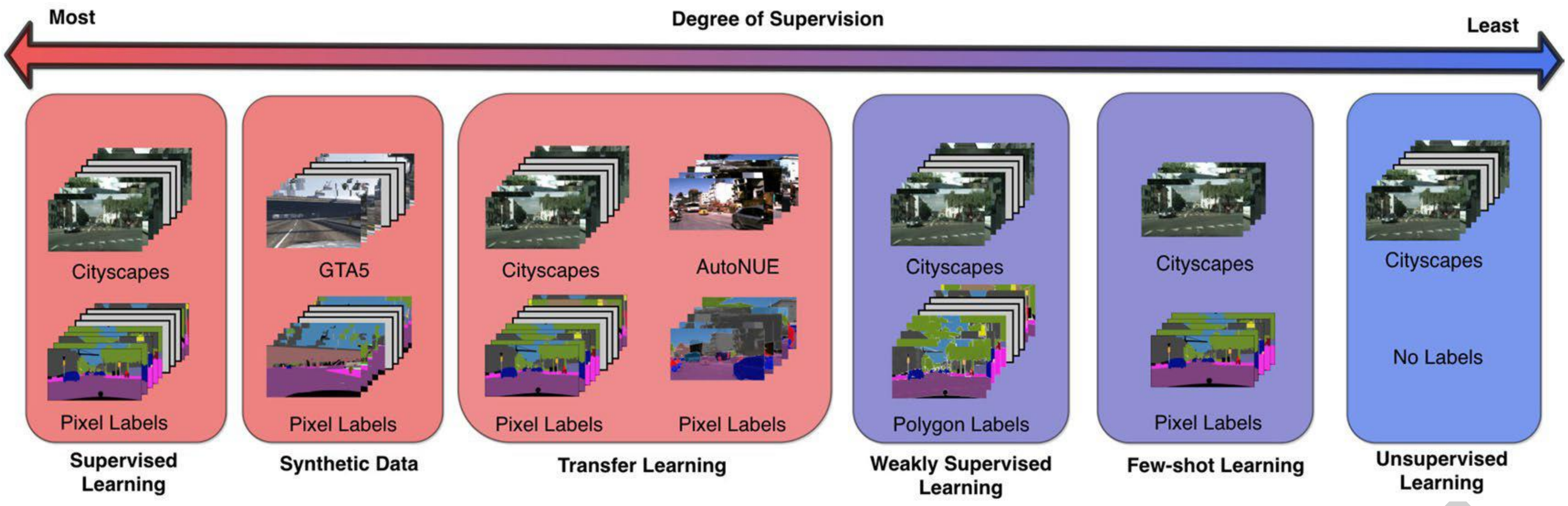
Self Supervised Learning

Space of Supervision



What is Self-Supervised Learning?

Self-Supervised Learning (SSL) is a special type of representation learning that enables learning good data representation from unlabelled dataset.

It is motivated by the idea of *constructing supervised learning tasks out of unsupervised datasets.* **Why?**

1. Data labeling is expensive and thus high-quality labeled dataset is limited.
2. Learning good representation makes it easier to transfer useful information to a variety of downstream tasks.
    - e.g. A downstream task has only a few examples.
    - e.g. Zero-shot transfer to new tasks.

Unsupervised Learning

“We expect unsupervised learning to become far more important in the longer term. Human and animal learning is largely unsupervised: we discover the structure of the world by observing it, not by being told the name of every object.”

