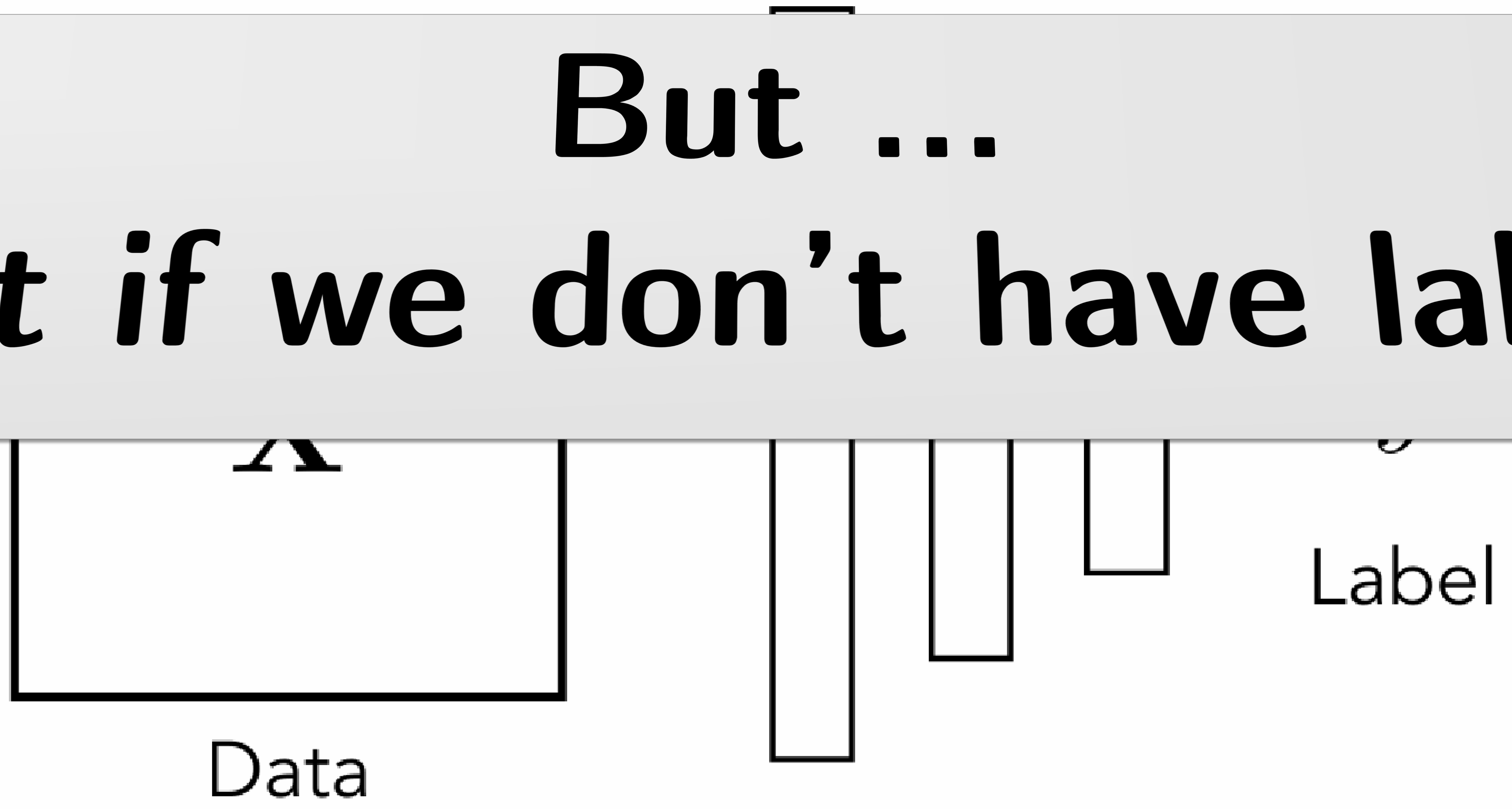
The "Prediction" Approach for Representation Learning



The "Prediction" Approach for Representation Learning

But ...

***what if* we don't have labels?**



Data prediction

aka "self-supervised learning"

Self-Supervised Learning

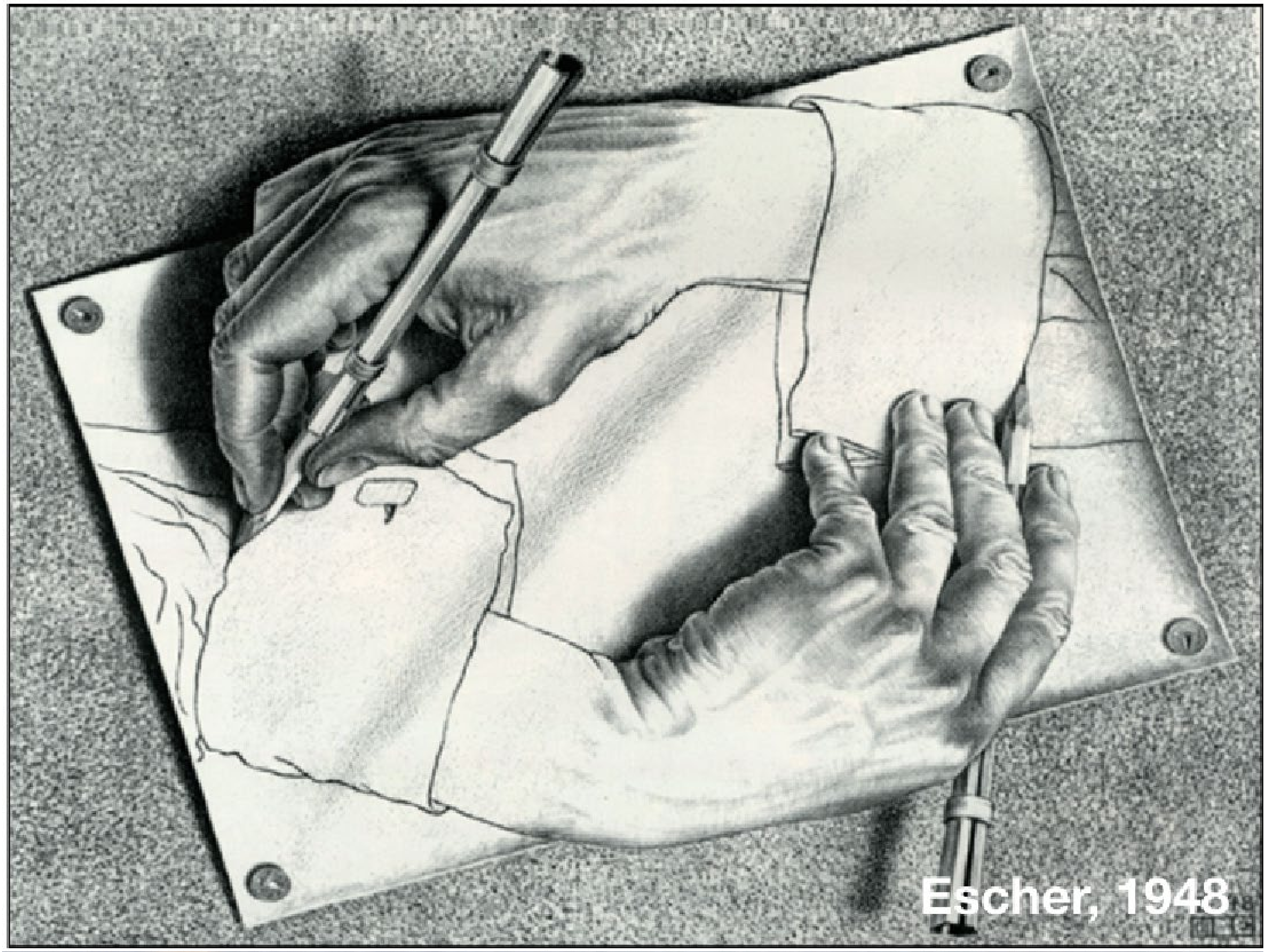
Build methods that learn from "raw" data (inputs only) — no labels!

- **Unsupervised Learning:**
  - older terminology … model isn't told what to predict
- **Self-Supervised Learning:**
  - model is trained to predict *some natural occurring signal* rather than predicting labels
- **Semi-Supervised Learning:**
  - train jointly with some labeled data and a lot of unlabeled data.

Self-Supervised Learning: A trick

- If you don't have labels, make labels.
- Convert "unsupervised" problem into "supervised"
- Cook up labels (prediction targets) from the data itself
    - This is often called a "pretext" task

Claim:
Training a model for "pretext" task can lead to very good representations



SSL: "Pretext then transfer"

Step 1: Pretrain a network on a pretext task that doesn't require supervision

Input Image: x

Encoder: ϕ

Features: $\phi \quad x$

Decoder: ψ

Prediction: y

Loss: $L(y, x)$

Step 2: Transfer encoder to downstream tasks via linear classifiers, KNN, finetuning

Input Image: x

Encoder: ϕ

Features: $\phi \quad x$

Downstream tasks: Image classification, object detection, semantic segmentation

Some Examples of Pretext Tasks

Examples of Pretext Tasks

Generative:

Predict part of the input signal

- Autoencoders (sparse, denoising, masked)
- Autoregressive
- GANs
- Colorization
- Inpainting

Discriminative:

Predict something about the input signal

- Context prediction
- Rotation
- Clustering
- Contrastive

Multimodal:

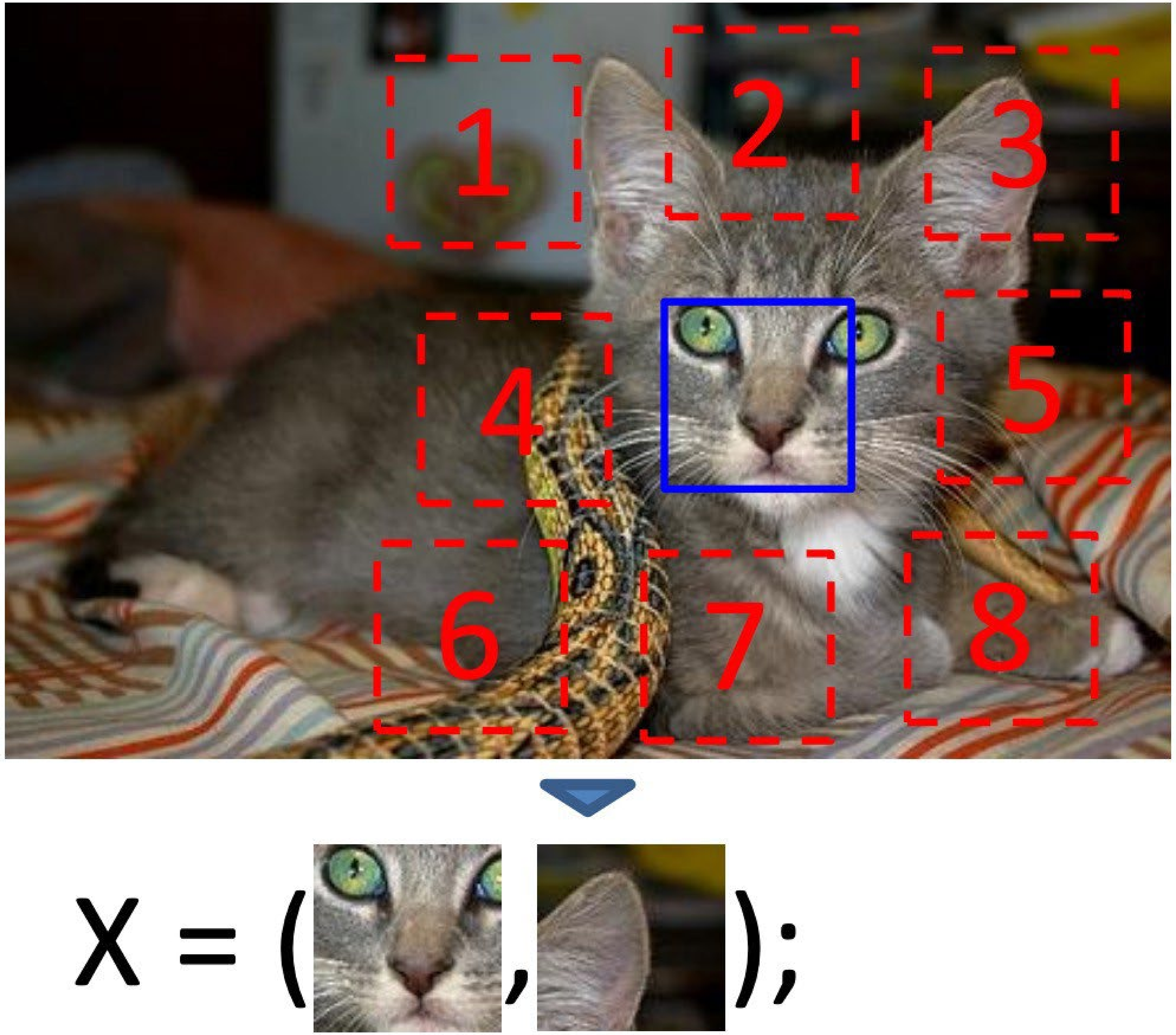
Use some signal in addition to RGB images

- Video
- 3D
- Sound
- Language

Context Prediction

Model predicts relative location of two patches from the same image.
<u>Discriminative</u> pretraining task

Intuition: Requires understanding objects and their parts

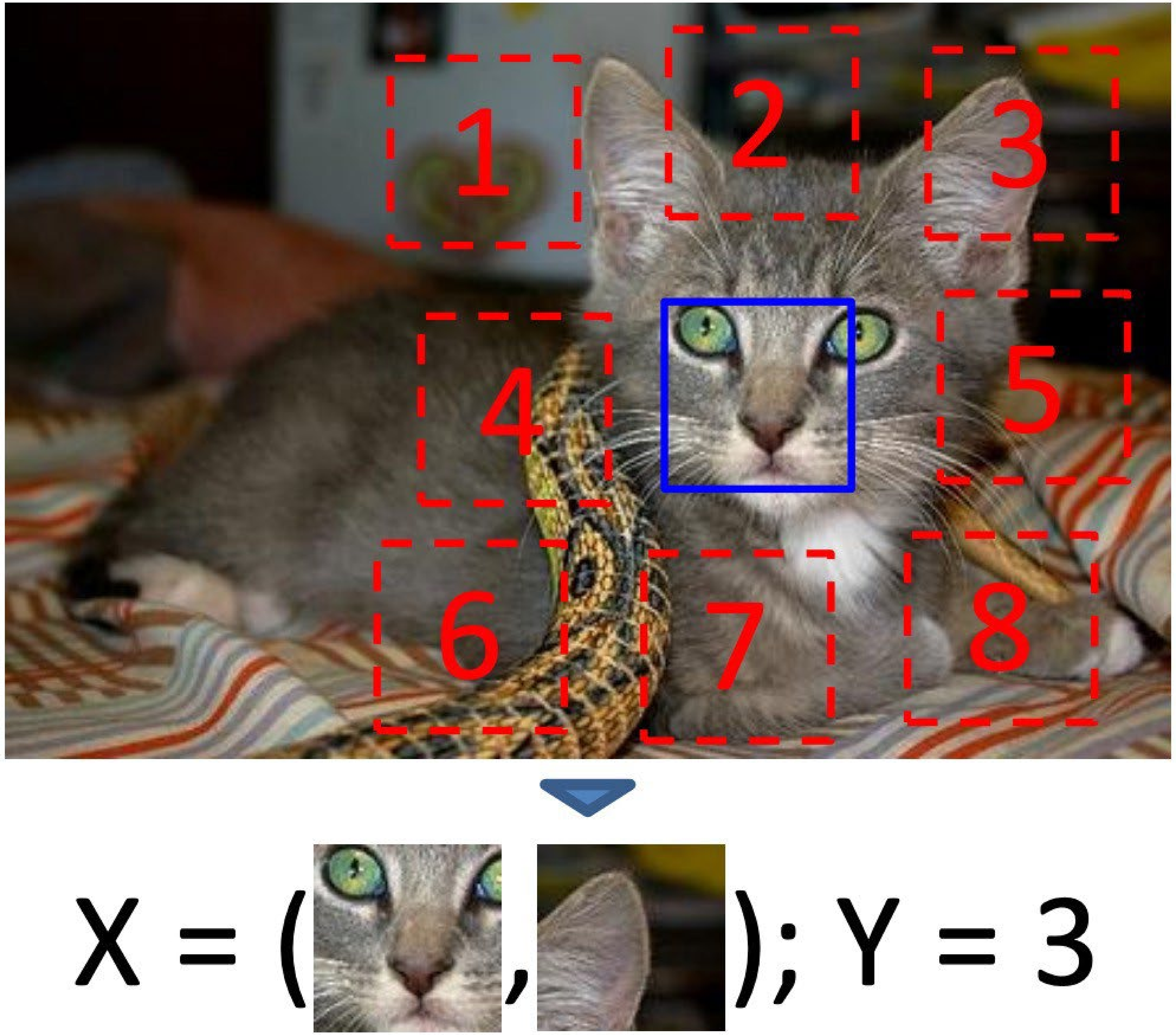


Doersch et al, “Unsupervised Visual Representation Learning by Context Prediction”, ICCV 2015

Context Prediction

Model predicts relative location of two patches from the same image.
<u>Discriminative</u> pretraining task

Intuition: Requires understanding objects and their parts

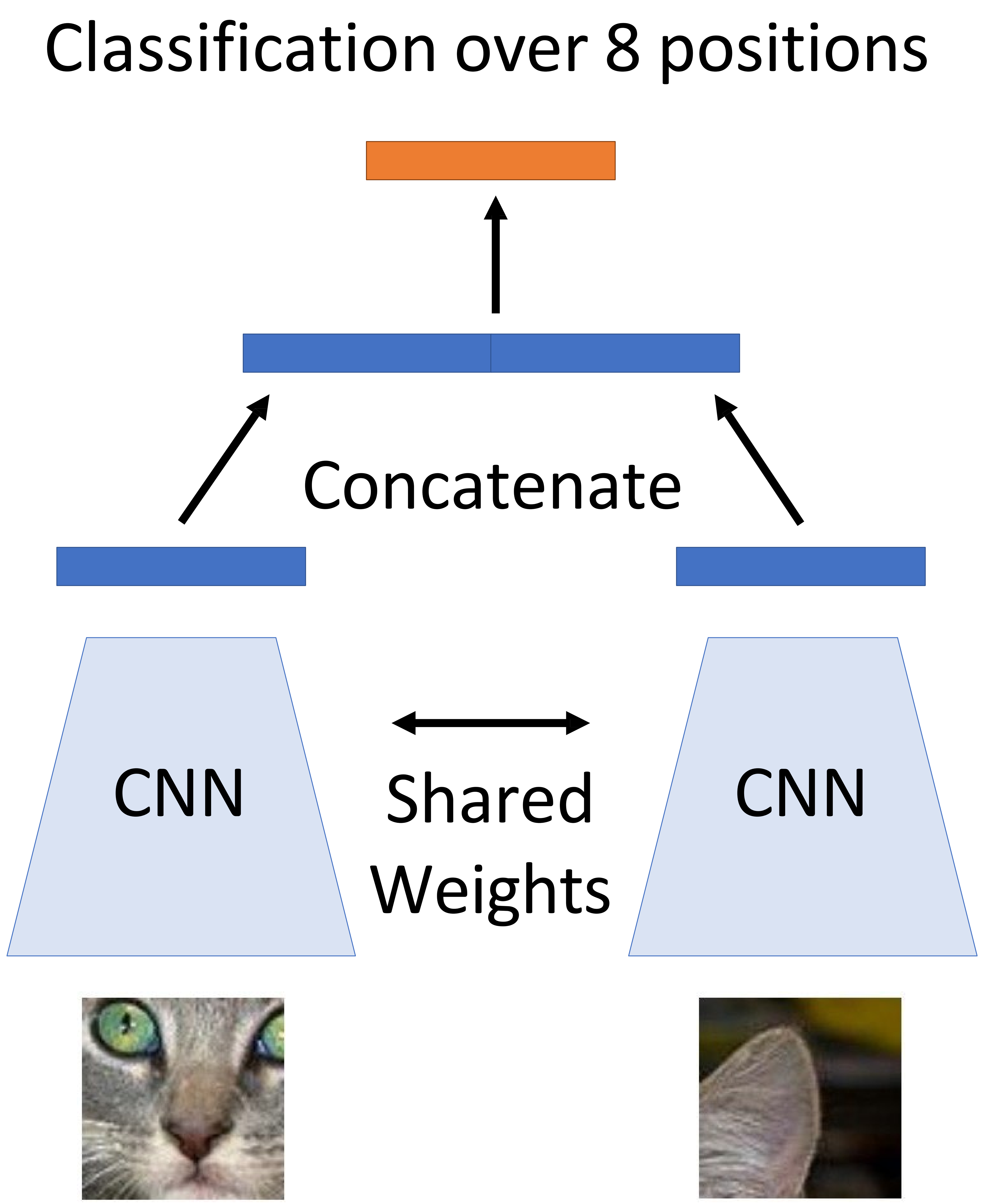


Doersch et al, “Unsupervised Visual Representation Learning by Context Prediction”, ICCV 2015

Context Prediction

Model predicts relative location of two patches from the same image.
Discriminative pretraining task

Intuition: Requires understanding objects and their parts



Doersch et al, "Unsupervised Visual Representation Learning by Context Prediction", ICCV 2015