- LeCun, Bengio, Hinton, Nature 2015

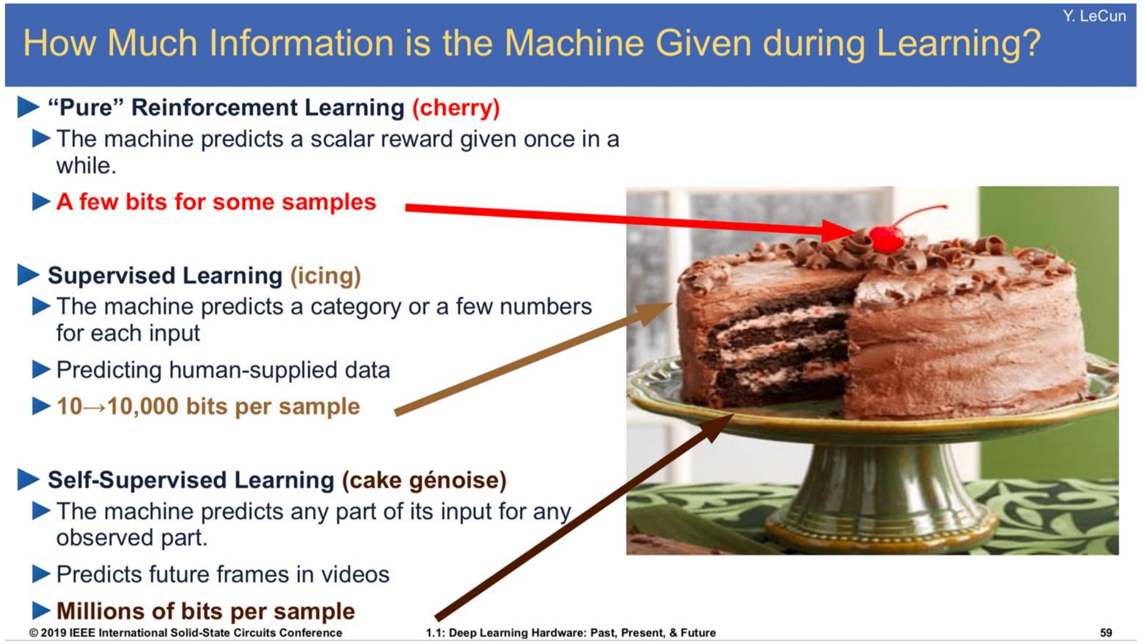
As I've said in previous statements: most of human and animal learning is unsupervised learning. If intelligence was a cake, unsupervised learning would be the cake, supervised learning would be the icing on the cake, and reinforcement learning would be the cherry on the cake. We know how to make the icing and the cherry, but we don't know how to make the cake.

- Yann LeCun, March 14, 2016 (Facebook)

Old and New

2016

2019



Source: Y. LeCun at NIPS 2016

“The Cake of Learning”

downstream tasks

feature extractor

Learn good features through self-supervision

How Much Information is the Machine Given during Learning?

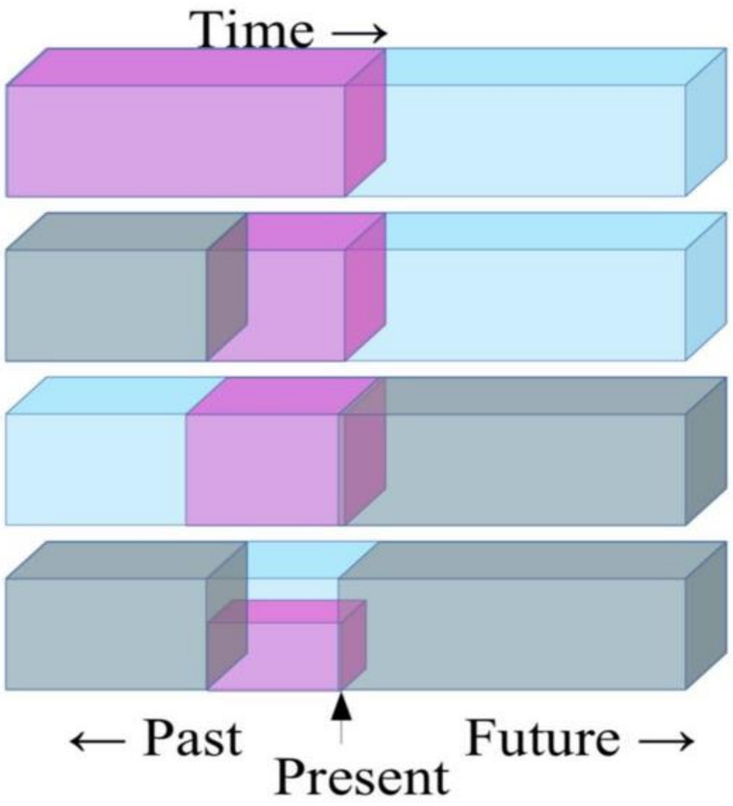
Y. LeCun

- **“Pure” Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

Self-Supervised Learning

General idea: pretend there is a part of the data you don't know and train the neural network to predict that.

- Predict any part of the input from any other part.
- Predict the future from the past.
- Predict the future from the recent past.
- Predict the past from the present.
- Predict the top from the bottom.
- Predict the occluded from the visible
- Pretend there is a part of the input you don't know and predict that.