Extension: Solving Jigsaw Puzzles

Rather than predict relative position of two patches, instead predict permutation to “unscramble” 9 shuffled patches

Noroozi and Favoro, “Unsupervised learning of visual representations by solving jigsaw puzzles”, ECCV 2016

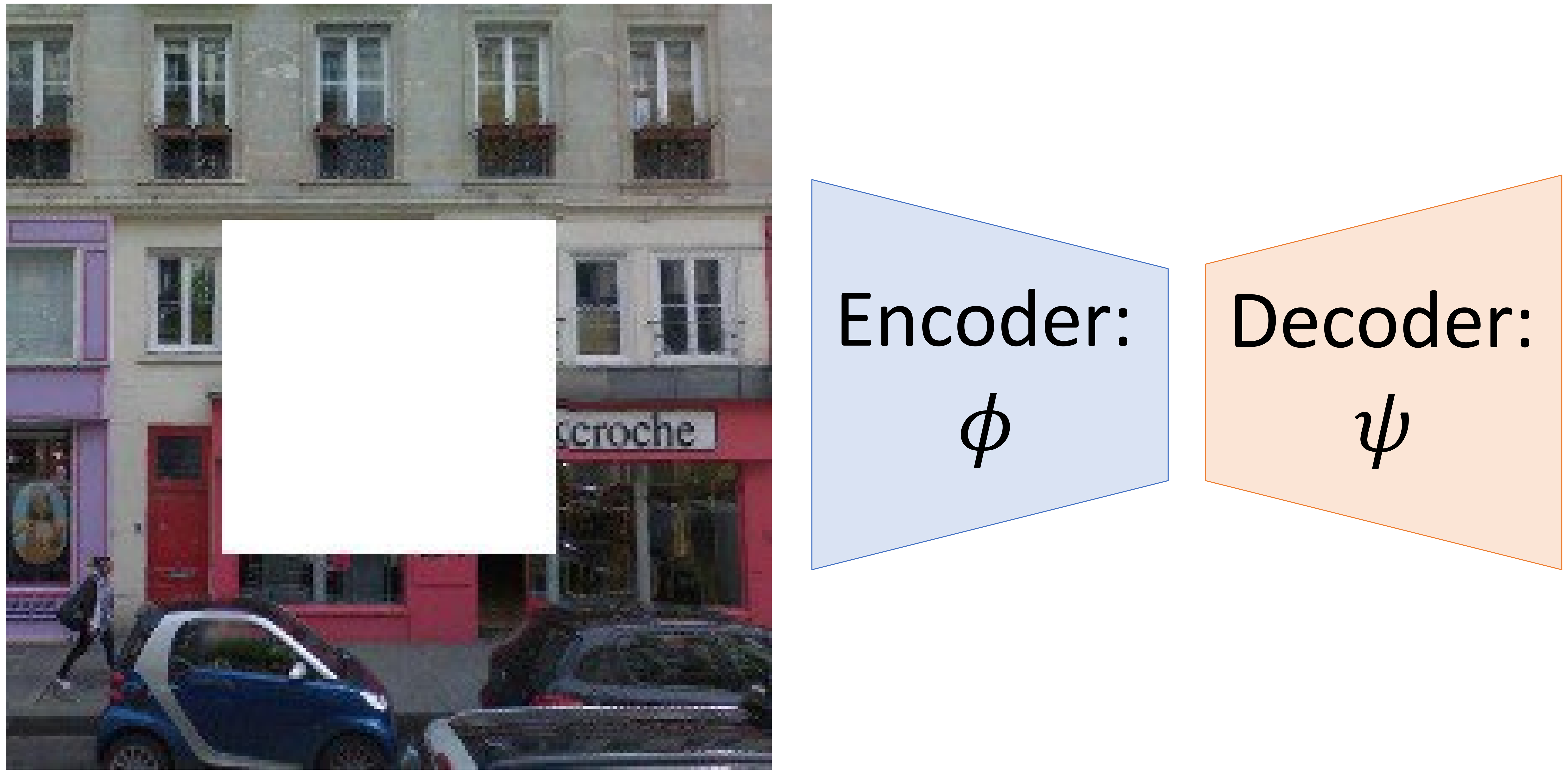
Extension: Solving Jigsaw Puzzles

Rather than predict relative position of two patches, instead predict permutation to “unscramble” 9 shuffled patches

Noroozi and Favoro, “Unsupervised learning of visual representations by solving jigsaw puzzles”, ECCV 2016

Context Encoders: Learning by Inpainting

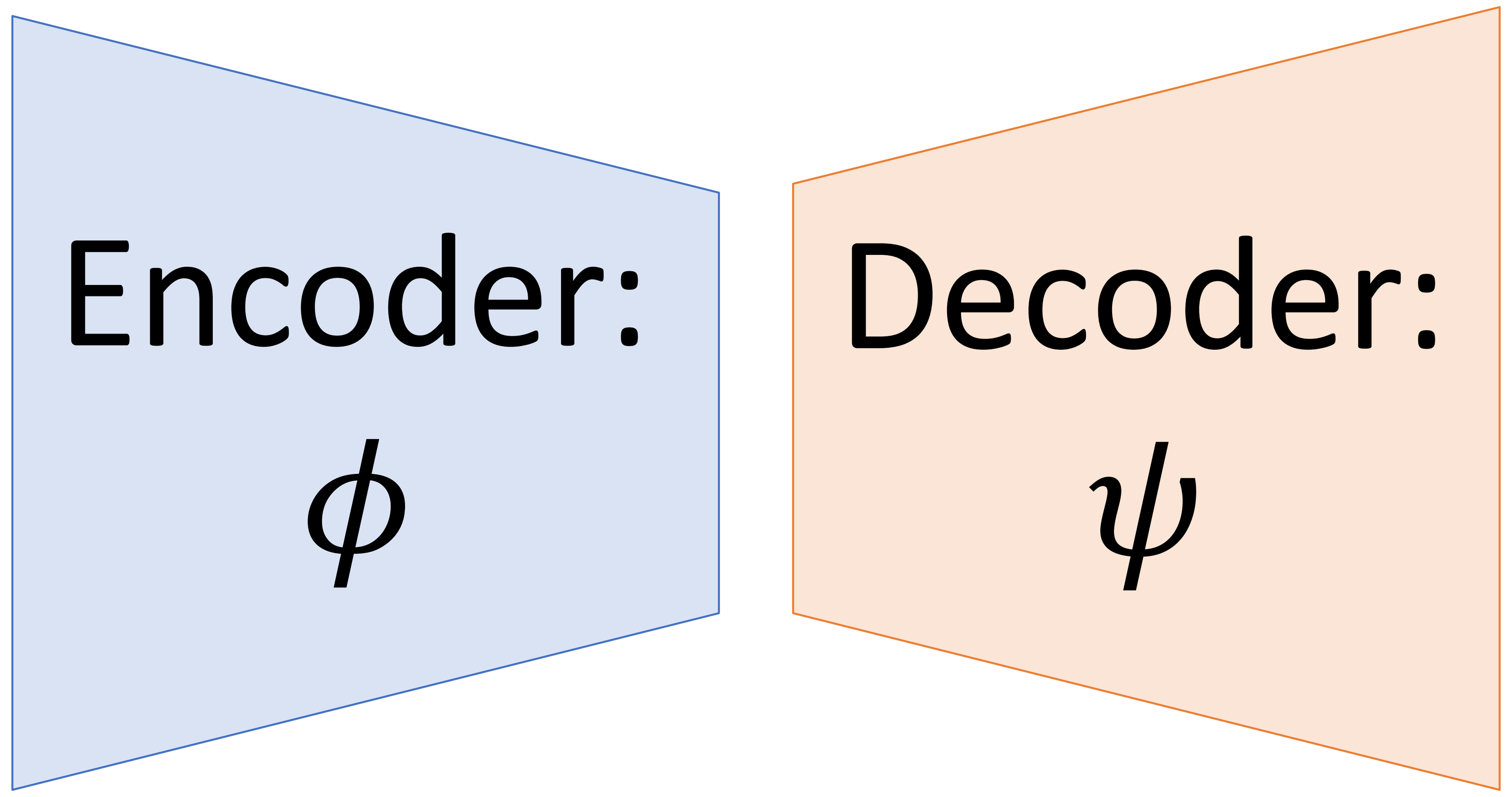
Input Image



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

Context Encoders: Learning by Inpainting

Input Image



Predict Missing Pixels

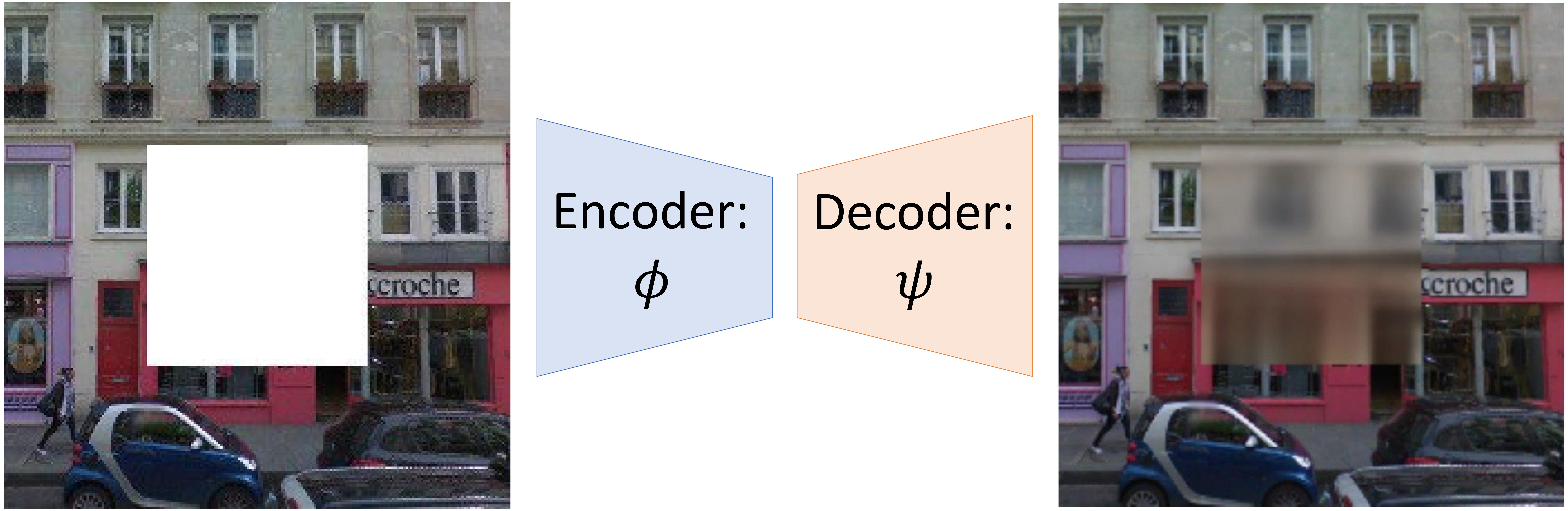
Human Artist

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

Context Encoders: Learning by Inpainting

Input Image

Predict Missing Pixels



L2 Loss

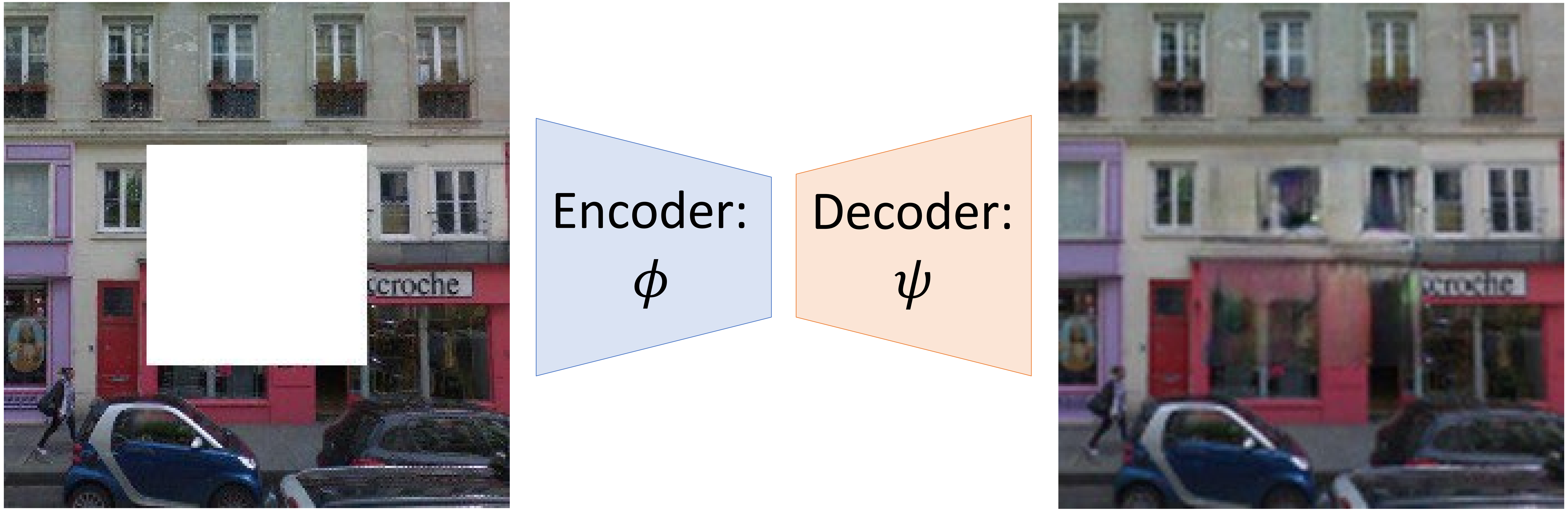
(Best for feature learning)

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

Context Encoders: Learning by Inpainting

Input Image

Predict Missing Pixels



L2 + Adversarial Loss
(Best for nice images)

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

Intuition: A model must be able to identify objects to be able to colorize them

Input: Grayscale Image

Output: Color Image

Zhang et al, “Colorful Image Colorization”, ECCV 2016

Colorization

Zhang et al, "Colorful Image Colorization", ECCV 2016

Pretext task: video coloring

Idea: model the temporal coherence of colors in videos

reference frame

t = 0

how should I color these frames?

t = 1

t = 2

t = 3

...

Source: [Vondrick et al., 2018]

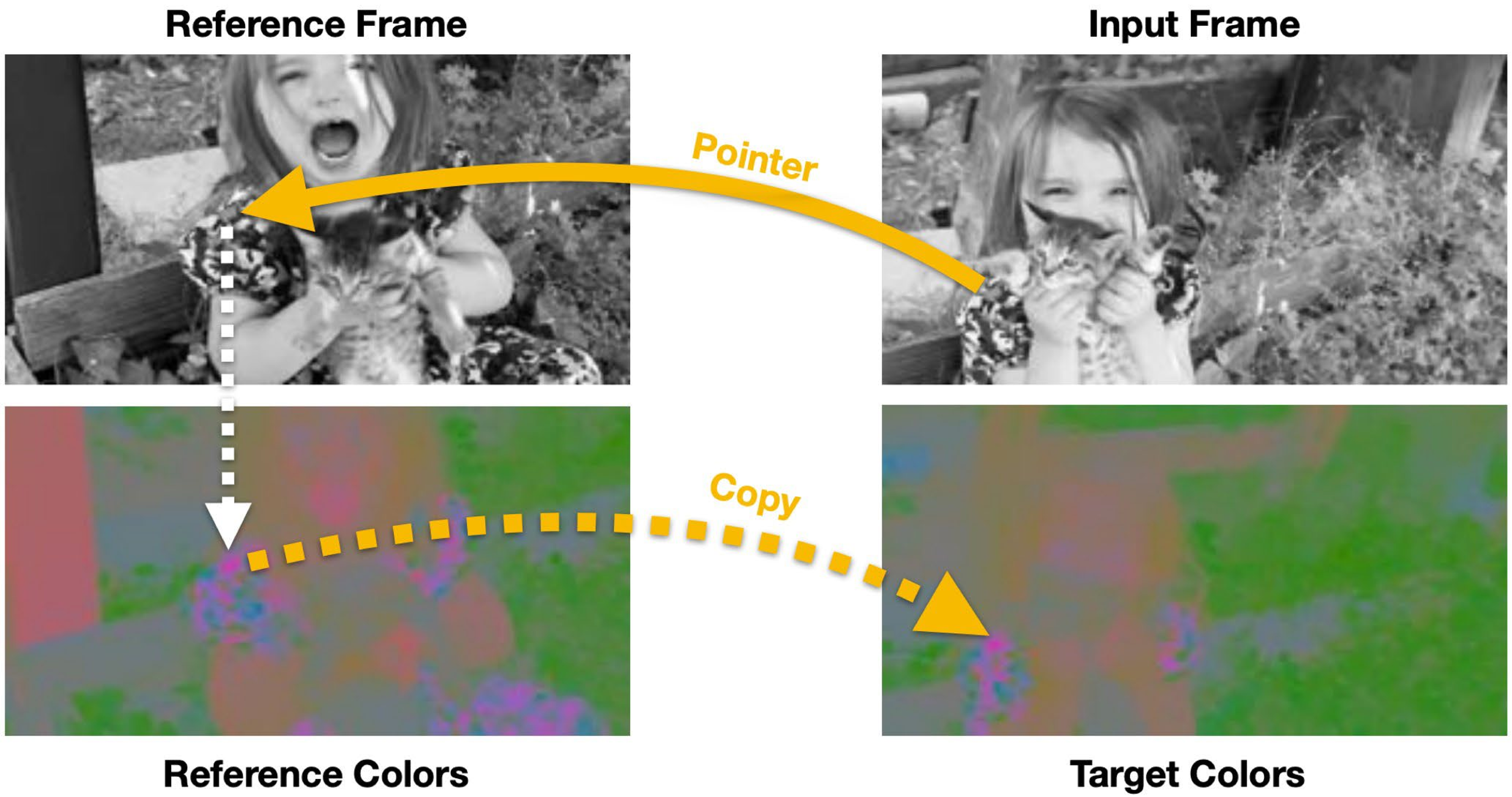
Pretext task: video coloring

Idea: model the temporal coherence of colors in videos

Hypothesis: learning to color video frames should allow model to learn to track regions or objects without labels!

Source: [Vondrick et al., 2018]

Learning to color videos



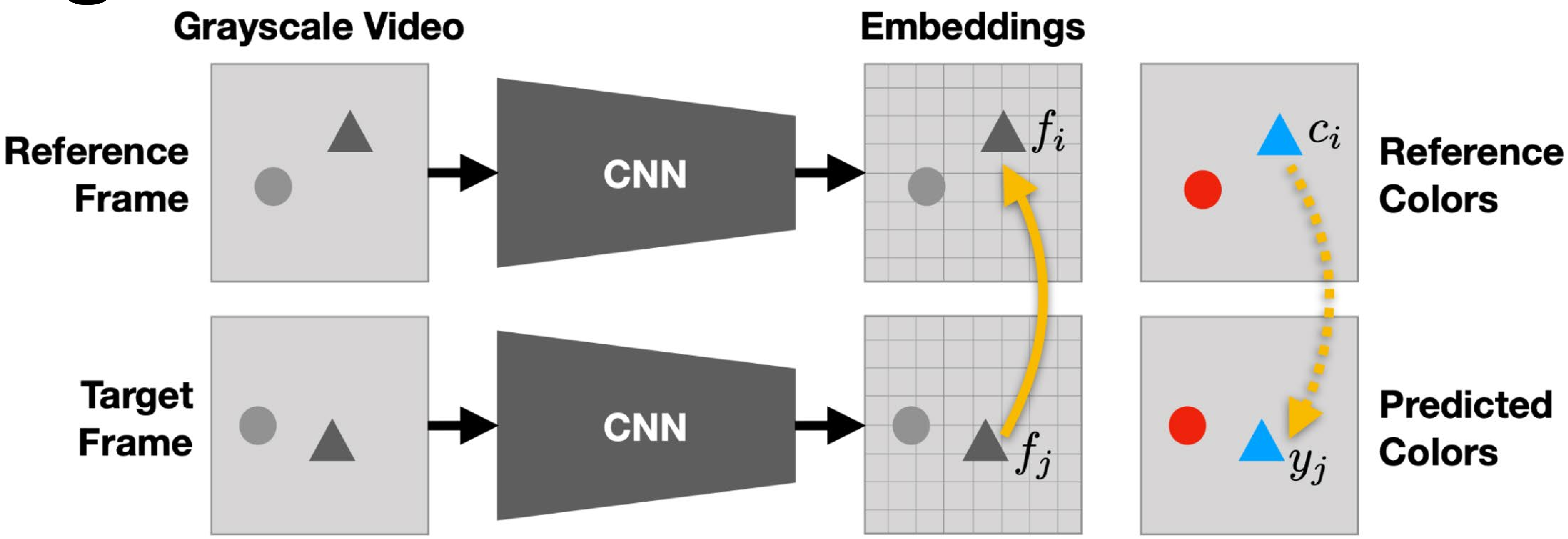
Learning objective:

Establish mappings between reference and target frames in a learned feature space.

Use the mapping as "pointers" to copy the correct color (LAB).

Source: Vondrick et al., 2018

Learning to color videos

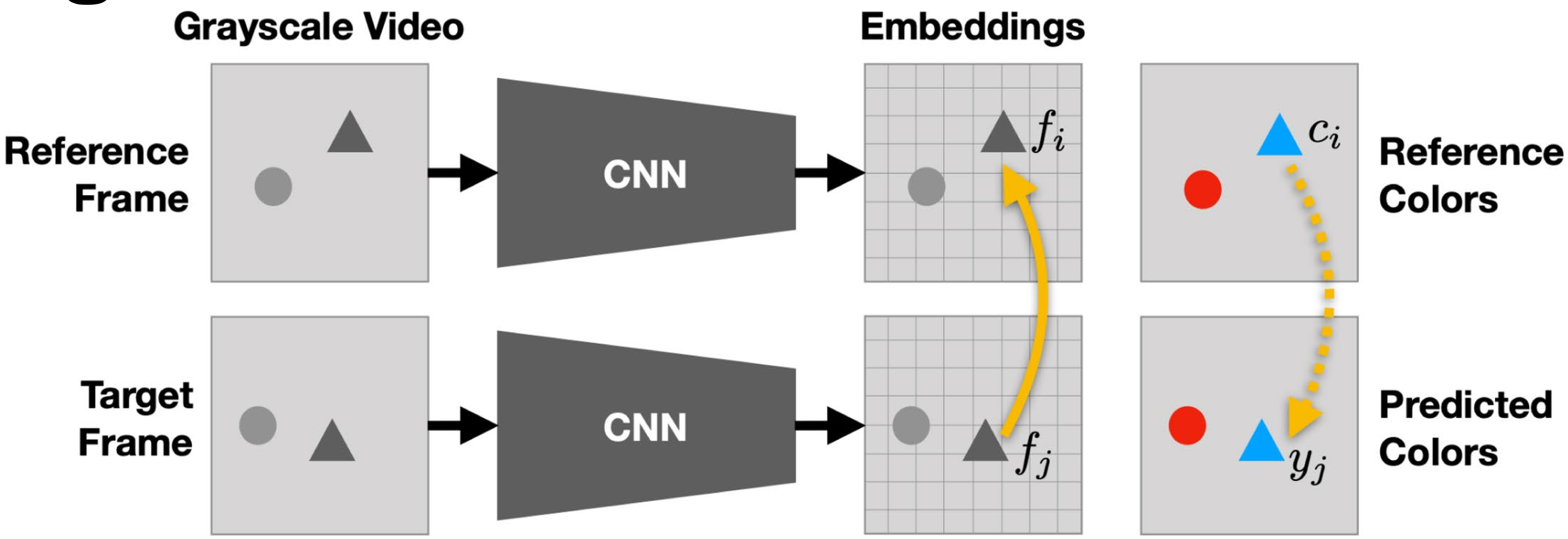


attention map on the reference frame

$$A_{ij} = \frac{\exp\left(f_i^T f_j\right)}{\sum_k \exp\left(f_k^T f_j\right)}$$

Source: Vondrick et al., 2018

Learning to color videos



attention map on the reference frame

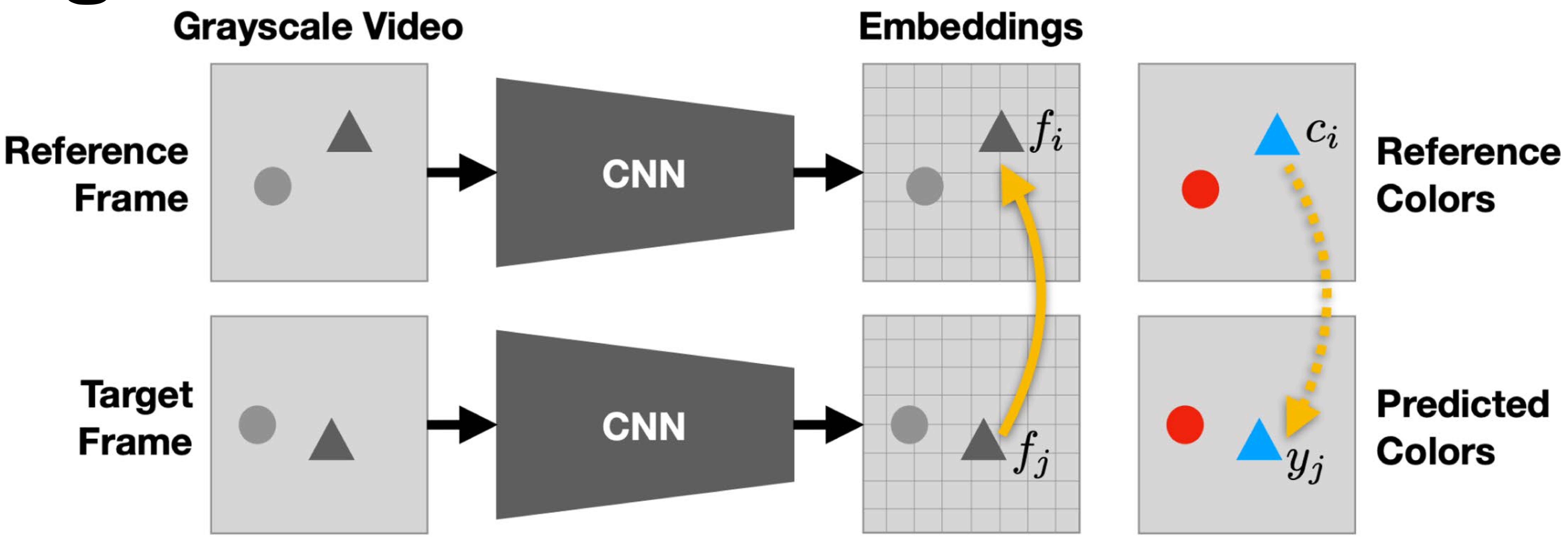
$$A_{ij} = \frac{\exp\left(f_i^T f_j\right)}{\sum_k \exp\left(f_k^T f_j\right)}$$

predicted color = weighted sum of the reference color

$$y_j = \sum_i A_{ij} c_i$$

Source: [Vondrick et al., 2018]

Learning to color videos



attention map on the reference frame

$$A_{ij} = \frac{\exp\left(f_i^T f_j\right)}{\sum_k \exp\left(f_k^T f_j\right)}$$

predicted color = weighted sum of the reference color

$$y_j = \sum_i A_{ij} c_i$$

loss between predicted color and ground truth color

$$\min_\theta \sum_j \mathcal{L}\left(y_j, c_j\right)$$

Source: [Vondrick et al., 2018]

Colorizing videos (qualitative)

reference frame

target frames (gray)

predicted color

Source: [Google AI blog post]

Colorizing videos (qualitative)

reference frame

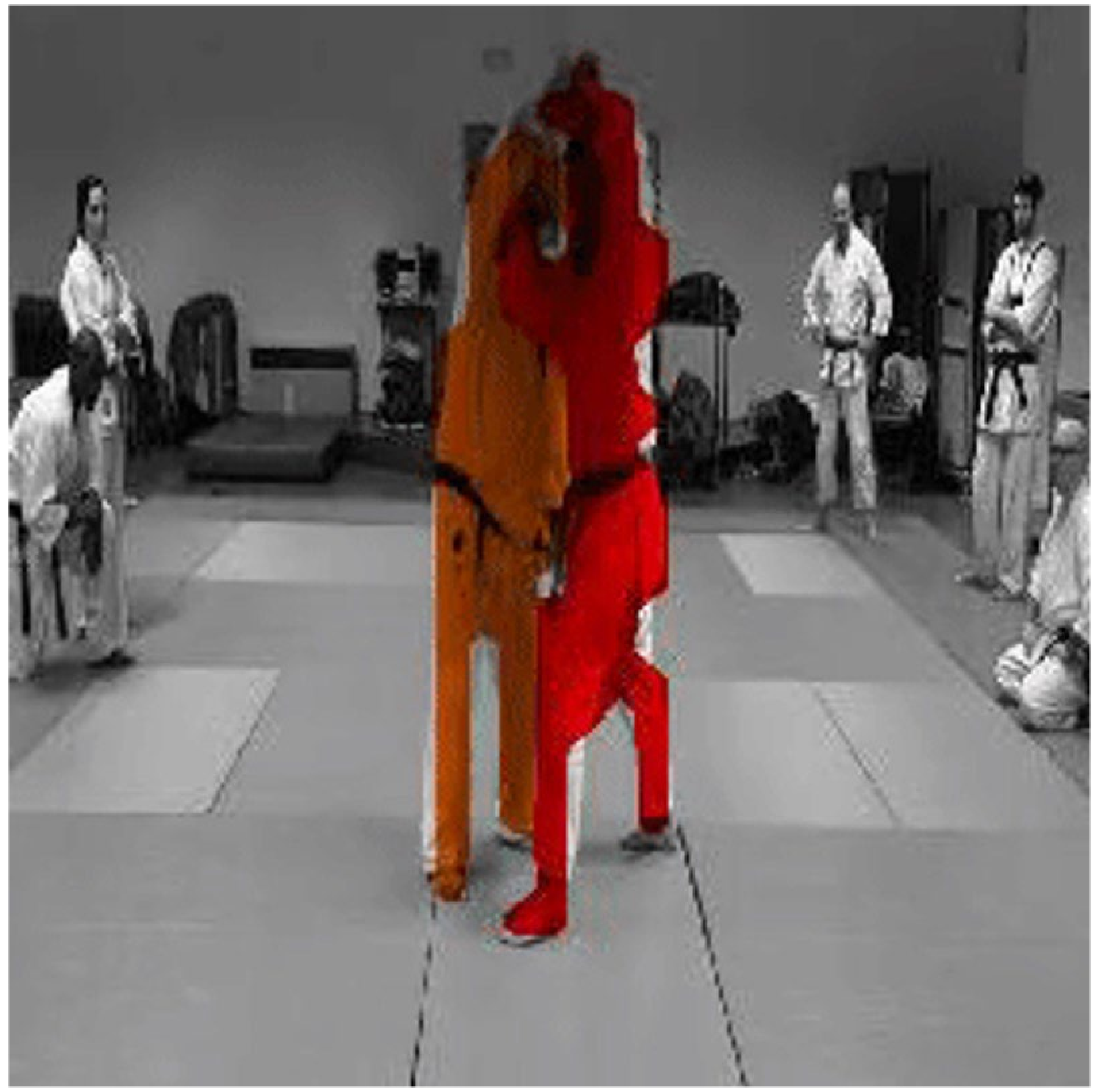
target frames (gray)

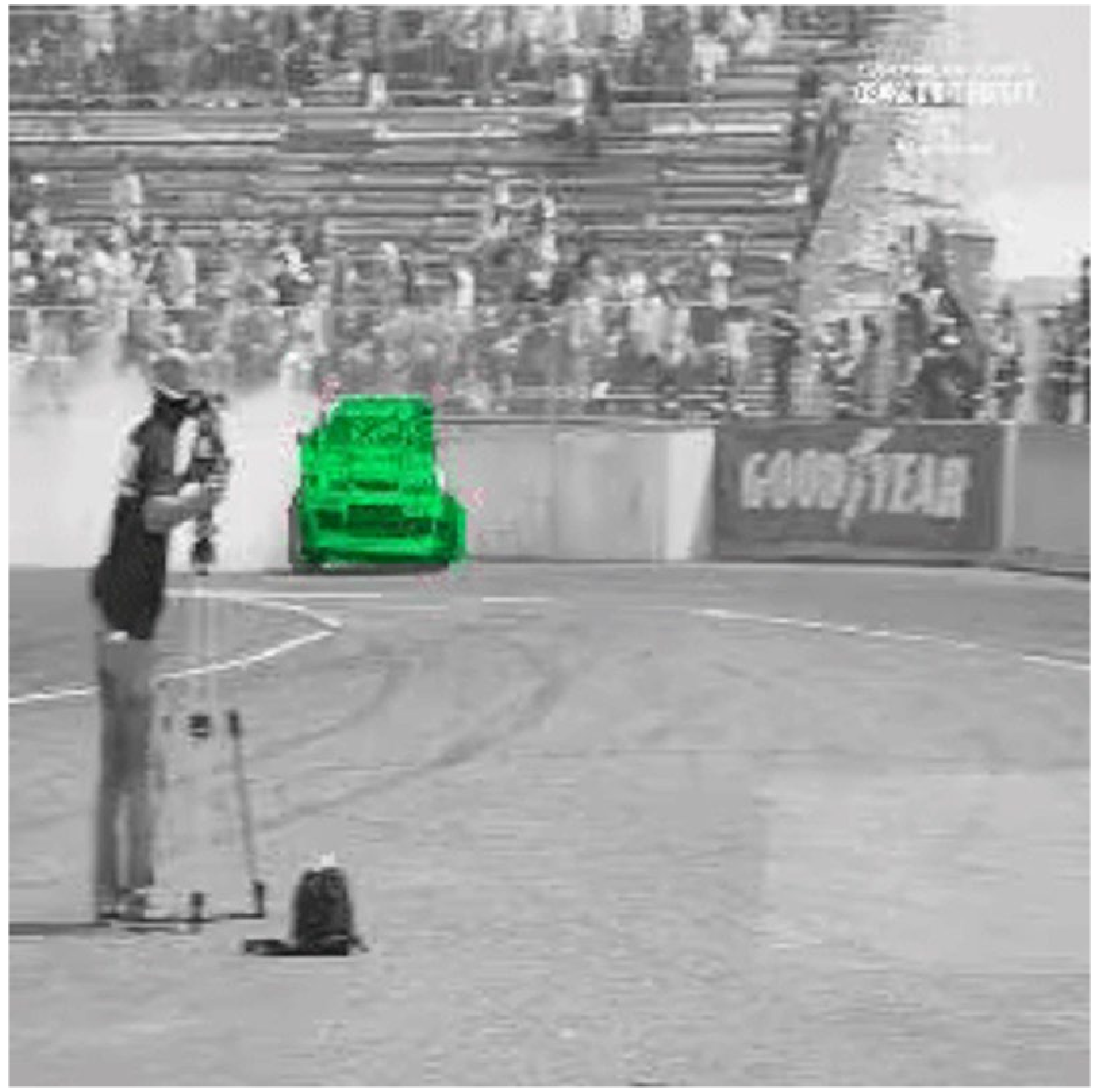
predicted color

Source: [Google AI blog post]()

Tracking emerges from colorization

Propagate segmentation masks using learned attention

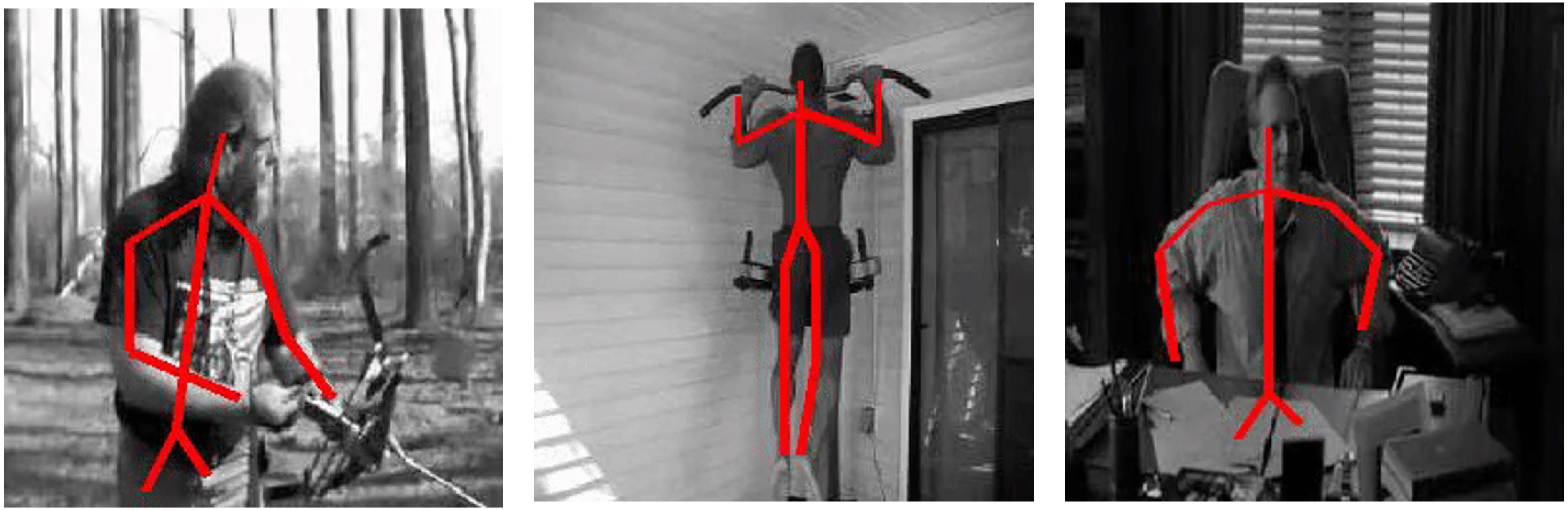




Source: [Google AI blog post]

Tracking emerges from colorization

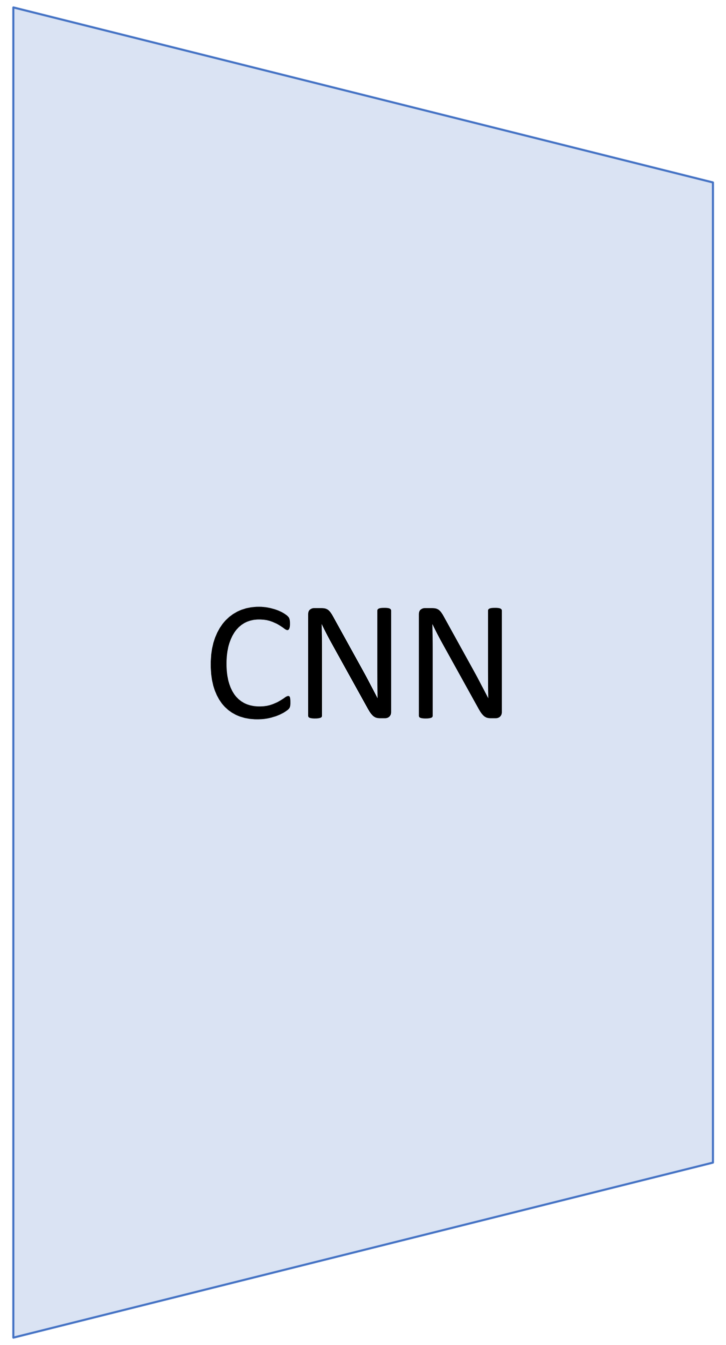
Propagate pose keypoints using learned attention



Source: [Google AI blog post]