 1.1: Deep Learning Hardware: Past, Present, & Future 58

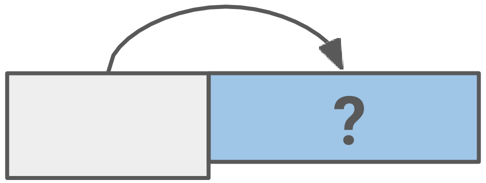
Two Popular Ways

- Self-prediction
- Contrastive learning

Methods for Framing Self-Supervised Learning Tasks

Self-prediction: Given an individual data sample, the task is to predict one part of the sample given the other part.

The part to be predicted pretends to be missing.

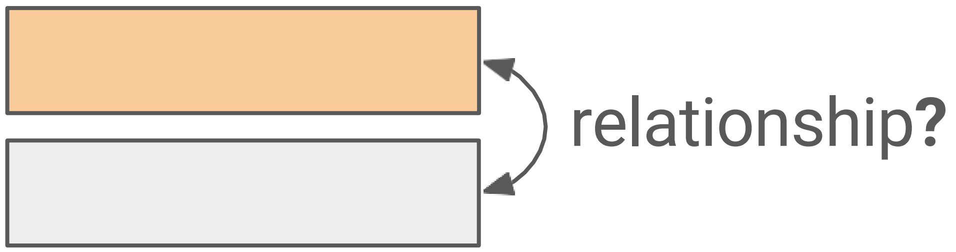


“Intra-sample” prediction

Methods for Framing Self-Supervised Learning Tasks

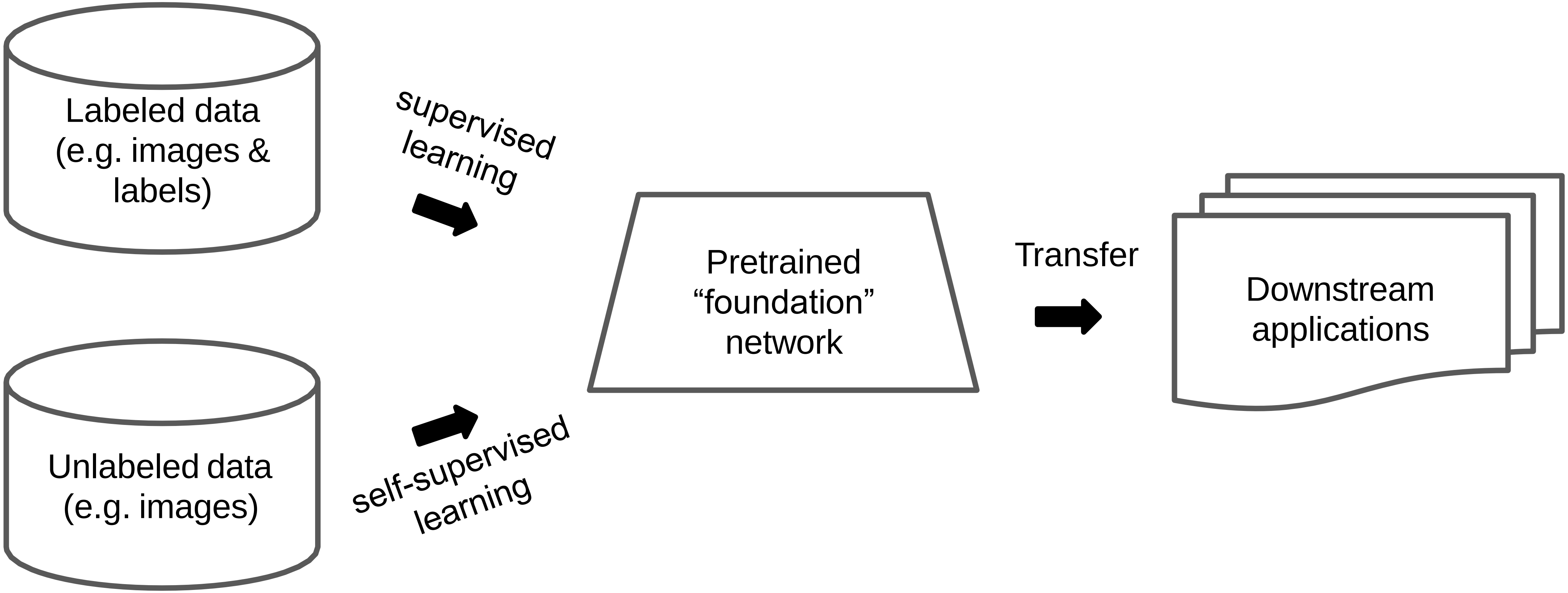
Contrastive learning: Given multiple data samples, the task is to predict the relationship among them.