Deep Clustering

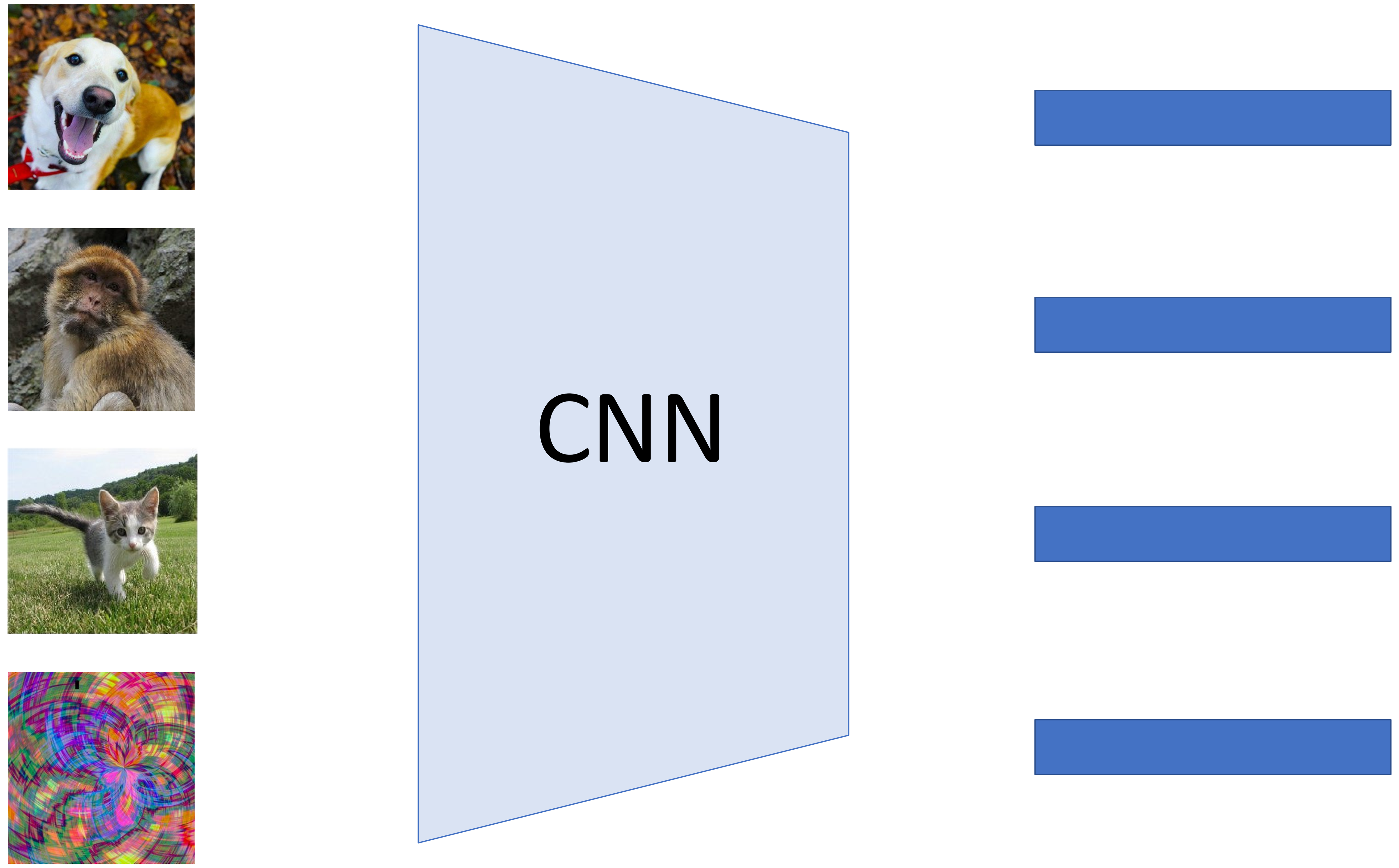
(1) Randomly initialize a CNN



Caron et al, "Deep Clustering for Unsupervised Learning of Visual Features", ECCV 2018
Caron et al, "Unsupervised Pre-Training of Image Features on Non-Curated Data", ICCV 2019
Yan et al, "ClusterFit: Improving Generalization of Visual Representations", CVPR 2020
Caron et al, "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments", NeurIPS 2020

Deep Clustering

(1) Randomly initialize a CNN

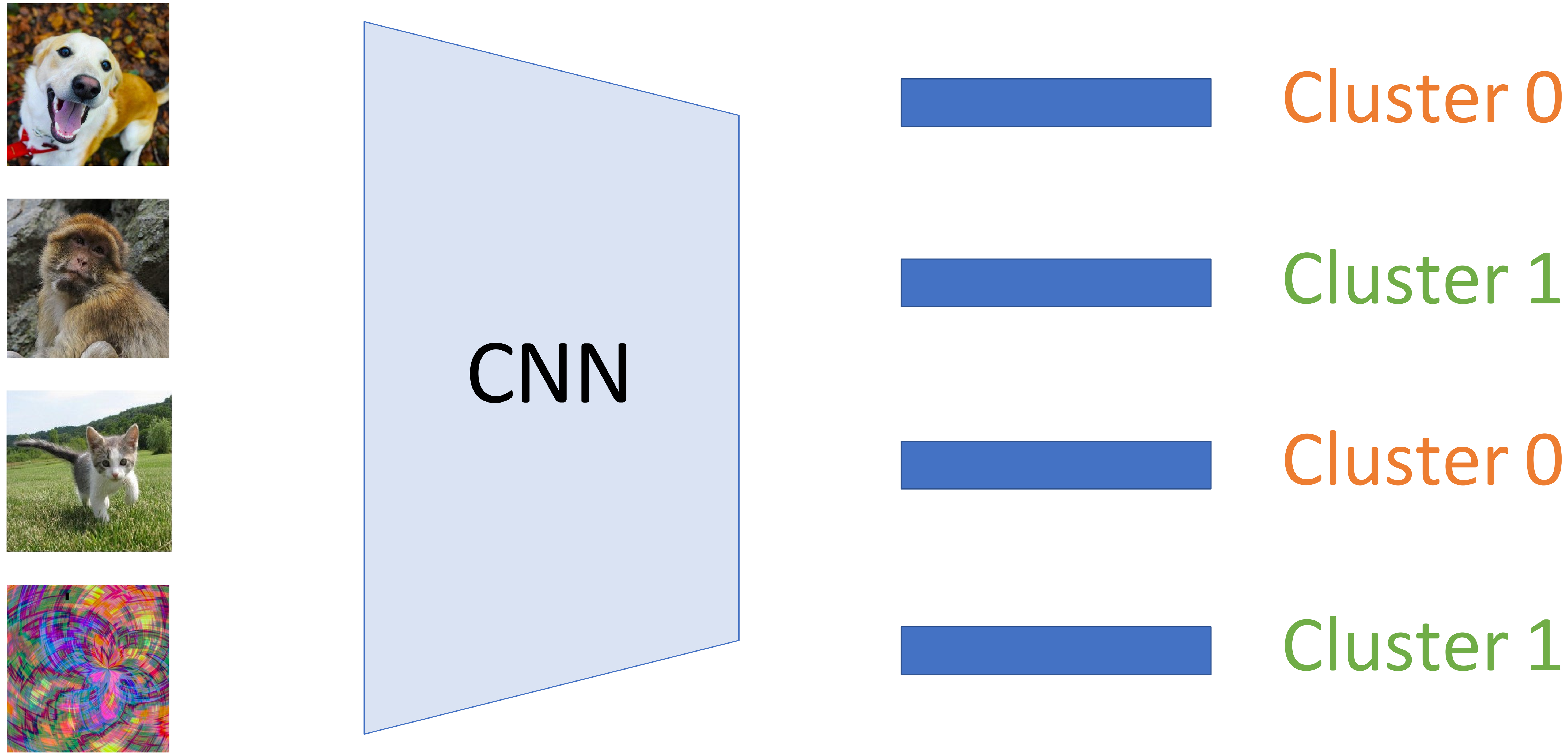


(2) Run many images through CNN, get their final-layer features

Caron et al, "Deep Clustering for Unsupervised Learning of Visual Features", ECCV 2018
Caron et al, "Unsupervised Pre-Training of Image Features on Non-Curated Data", ICCV 2019
Yan et al, "ClusterFit: Improving Generalization of Visual Representations", CVPR 2020
Caron et al, "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments", NeurIPS 2020

Deep Clustering

(1) Randomly initialize a CNN



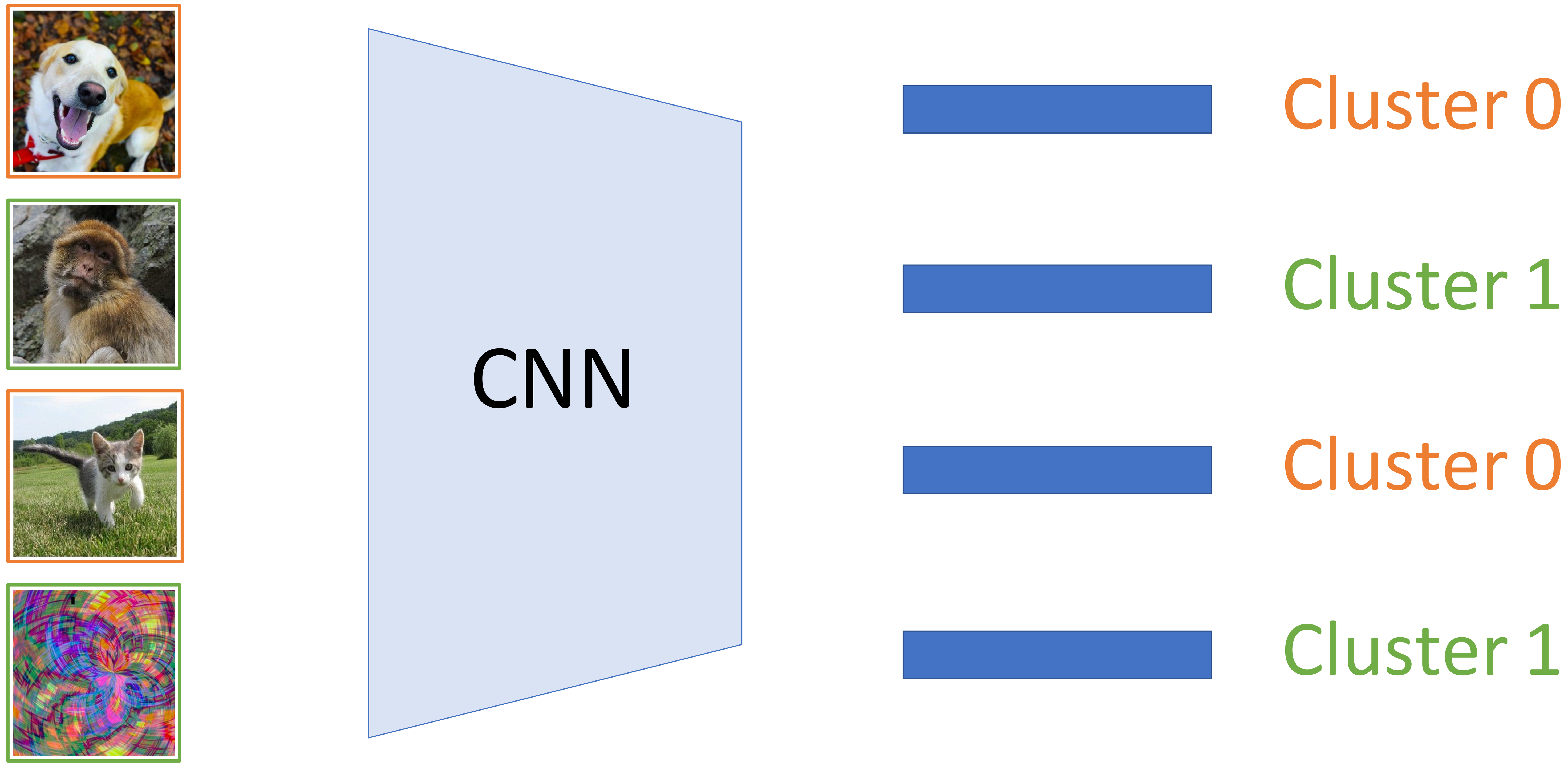
(3) Cluster the features with K-Means; record cluster for each feature

(2) Run many images through CNN, get their final-layer features

Caron et al, “Deep Clustering for Unsupervised Learning of Visual Features”, ECCV 2018
Caron et al, “Unsupervised Pre-Training of Image Features on Non-Curated Data”, ICCV 2019
Yan et al, “ClusterFit: Improving Generalization of Visual Representations”, CVPR 2020
Caron et al, “Unsupervised Learning of Visual Features by Contrasting Cluster Assignments“, NeurIPS 2020

Deep Clustering

(1) Randomly initialize a CNN



(3) Cluster the features with K-Means; record cluster for each feature

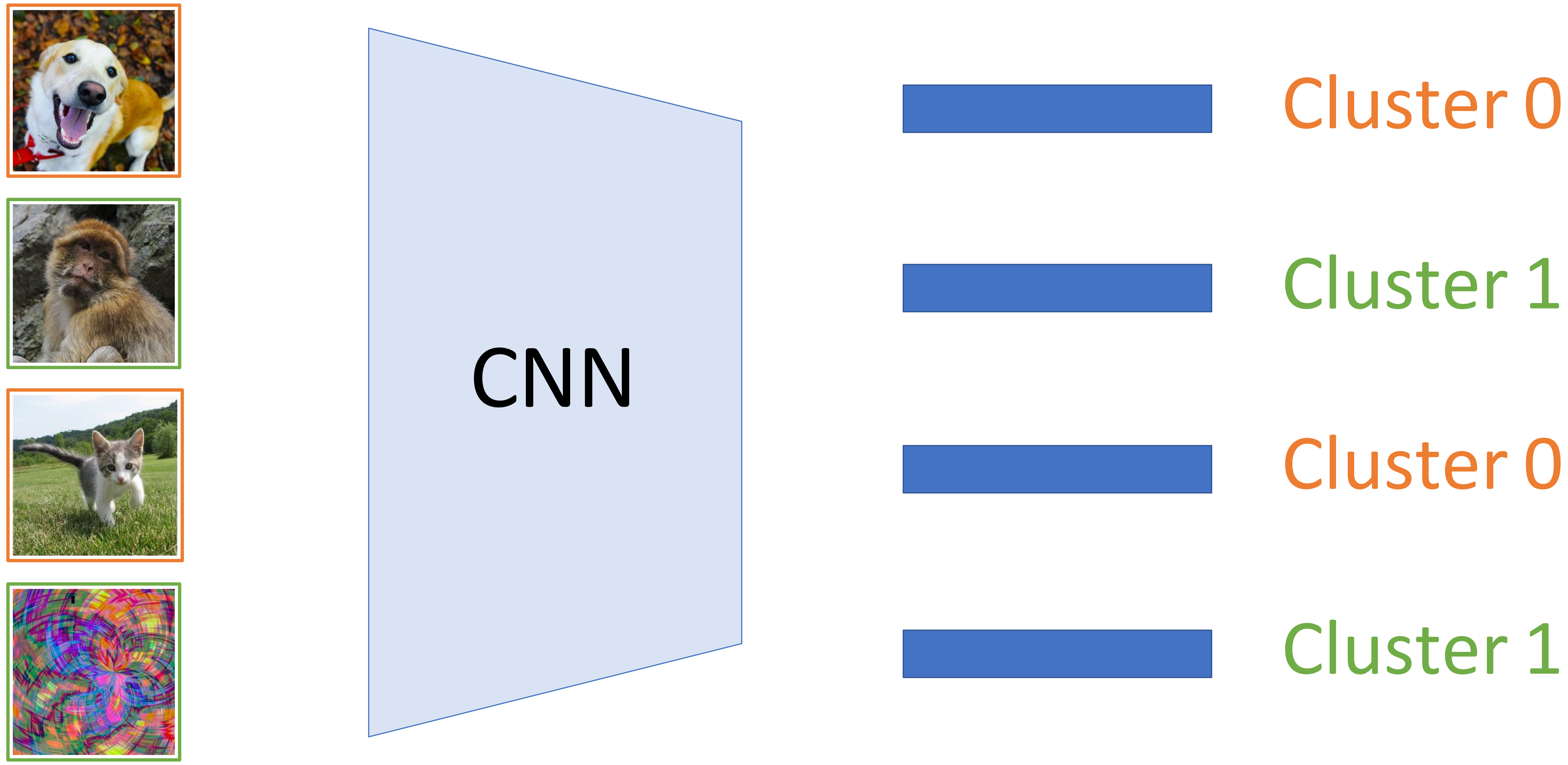
(4) Use cluster assignments as pseudo-labels for each image; train the CNN to predict cluster assignments

(2) Run many images through CNN, get their final-layer features

Caron et al, "Deep Clustering for Unsupervised Learning of Visual Features", ECCV 2018
Caron et al, "Unsupervised Pre-Training of Image Features on Non-Curated Data", ICCV 2019
Yan et al, "ClusterFit: Improving Generalization of Visual Representations", CVPR 2020
Caron et al, "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments", NeurIPS 2020

Deep Clustering

(1) Randomly initialize a CNN



(2) Run many images through CNN, get their final-layer features

(3) Cluster the features with K-Means; record cluster for each feature

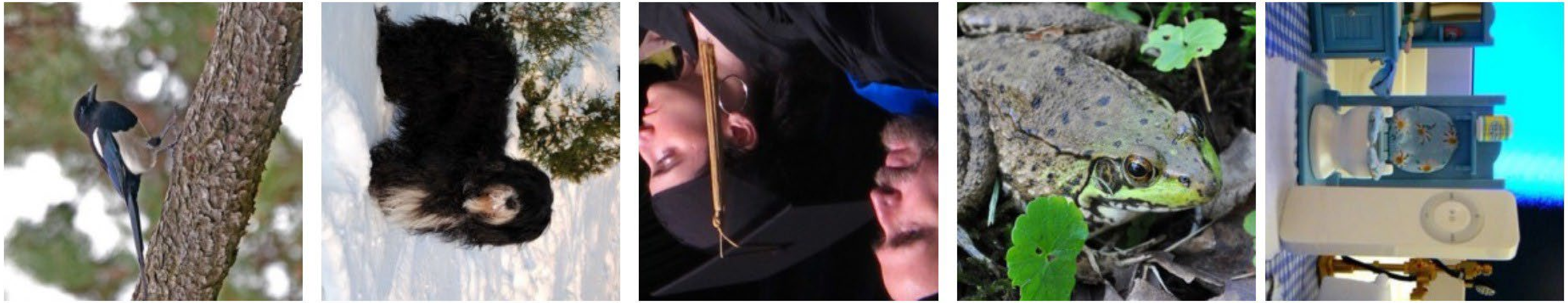
(4) Use cluster assignments as pseudo-labels for each image; train the CNN to predict cluster assignments

(5) Repeat: GOTO (2)

Caron et al, "Deep Clustering for Unsupervised Learning of Visual Features", ECCV 2018
Caron et al, "Unsupervised Pre-Training of Image Features on Non-Curated Data", ICCV 2019
Yan et al, "ClusterFit: Improving Generalization of Visual Representations", CVPR 2020
Caron et al, "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments", NeurIPS 2020

RotNet: Predict Rotation

4-way classification task: How much was each image rotated? (0, 90, 180, or 270 degrees)

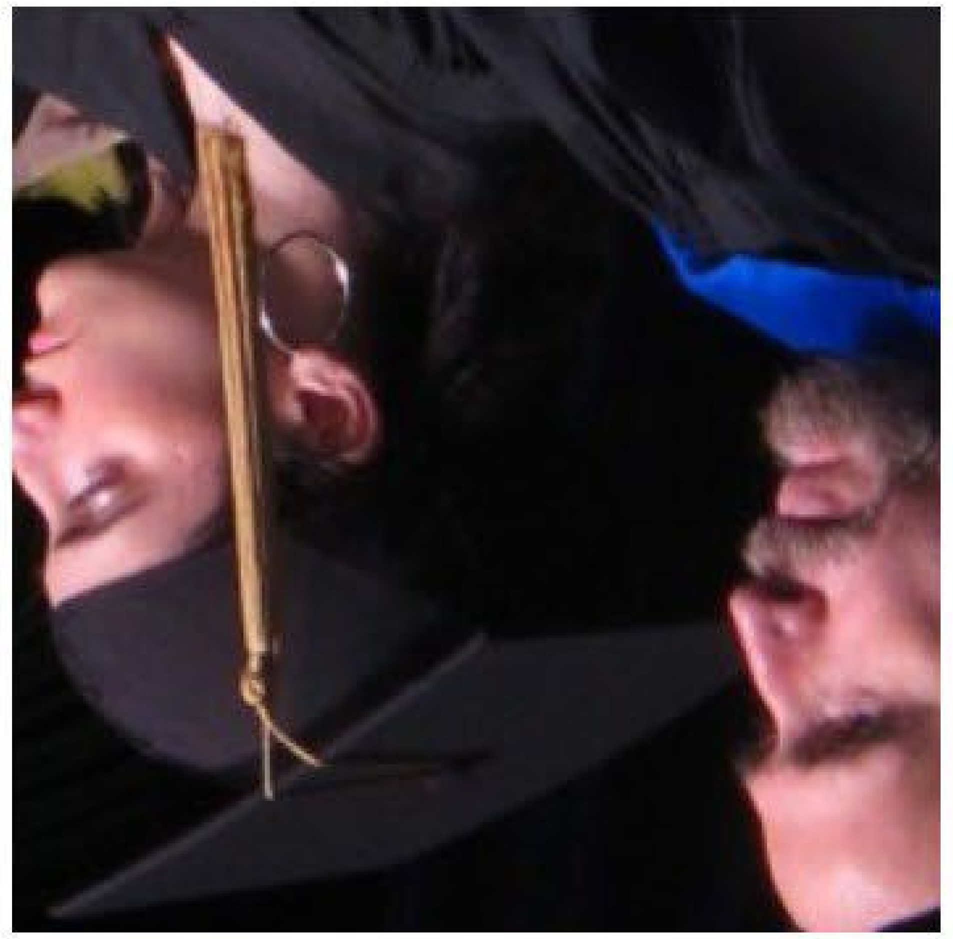


Gidaris et al, “Unsupervised representation learning by predicting image rotations”, ICLR 2018

RotNet: Predict Rotation

4-way classification task: How much was each image rotated? (0, 90, 180, or 270 degrees)