The multiple samples can be selected from the dataset based on some known logics (e.g. the order of words / sentences) or fabricated by altering the original version.



"Inter-sample" prediction

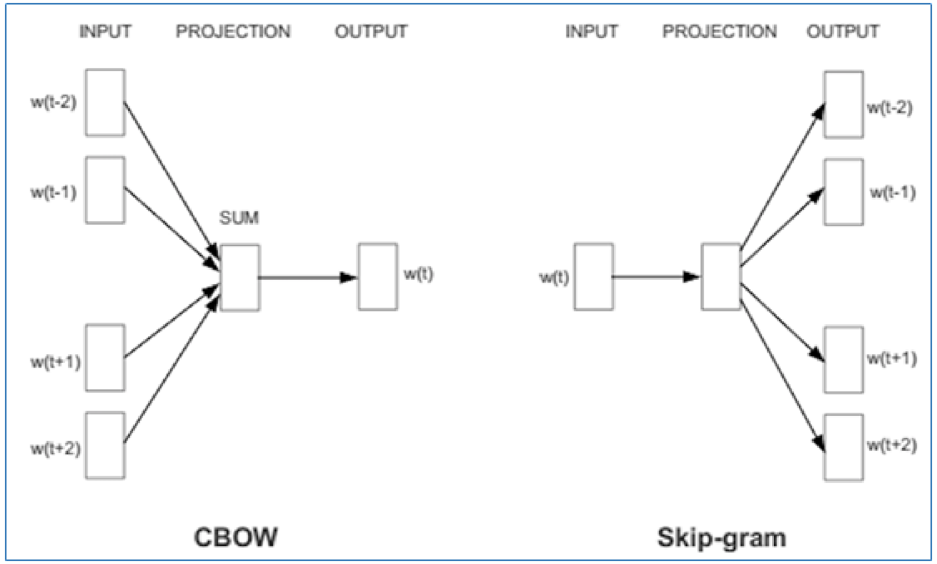
The paradigm of learning “foundation” models

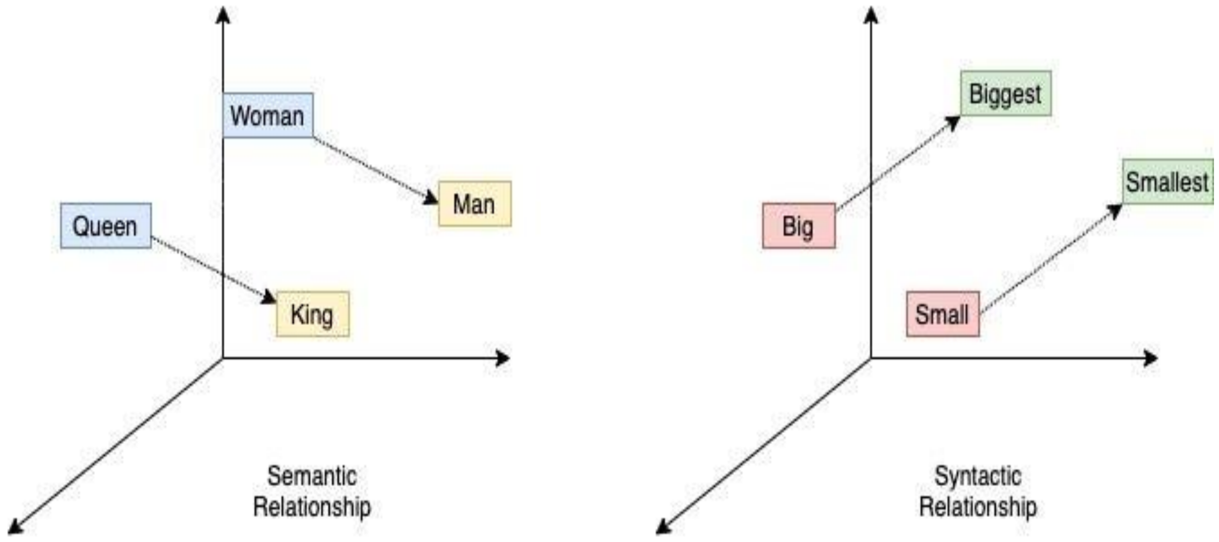


“Self-supervised learning” is “supervised learning” without specific task annotations.