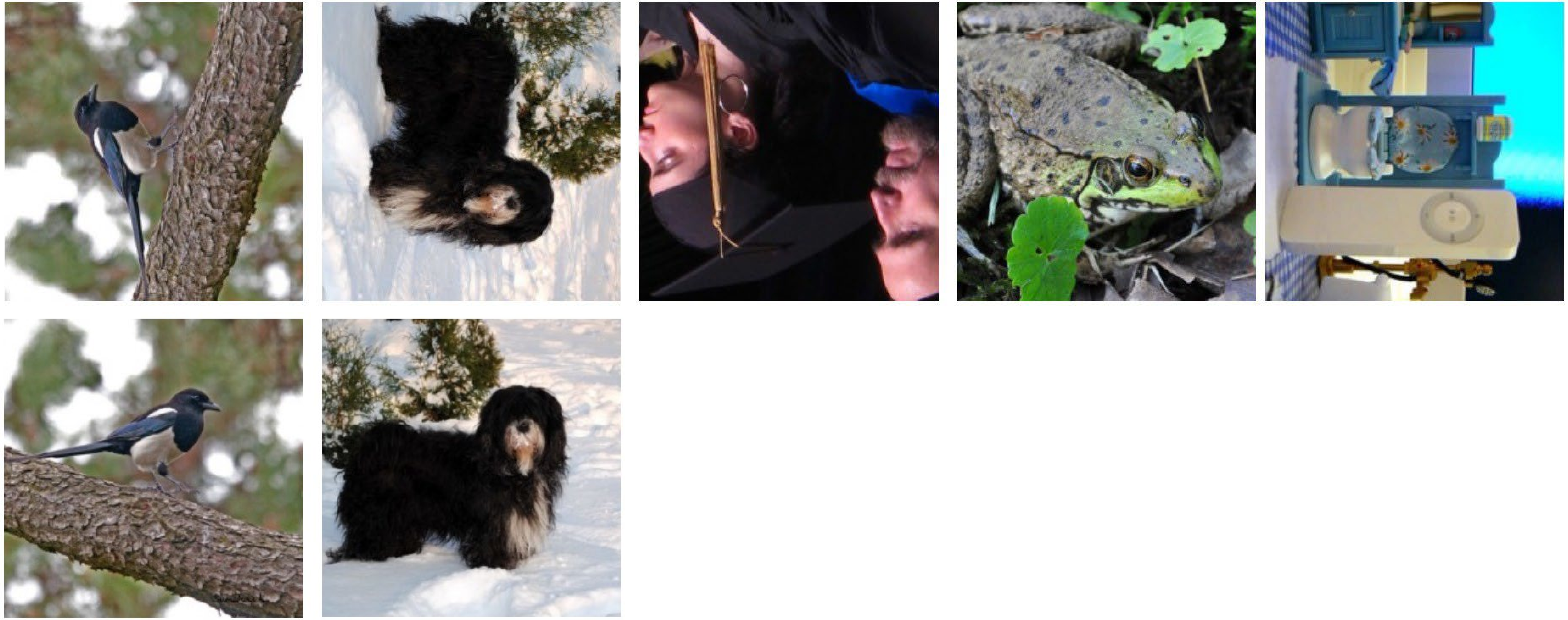


90

Gidaris et al, “Unsupervised representation learning by predicting image rotations”, ICLR 2018

RotNet: Predict Rotation

4-way classification task: How much was each image rotated? (0, 90, 180, or 270 degrees)

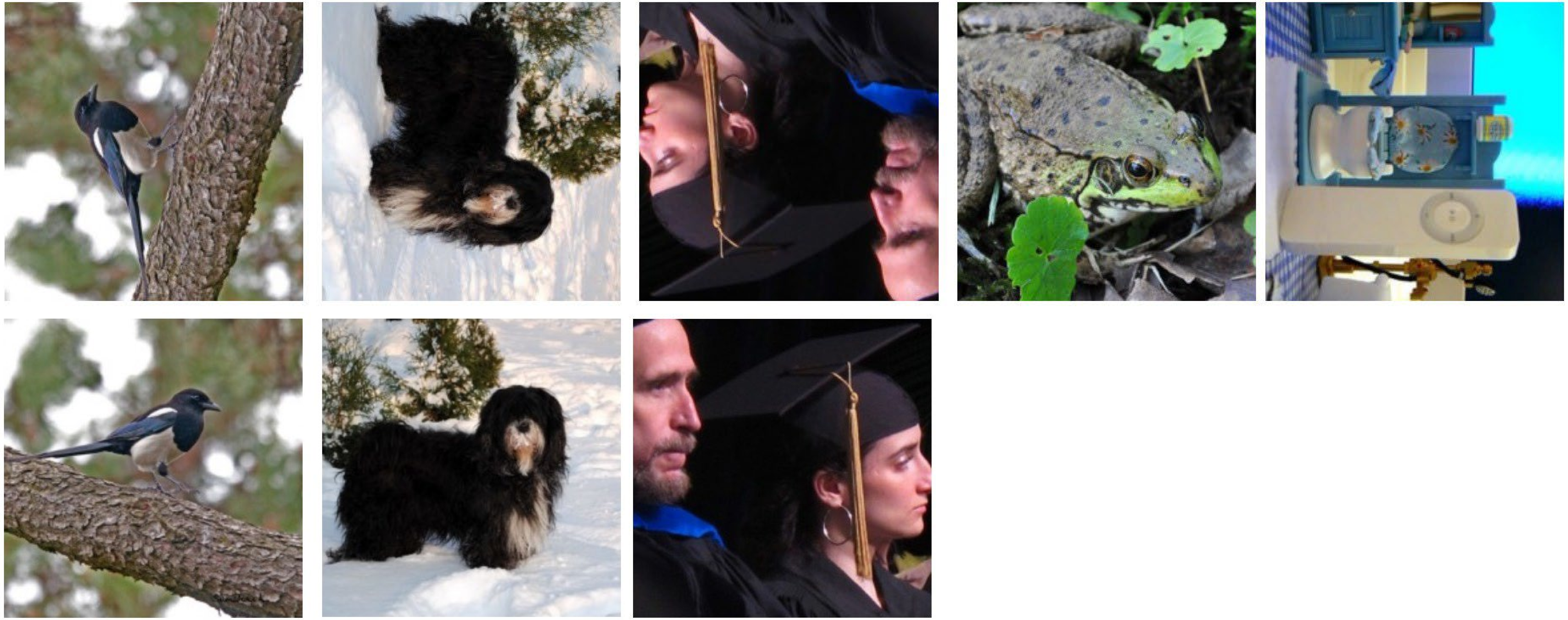


90

Gidaris et al, “Unsupervised representation learning by predicting image rotations”, ICLR 2018

RotNet: Predict Rotation

4-way classification task: How much was each image rotated? (0, 90, 180, or 270 degrees)



90 270 180

Gidaris et al, “Unsupervised representation learning by predicting image rotations”, ICLR 2018

RotNet: Predict Rotation

4-way classification task: How much was each image rotated? (0, 90, 180, or 270 degrees)

90 270 180

Gidaris et al, “Unsupervised representation learning by predicting image rotations”, ICLR 2018

RotNet: Predict Rotation

4-way classification task: How much was each image rotated? (0, 90, 180, or 270 degrees)

90 270 180 0 270

Gidaris et al, “Unsupervised representation learning by predicting image rotations”, ICLR 2018

Summary:
pretext tasks via 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 often do not care about the performance of these pretext tasks, but rather how useful the learned features are for downstream tasks (classification, detection, segmentation).

Summary:
pretext tasks via image transformations

- Pretext tasks focus on “visual common sense”
  - e.g., predict rotations, inpainting, rearrangement, and colorization.
- We often do not 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
  - (2) the learned representations may not be general.

Which SSL Method is best?

Fair evaluation of SSL methods is very hard …
No theory, so we need to rely on experiments !!!

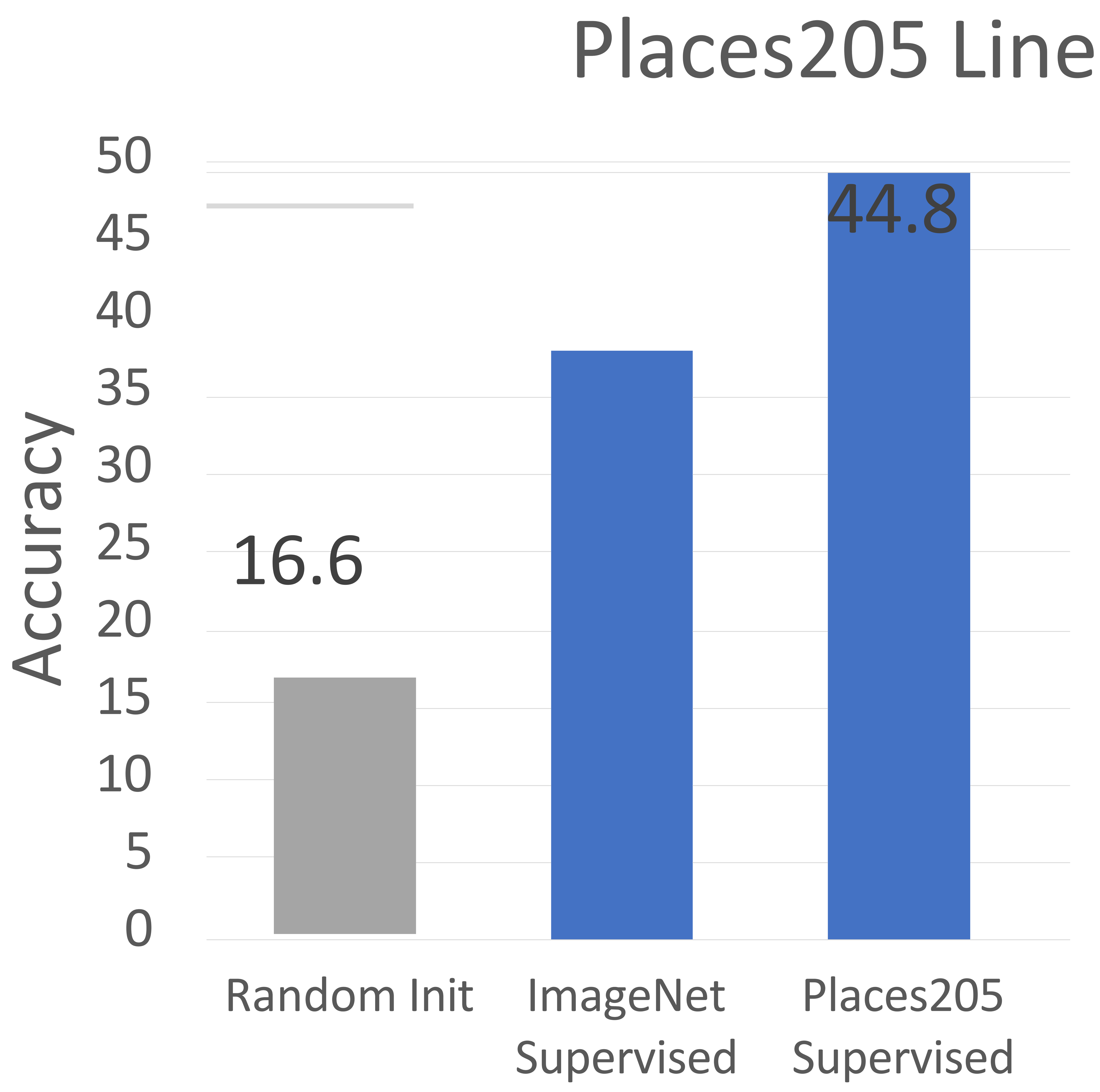
Many choices in experimental setup, huge variations from paper to paper:

- CNN architecture? AlexNet, ResNet50, something else?
- Pretraining dataset? ImageNet, or something else?
- Downstream task? ImageNet classification, detection, something else?
- Pretraining hyperparameters? Learning rates, training iterations, data augmentation?

- Transfer learning protocol?
    - Linear probe? From which layer? How to train linear models? SGD, something else?
    - Transfer learning hyperparameters? Data augmentation or BatchNorm during transfer learning?
    - Fine-tune? which layer? Linear or nonlinear? Fine-tuning hyperparameters?
    - KNN? What value of K? Normalization on features?

Which SSL Method is best?

Some papers have tried to do fair comparisons of many SSL methods



Goyal et al, "Scaling and Benchmarking Self-Supervised Visual Representation Learning", ICCV 2019

Which SSL Method is best?

Some papers have tried to do fair comparisons of many SSL methods

Goyal et al, "Scaling and Benchmarking Self-Supervised Visual Representation Learning", ICCV 2019

Which SSL Method is best?

Some papers have tried to do fair comparisons of many SSL methods

Goyal et al, "Scaling and Benchmarking Self-Supervised Visual Representation Learning", ICCV 2019

Which SSL Method is best?

Some papers have tried to do fair comparisons of many SSL methods

Goyal et al, "Scaling and Benchmarking Self-Supervised Visual Representation Learning", ICCV 2019

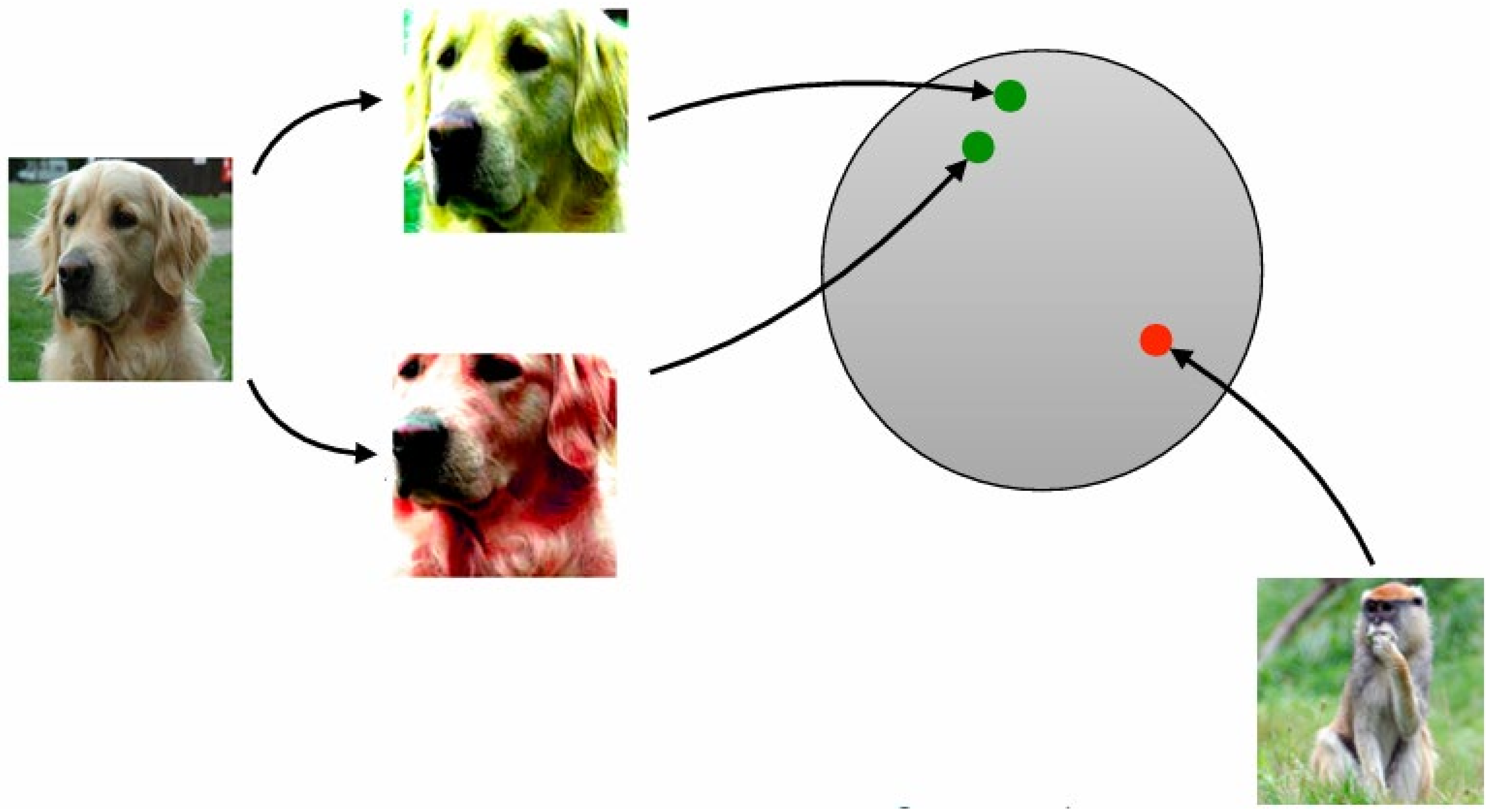
Let’s take a step back ...

A simpler idea ...

Similarity based Representation Learning

- Build representations via feedback in terms of similarity:

  pairs of similar / dissimilar inputs



Background: Metric Learning

In mathematics, a **metric space** is a set together with a notion of *distance* between its elements, usually called points. The distance is measured by a function called a **metric** or **distance function**.[1] Metric spaces are a general setting for studying many of the concepts of mathematical analysis and geometry.

- How should we compute similarity between images?

- Idea 1: Euclidean distance in pixel space $\|x_1 - x_2\|_2$
  - Images with the same background but different foreground will have very high similarity (e.g. cat in snow vs dog in snow) – BAD!

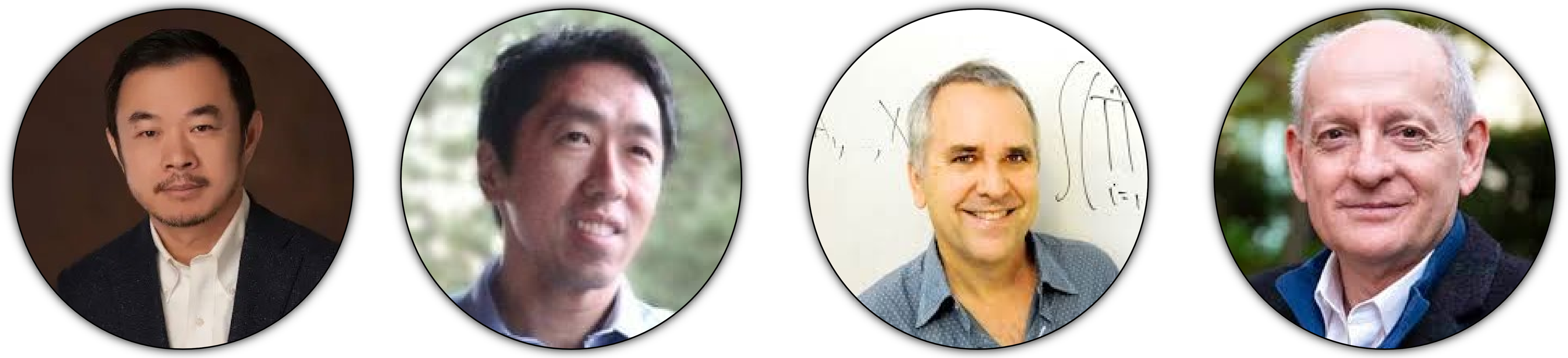
- Goal: learn a metric where:
  - Data points that belong together are similar (closer)
  - Data points that are different are dissimilar (farther)

Background: Metric Learning

Distance metric learning, with application to clustering with side-information

Eric P. Xing, Andrew Y. Ng, Michael I. Jordan and Stuart Russell
University of California, Berkeley
Berkeley, CA 94720
{epxing,ang,jordan,russell}@cs.berkeley.edu