Pre-Text Tasks

Word2Vec: Word Embedding





Pretext Task of "Filling the Blank"

Training with no human supervision (Self-Supervised)

King – Queen = Man - Woman

"Semantic" manipulation of text/words

Examples from NLP

Self-supervised learning has driven the recent progress in the ***Natural Language Processing*** (NLP) field

- Models like ELMO, BERT, RoBERTa, ALBERT, Turing NLG, GPT-3 have demonstrated immense potential for automated NLP

Employing various pretext tasks for leaning from raw text produced rich feature representations, useful for different downstream tasks

Pretext tasks in NLP:

- Predict the center word given a window of surrounding words
  - The word highlighted with green color needs to be predicted

- Predict the surrounding words given the center word

Examples from NLP

Pretext tasks in NLP:

- From three consecutive sentences, predict the previous and the next sentence, given the center sentence

Previous sentence	Iron man fails to lift Thor's hammer
Center Sentence	Captain America tries lifting Thor's hammer
Next Sentence	The hammer moves a bit

predict

- Predict the previous or the next word, given surrounding words

_______ is impossible

↓

Nothing is impossible

Right-to-left prediction

- Predict randomly masked words in sentences

Randomly masked: A quick [MASK] fox jumps over the [MASK] dog

↓ ↓

Predict: A quick brown fox jumps over the lazy dog

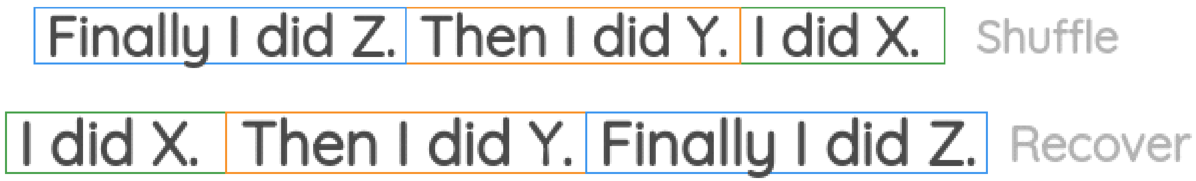
Examples from NLP

Pretext tasks in NLP:

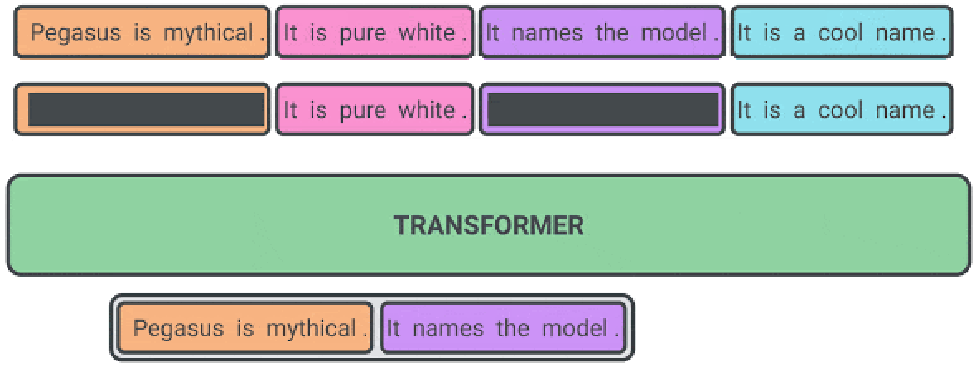
- Predict if the ordering of two sentences is correct

Sentence 1	Sentence 2	Next Sentence
I am going outside	I will be back in the evening	yes
I am going outside	You know nothing John Snow	no