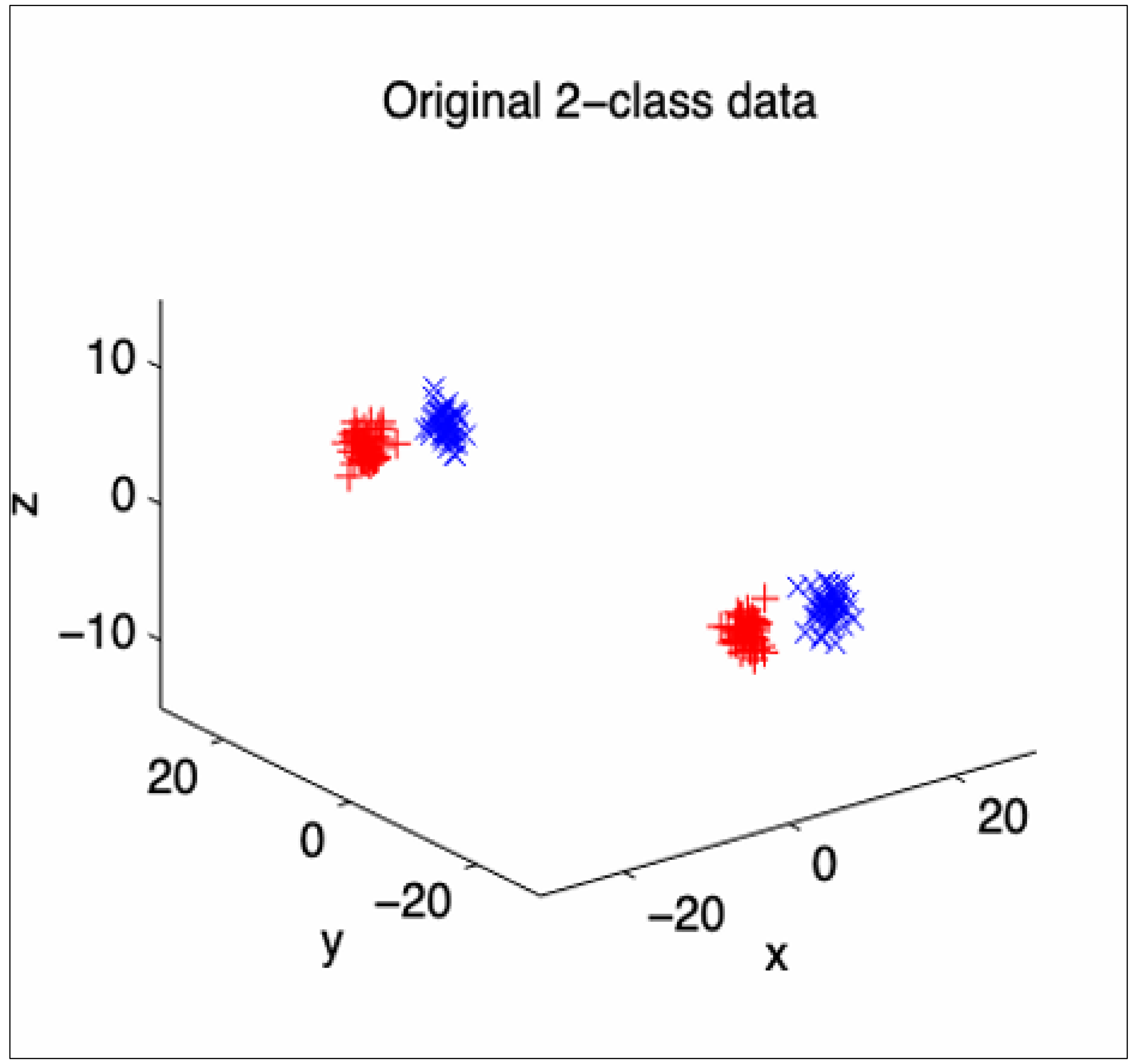
introduced the term and problem in 2003

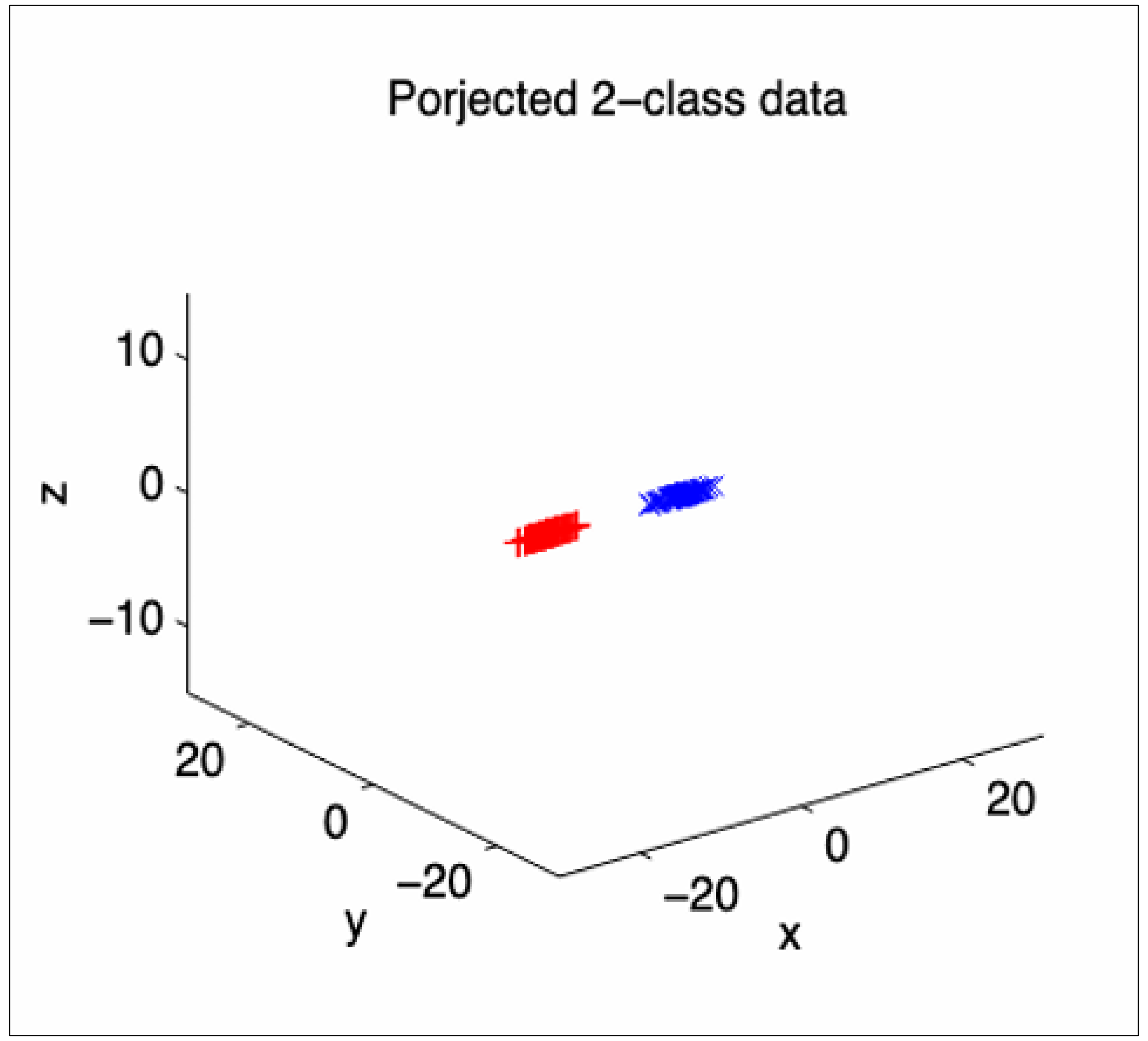


Many related ideas and follow-up work

$$\min_{\mathbf{A}\succeq 0} \sum_{(i,j)\in\mathcal{S}} d_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j)^2$$ min distance of similar points

$$\text{s.t.} \sum_{(k,\ell)\in\mathcal{D}} d_{\mathbf{A}}(\mathbf{x}_k, \mathbf{x}_\ell)^2 \geq 1$$ keep distance of dissimilar points





Contrastive Learning

Problem 1: How to compute similarity if we don't have labels for images?

Contrastive Learning

Problem 1: How to compute similarity if we don't have labels for images?
Solution? Euclidean Distance between features $\|\phi(x_1) - \phi(x_2)\|_2$

Contrastive Learning

Problem 1: How to compute similarity if we don't have labels for images?
Solution? Euclidean Distance between features $||\phi(x_1) - \phi(x_2)||_2$

Problem 2: Objective Function ?

Similar images should have similar features Dissimilar images should have dissimilar features

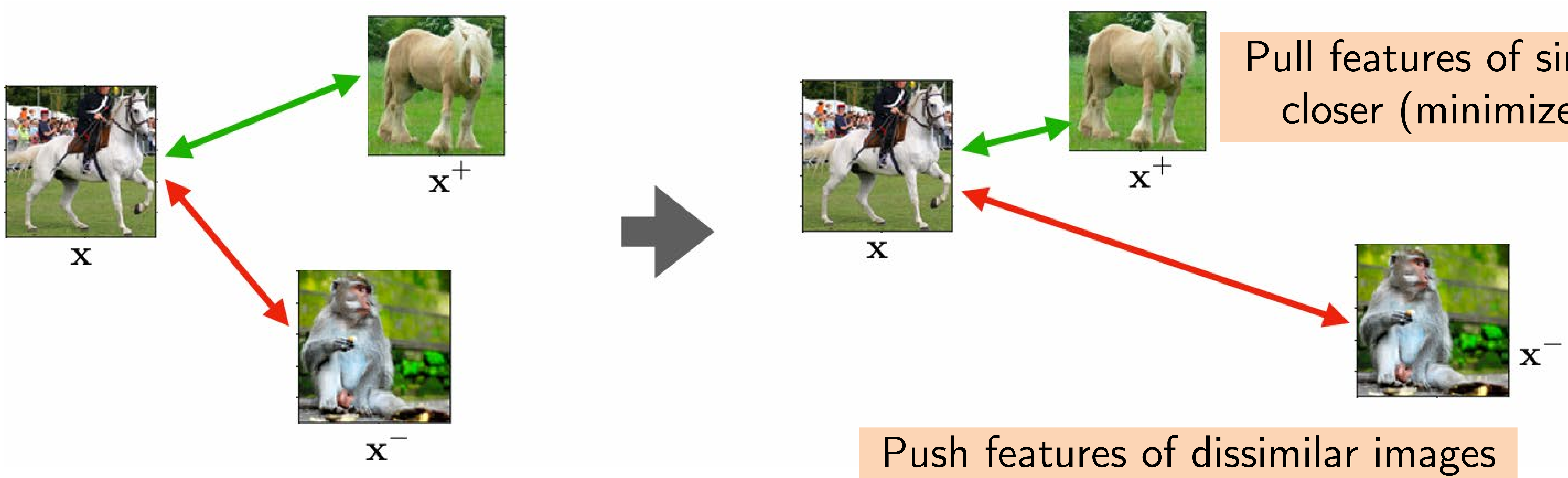
Contrastive Learning

Problem 1: How to compute similarity if we don't have labels for images?
Solution? Euclidean Distance between features $\|\phi(x_1) - \phi(x_2)\|_2$

Problem 2: Objective Function ?

Similar images should have similar features **Dissimilar** images should have dissimilar features

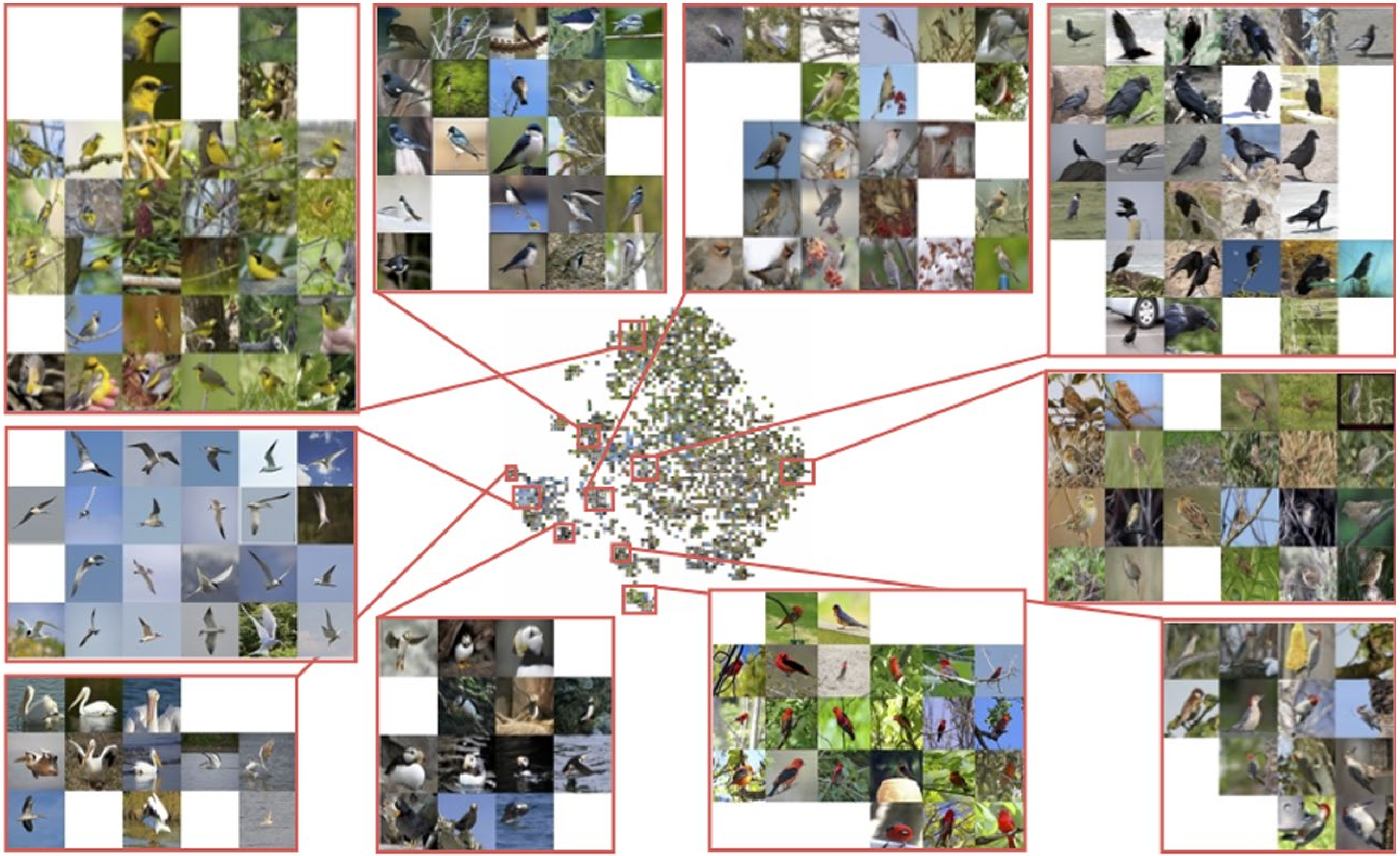
Examples of Contrastive Pairs



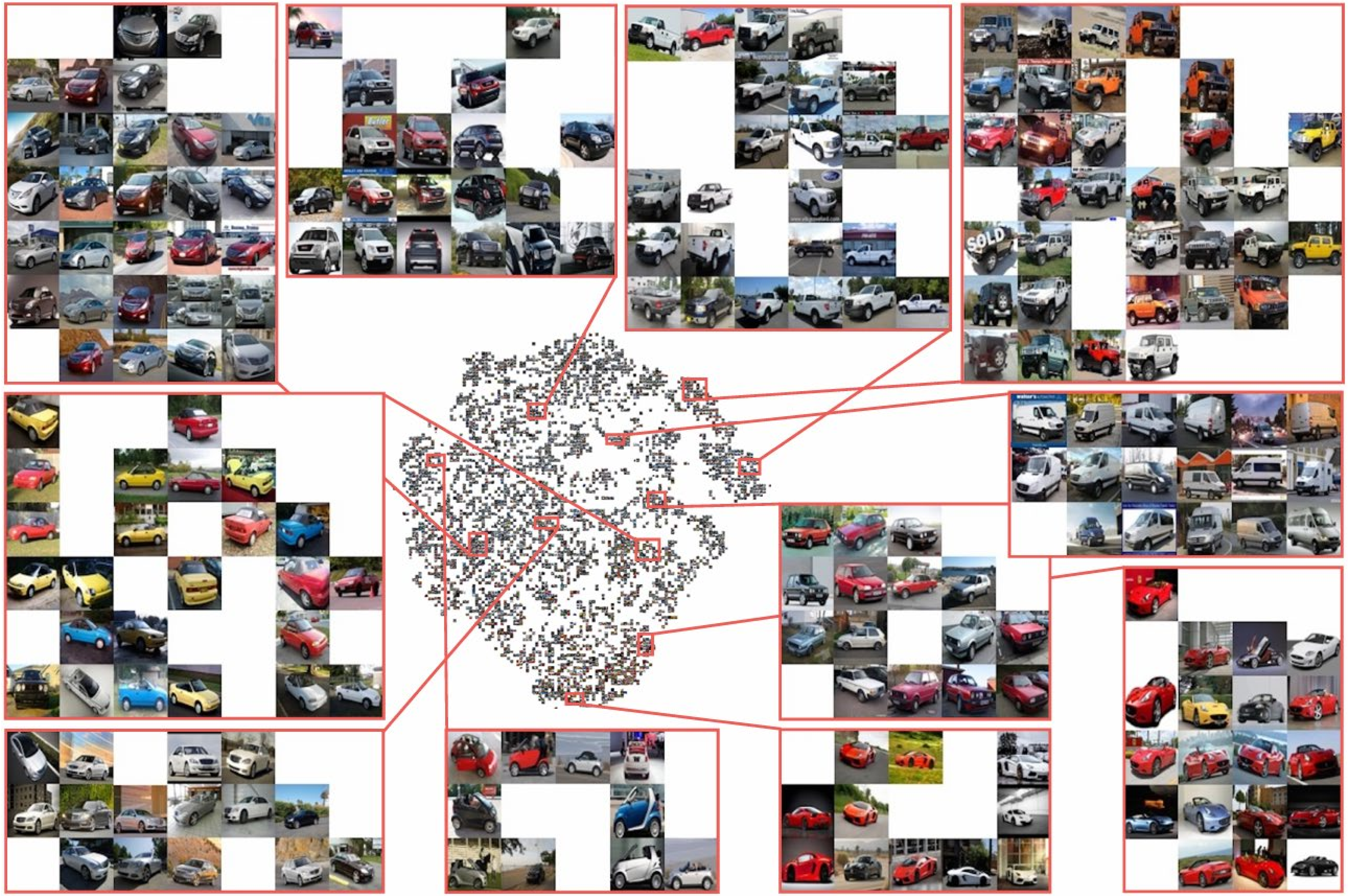
Pull features of similar images closer (minimize distance)

Push features of dissimilar images apart (maximize distance)

Examples of the Embedding Space



Examples of the Embedding Space

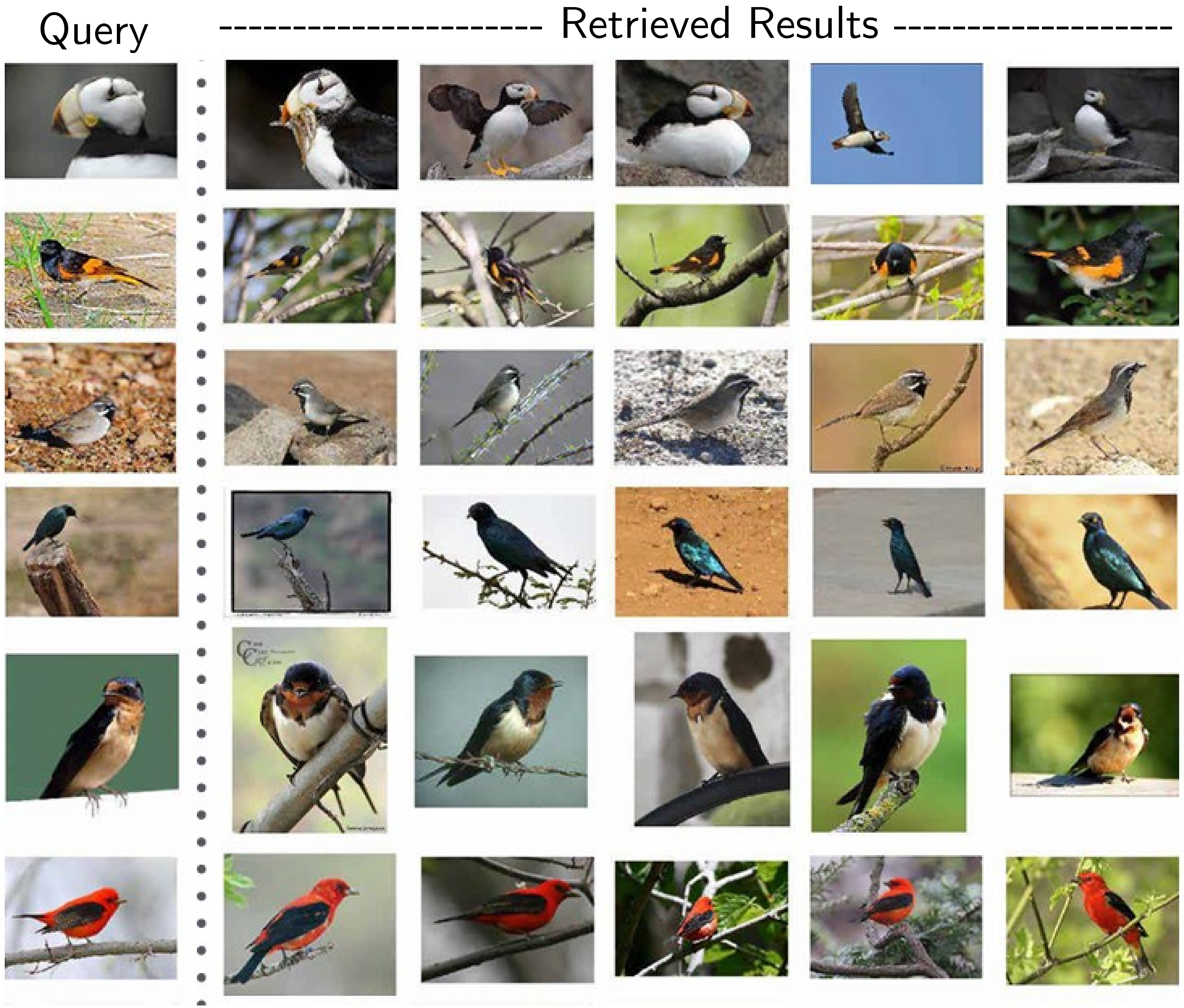


What can you do with this embedding space?

What can you do with this embedding space?

RETRIEVAL

- Given a query image (left column), find similar images
- All you have to do is find the nearest neighbors in the embedding space and return the results
- Embedding space now has a notion of "similarity"
  - Similar datapoints are neighbors
  - Dissimilar datapoints are not



results from Song et al. CVPR 2016

What can you do with this embedding space?

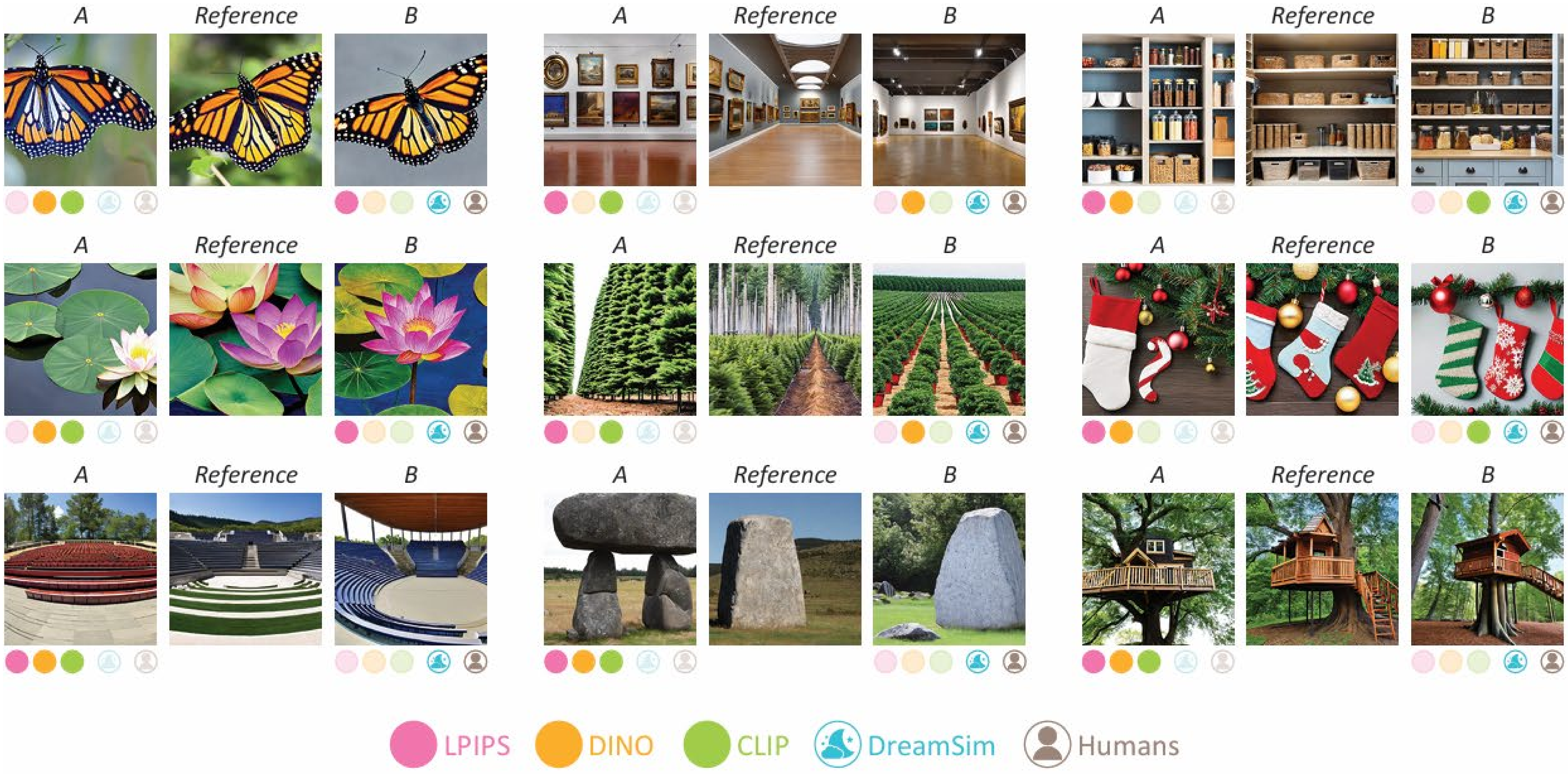
RETRIEVAL

- Given a query image (left column), find similar images
- All you have to do is find the nearest neighbors in the embedding space and return the results
- Embedding space now has a notion of "similarity"
  - Similar datapoints are neighbors
  - Dissimilar datapoints are not

Figure 1: Example retrieval results on our *Online Products* dataset using the proposed embedding. The images in the first column are the query images.

results from Song et al. CVPR 2016

Challenges: "Similarity" is hard …
What makes an image "similar" ?



Similar in:

- Pose
- Perspective
- Foreground color
- Number of items
- Object shape

figure: Fu*, Tamir*, Sundaram* et al 2023

Where can we get pairs of similar and dissimilar images from?

Similar images should have similar features

Dissimilar images should have dissimilar features

Where can we get pairs of similar and dissimilar images from?

DATA AUGMENTATION

Contrastive Learning with Data Augmentation

figure: Ranjay Krishna

Contrastive Learning Formulation

- We want:

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

x: reference sample; x^+ positive sample; x^- negative sample

- Objective:

Loss function given 1 positive sample and N - 1 negative samples:

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

Loss function given 1 positive sample and N - 1 negative samples:

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

x x^+

x x_1^- x_2^- x_3^- ...

Loss function given 1 positive sample and N - 1 negative samples:

$$L = -\mathbb{E}_X \left[\log \frac{\overline{\exp(s(f(x), f(x^+))}}{\underbrace{\exp(s(f(x), f(x^+))}_{\text{score for the positive pair}} + \underbrace{\sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))}_{\text{score for the N-1 negative pairs}}} \right]$$

This seems familiar …

Loss function given 1 positive sample and N - 1 negative samples:

$$L = -\mathbb{E}_X \left[\log \frac{\overline{\exp(s(f(x), f(x^+))}}{\underbrace{\exp(s(f(x), f(x^+))}_{\text{score for the positive pair}} + \underbrace{\sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))}_{\text{score for the N-1 negative pairs}}} \right]$$

This seems familiar …

Cross entropy loss for a N-way softmax classifier!

I.e., learn to find the positive sample from the N samples

Loss function given 1 positive sample and N - 1 negative samples:

$$L = -\mathbb{E}_X \left[\log \frac{\overbrace{\exp(s(f(x), f(x^+))}}{\underbrace{\exp(s(f(x), f(x^+))}_{\text{score for the positive pair}} + \underbrace{\sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))}_{\text{score for the N-1 negative pairs}}} \right]$$

This seems familiar …

Cross entropy loss for a N-way softmax classifier!

I.e., learn to find the positive sample from the N samples

Very similar to a softmax classifier

We want to compare the reference image against all other positive and negative images.

We can exponentiate and normalize these scores like we did with the softmax classifier.

Constrastive Learning Loss

Loss function given 1 positive sample and N - 1 negative samples:

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

Commonly known as the InfoNCE loss (van den Oord et al., 2018)

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: A Simple Framework for Contrastive learning

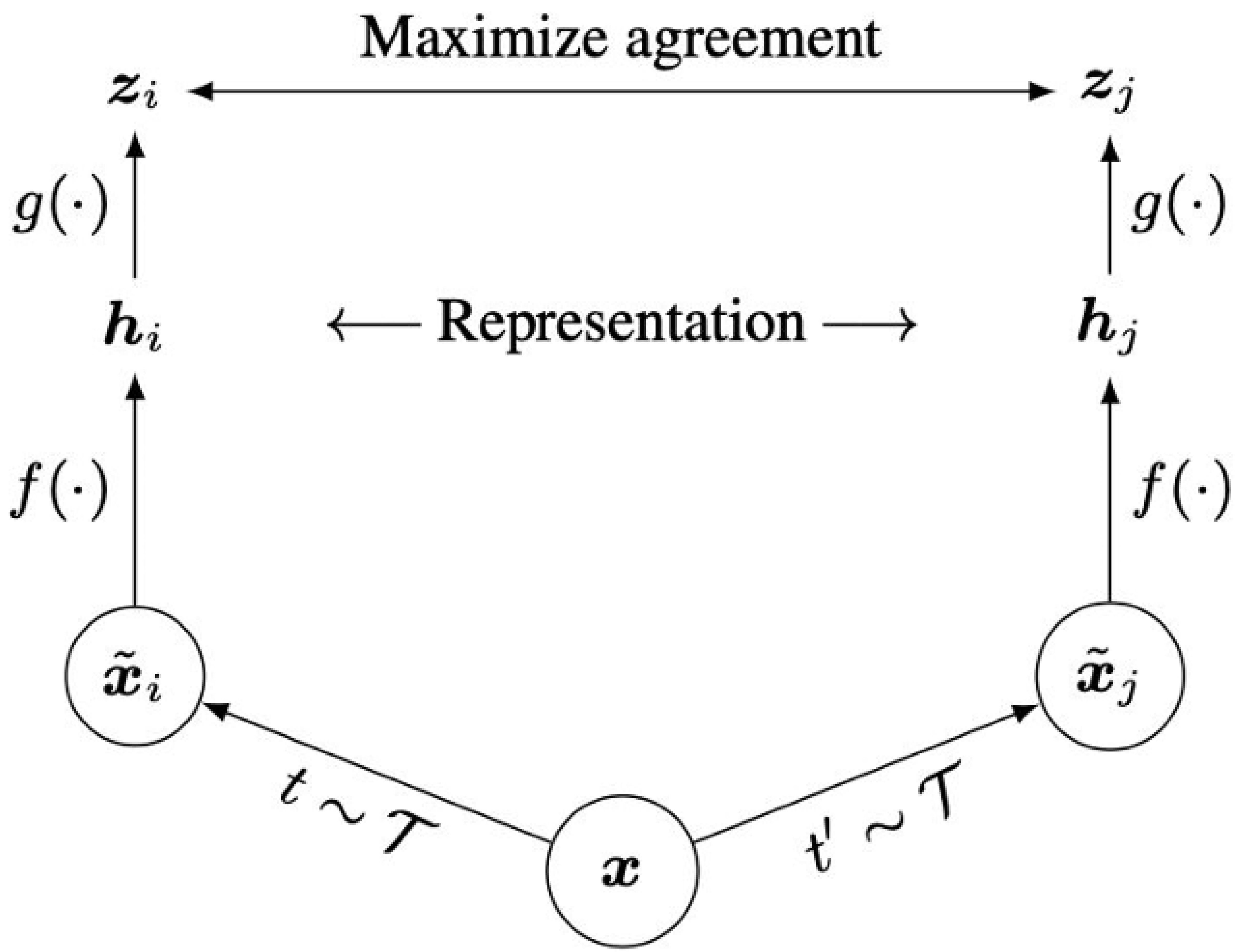
Cosine similarity as the score function:

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

Use a projection network ***h(·)*** to project features to a space where contrastive learning is applied

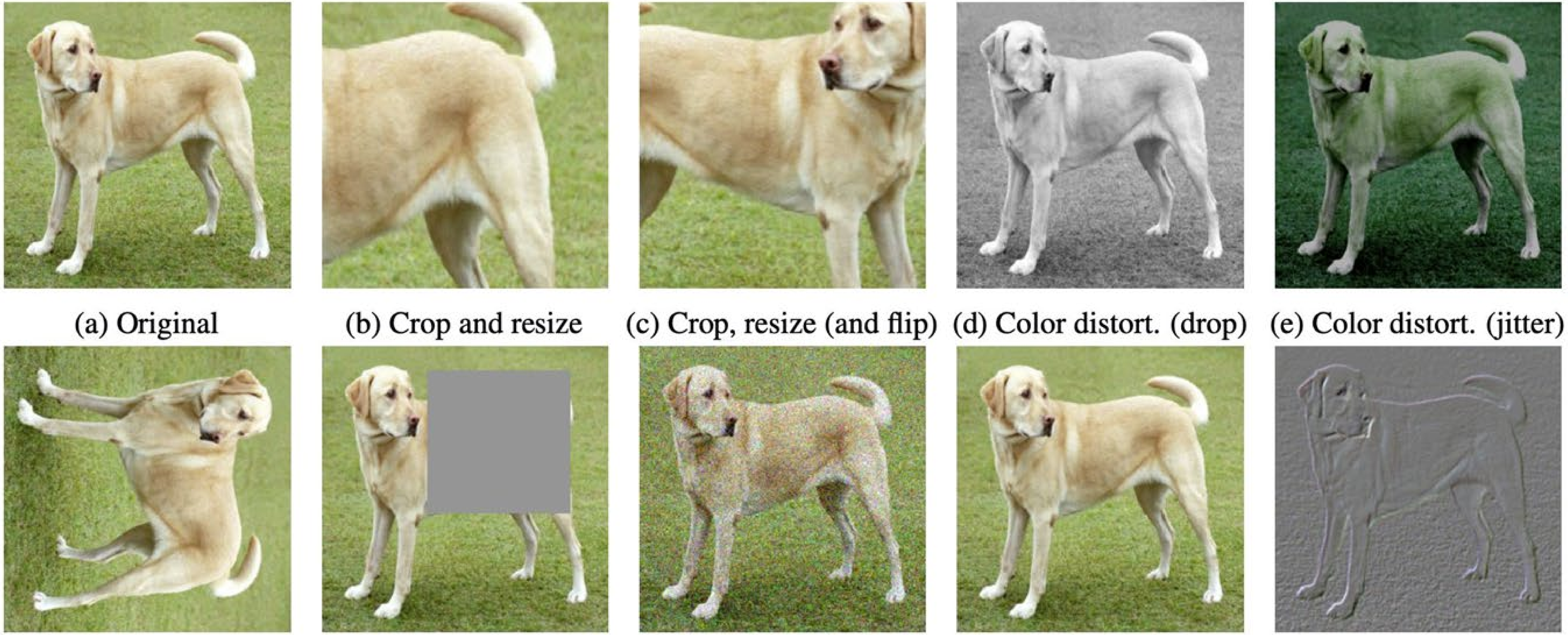
Generate positive samples through data augmentation:

- random cropping, random color distortion, and random blur.



Source: [Chen et al., 2020]

SimCLR: Data Augmentation Strategies



(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

(f) Rotate {90°, 180°, 270°} (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

Source: [Chen et al., 2020]

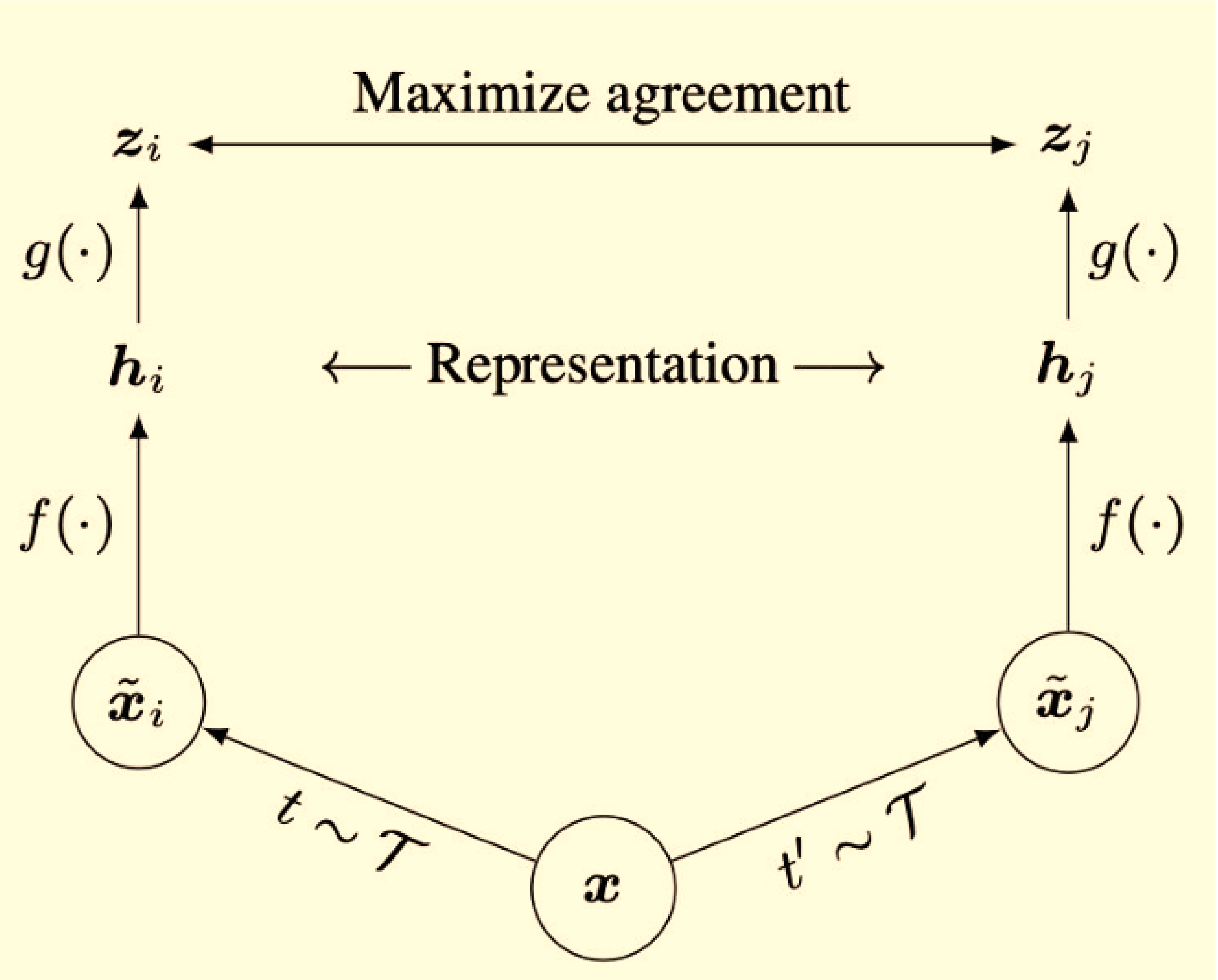
SimCLR: Algorithm Sketch

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**
    $\boxed{\text{draw two augmentation functions } t\sim\mathcal{T},\ t'\sim\mathcal{T}}$
    # the first augmentation
    $\boxed{\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)}$
    $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$ # representation
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$ # projection
    # the second augmentation
    $\boxed{\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)}$
    $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$ # 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** $\boxed{\ell(i,j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away $g(\cdot)$

Generate a positive pair by sampling data augmentation functions

Iterate through and use each of the 2N sample as reference, compute average loss

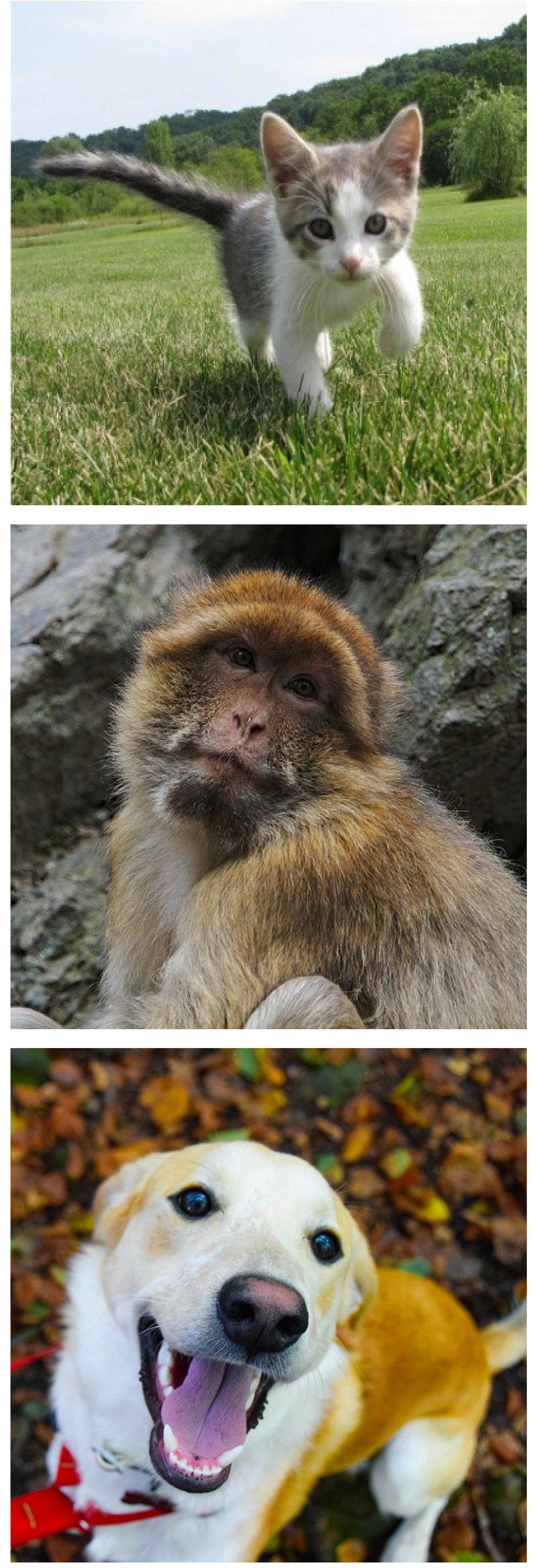


InfoNCE loss:
Use all non-positive samples in the batch as x^-

Source: [Chen et al., 2020]

SimCLR Training

Batch of
N images

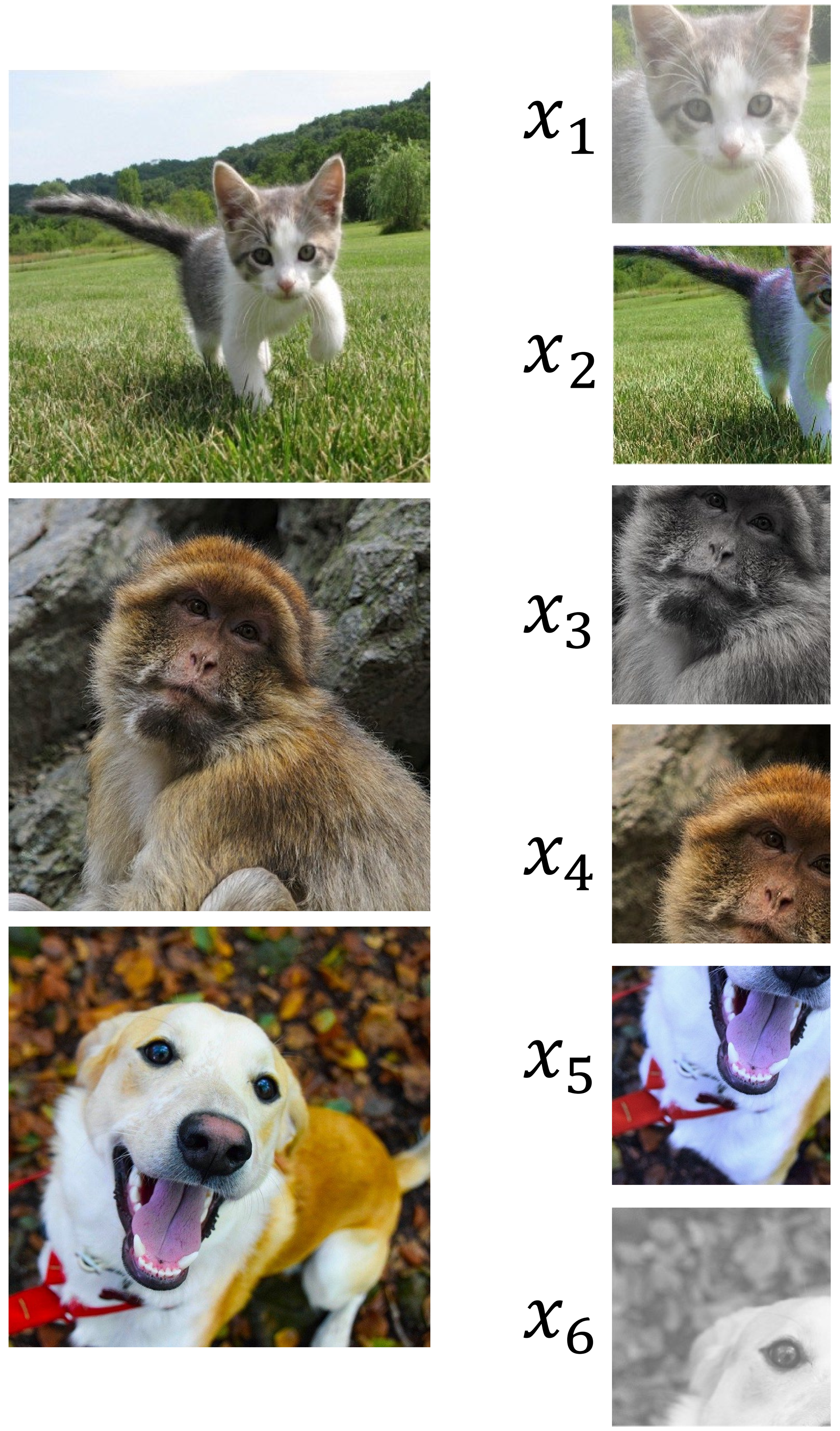


Hadsell et al, “Dimensionality Reduction by Learning and Invariant Mapping”, CVPR 2006
Wu et al, “Unsupervised Feature Learning by Non-Parametric Instance-Level Discrimination”, CVPR 2018
Van den Oord et al, “Representation Learning with Contrastive Predictive Coding”, NeurIPS 2018
Hjelm et al, “Learning deep representations by mutual information estimation and maximization”, ICLR 2019
Bachman et al, “Learning Representations by Maximizing Mutual Information Across Views”, NeurIPS 2019
Henaff et al, “Data-Efficient Image Recognition with Contrastive Predictive Coding”, ICML 2020
Tian et al, “Contrastive Multiview Coding”, ECCV 2020
He et al, “Momentum Contrast for Unsupervised Visual Representation Learning”, CVPR 2020
Chen et al, “A Simple Framework for Contrastive Learning of Visual Representations”, ICML 2020

SimCLR Training

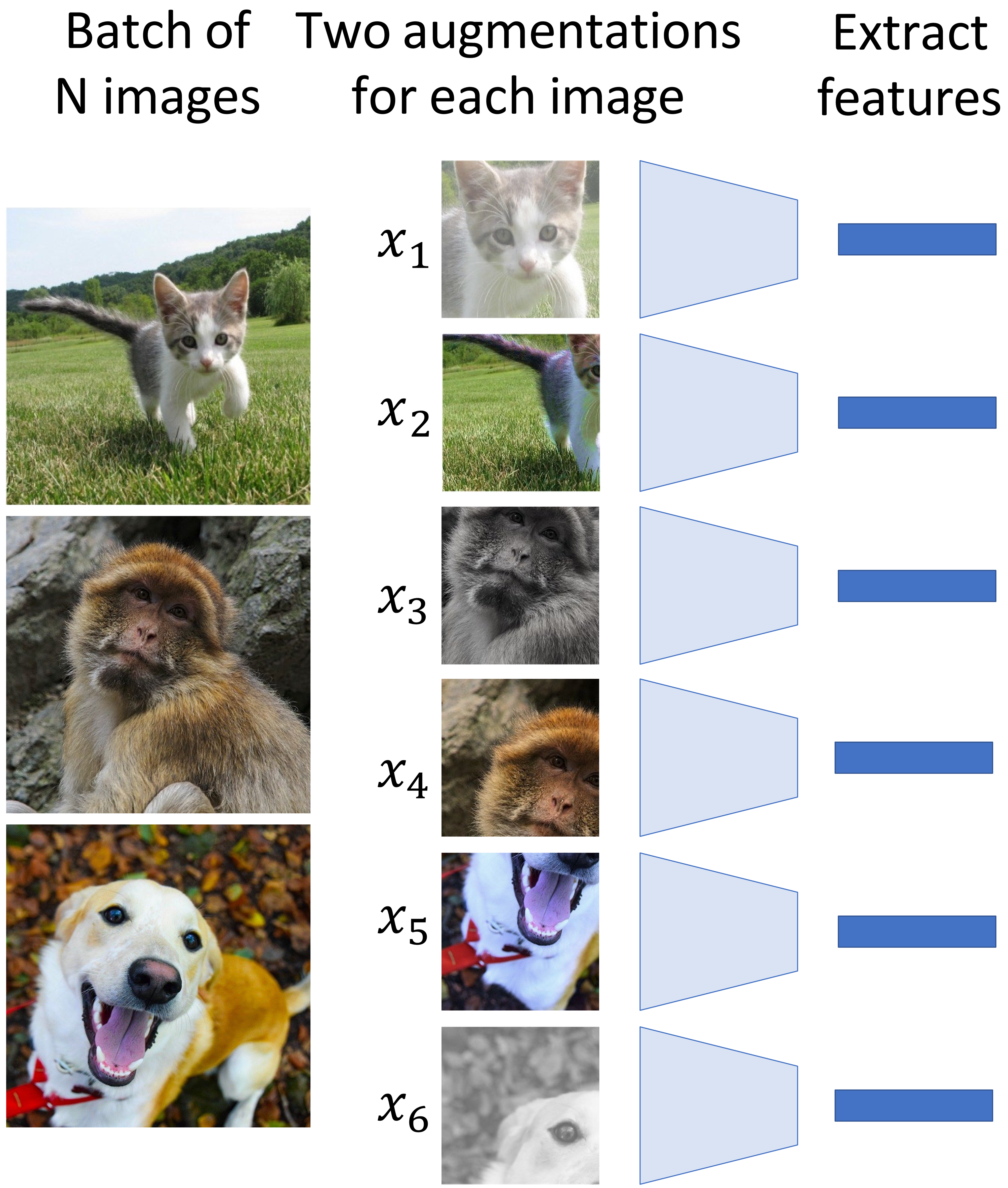
Batch of N images

Two augmentations for each image



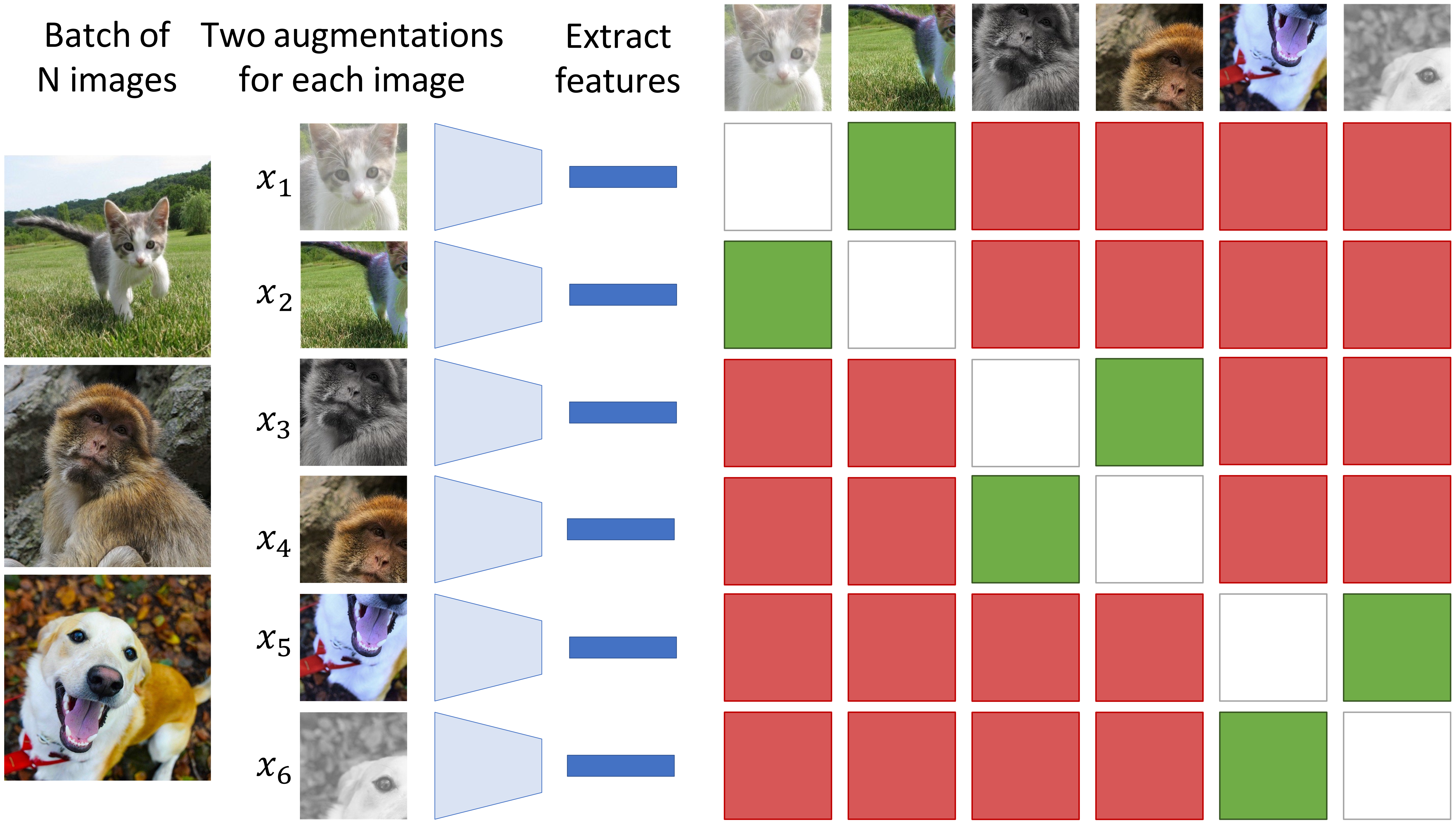
Hadsell et al, "Dimensionality Reduction by Learning and Invariant Mapping", CVPR 2006
Wu et al, "Unsupervised Feature Learning by Non-Parametric Instance-Level Discrimination", CVPR 2018
Van den Oord et al, "Representation Learning with Contrastive Predictive Coding", NeurIPS 2018
Hjelm et al, "Learning deep representations by mutual information estimation and maximization", ICLR 2019
Bachman et al, "Learning Representations by Maximizing Mutual Information Across Views", NeurIPS 2019
Henaff et al, "Data-Efficient Image Recognition with Contrastive Predictive Coding", ICML 2020
Tian et al, "Contrastive Multiview Coding", ECCV 2020
He et al, "Momentum Contrast for Unsupervised Visual Representation Learning", CVPR 2020
Chen et al, "A Simple Framework for Contrastive Learning of Visual Representations", ICML 2020

SimCLR Training



Hadsell et al, "Dimensionality Reduction by Learning and Invariant Mapping", CVPR 2006
Wu et al, "Unsupervised Feature Learning by Non-Parametric Instance-Level Discrimination", CVPR 2018
Van den Oord et al, "Representation Learning with Contrastive Predictive Coding", NeurIPS 2018
Hjelm et al, "Learning deep representations by mutual information estimation and maximization", ICLR 2019
Bachman et al, "Learning Representations by Maximizing Mutual Information Across Views", NeurIPS 2019
Henaff et al, "Data-Efficient Image Recognition with Contrastive Predictive Coding", ICML 2020
Tian et al, "Contrastive Multiview Coding", ECCV 2020
He et al, "Momentum Contrast for Unsupervised Visual Representation Learning", CVPR 2020
Chen et al, "A Simple Framework for Contrastive Learning of Visual Representations", ICML 2020

SimCLR Training



Each image tries to predict which of the *other* 2N-1 images came from the same original image

Similarity between x_i and x_j:

$$s_{i,j} = \frac{\phi(x_i)^T \phi(x_j)}{\|\phi(x_i)\| \cdot \|\phi(x_i)\|}$$

If (x_i, x_j) is a positive pair, then loss for x_i is:

$$L_i = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{\substack{k=1 \\ k \neq i}}^{2N} \exp(s_{i,k}/\tau)}$$

(τ is a *temperature*)

Interpretation: Cross-entropy loss over the other 2N-1 elements in the batch!

Hadsell et al, "Dimensionality Reduction by Learning and Invariant Mapping", CVPR 2006
Wu et al, "Unsupervised Feature Learning by Non-Parametric Instance-Level Discrimination", CVPR 2018
Van den Oord et al, "Representation Learning with Contrastive Predictive Coding", NeurIPS 2018
Hjelm et al, "Learning deep representations by mutual information estimation and maximization", ICLR 2019
Bachman et al, "Learning Representations by Maximizing Mutual Information Across Views", NeurIPS 2019
Henaff et al, "Data-Efficient Image Recognition with Contrastive Predictive Coding", ICML 2020
Tian et al, "Contrastive Multiview Coding", ECCV 2020
He et al, "Momentum Contrast for Unsupervised Visual Representation Learning", CVPR 2020
Chen et al, "A Simple Framework for Contrastive Learning of Visual Representations", ICML 2020

ImageNet Linear Classification from SSL Features

He et al, “Momentum Contrast for Unsupervised Visual Representation Learning”, CVPR 2020
Chen et al, “A Simple Framework for Contrastive Learning of Visual Representations”, ICML 2020
Chen et al, “An Empirical Study of Training Self-Supervised Vision Transformers”, ICCV 2021

(Lots of caveats here … different architectures, etc)

ImageNet Linear Classification from SSL Features

He et al, "Momentum Contrast for Unsupervised Visual Representation Learning", CVPR 2020
Chen et al, "A Simple Framework for Contrastive Learning of Visual Representations", ICML 2020
Chen et al, "An Empirical Study of Training Self-Supervised Vision Transformers", ICCV 2021

(Lots of caveats here … different architectures, etc)

But how did you get the pretraining data?

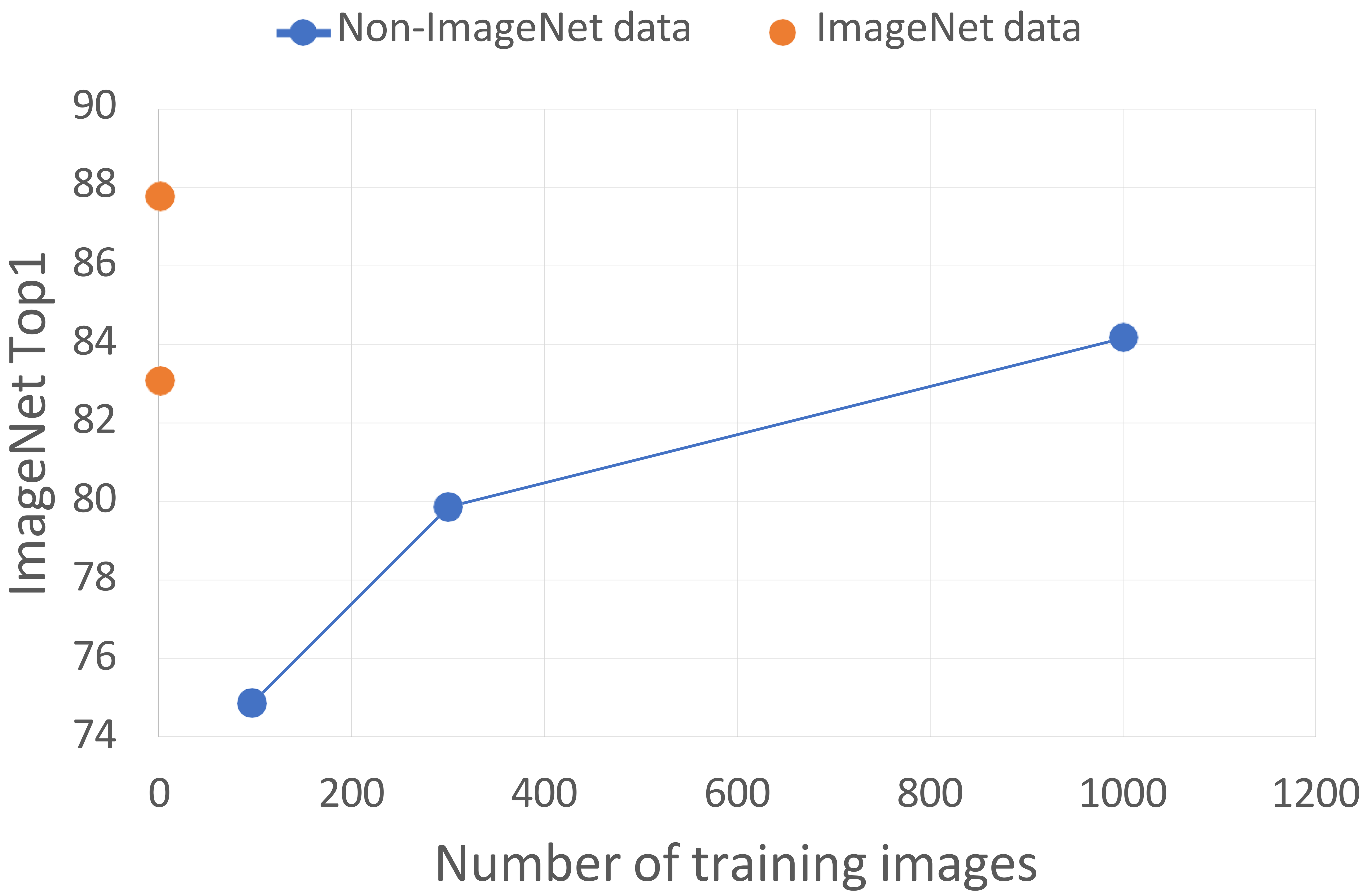
The motivation of SSL is scaling to large data that can't be labeled

Most papers pretrain on (unlabeled) ImageNet, then evaluate on ImageNet!

Unlabeled ImageNet is still curated: single object per image, balanced classes

Self-Supervised Learning on larger datasets hasn't been as successful as NLP

Idea: What if we go beyond isolated images?



Caron et al, "Unsupervised pre-training of images features on non-curated data", ICCV 2019
Chen et al, "Big self-supervised models are strong semi-supervised learners", NeurIPS 2020
Dosovitskiy et al, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ICLR 2021
Goyal et al, "Self-supervised Pretraining of Visual Features in the Wild", arXiv 2021
He et al, "Masked Autoencoders are Scalable Vision Learners", arXiv 2021

Multimodal Self-Supervised Learning

Don't learn from isolated images -- take images together with some **context**

Video: Image together with adjacent video frames

Agrawal et al, "Learning to See by Moving", ICCV 2015
Wang et al, "Unsupervised Learning of Visual Representations using Videos", ICCV 2015
Pathak et al, "Learning Features by Watching Objects Move", CVPR 2017

Sound: Image with audio track from video

Owens et al, "Ambient Sound Provides Supervision for Visual Learning", ECCV 2016
Arandjelovic and Zisserman, "Look, Listen and Learn", ICCV 2017

3D: Image with depth map or point cloud

Xie et al, "PointContrast: Unsupervised Pre-training for 3D Point Cloud Understanding", ECCV 2020
Zhang et al, "Self-supervised pretraining of 3D features on any point-cloud", CVPR 2021

Language: Image with natural-language text

Sariyildiz et al, "Learning Visual Representations with Caption Annotations", ECCV 2020
Desai and Johnson, "VirTex: Learning Visual Representations from Textual Annotations", CVPR 2021
Radford et al, "Learning Transferable Visual Models form Natural Language Supervision", ICML 2021
Jia et al, "Scaling up Visual and Vision-Language Representation Learning with Noisy Text Supervision", ICLR 2021
Desai et al, "RedCaps: Web-curated Image-Text data created by the people, for the people", NeurIPS 2021

Next time: Multimodal (Self-Supervised) Learning