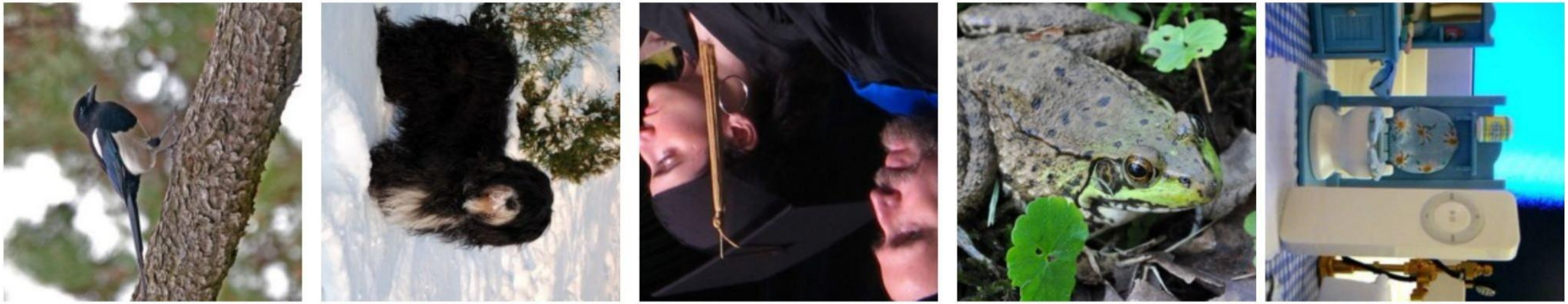
- Predict the order of words in a randomly shuffled sentence



- Predict masked sentences in a document



Pretext task: predict rotations

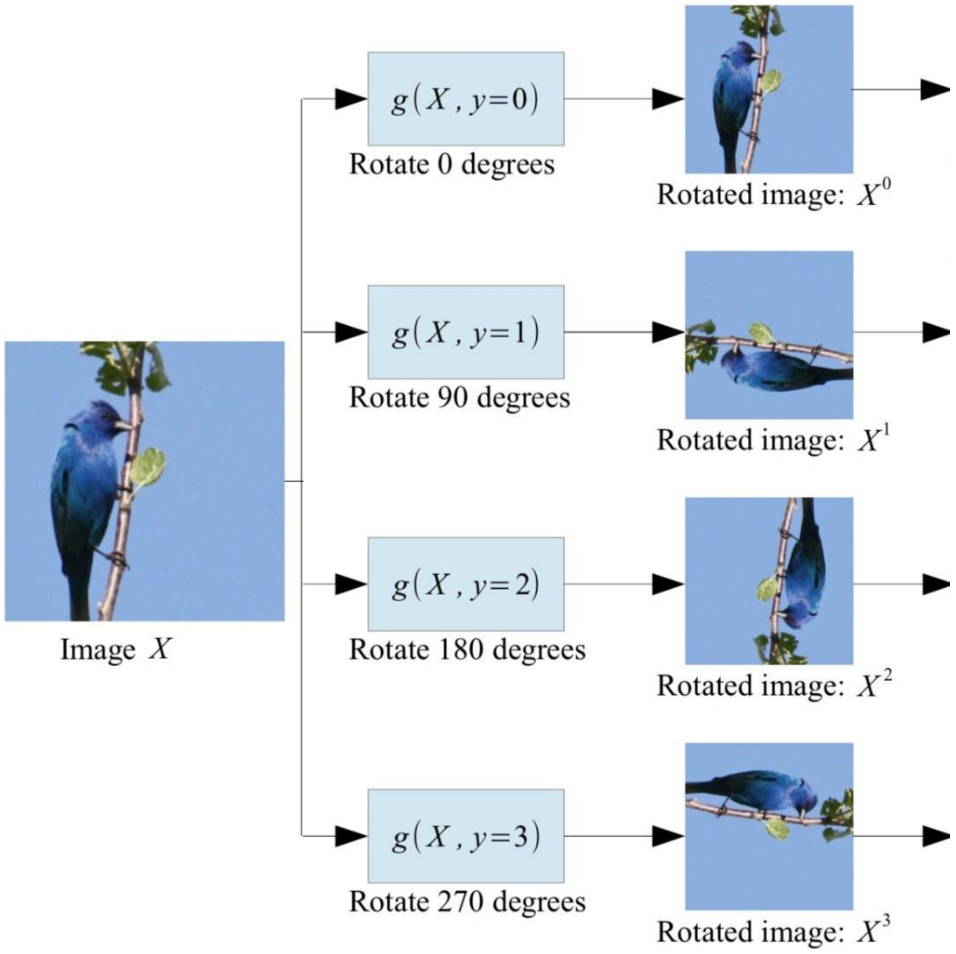


90° rotation | 270° rotation | 180° rotation | 0° rotation | 270° rotation

Hypothesis: a model could recognize the correct rotation of an object only if it has the “visual commonsense” of what the object should look like unperturbed.

(Image source: Gidaris et al. 2018)

Pretext task: predict rotations

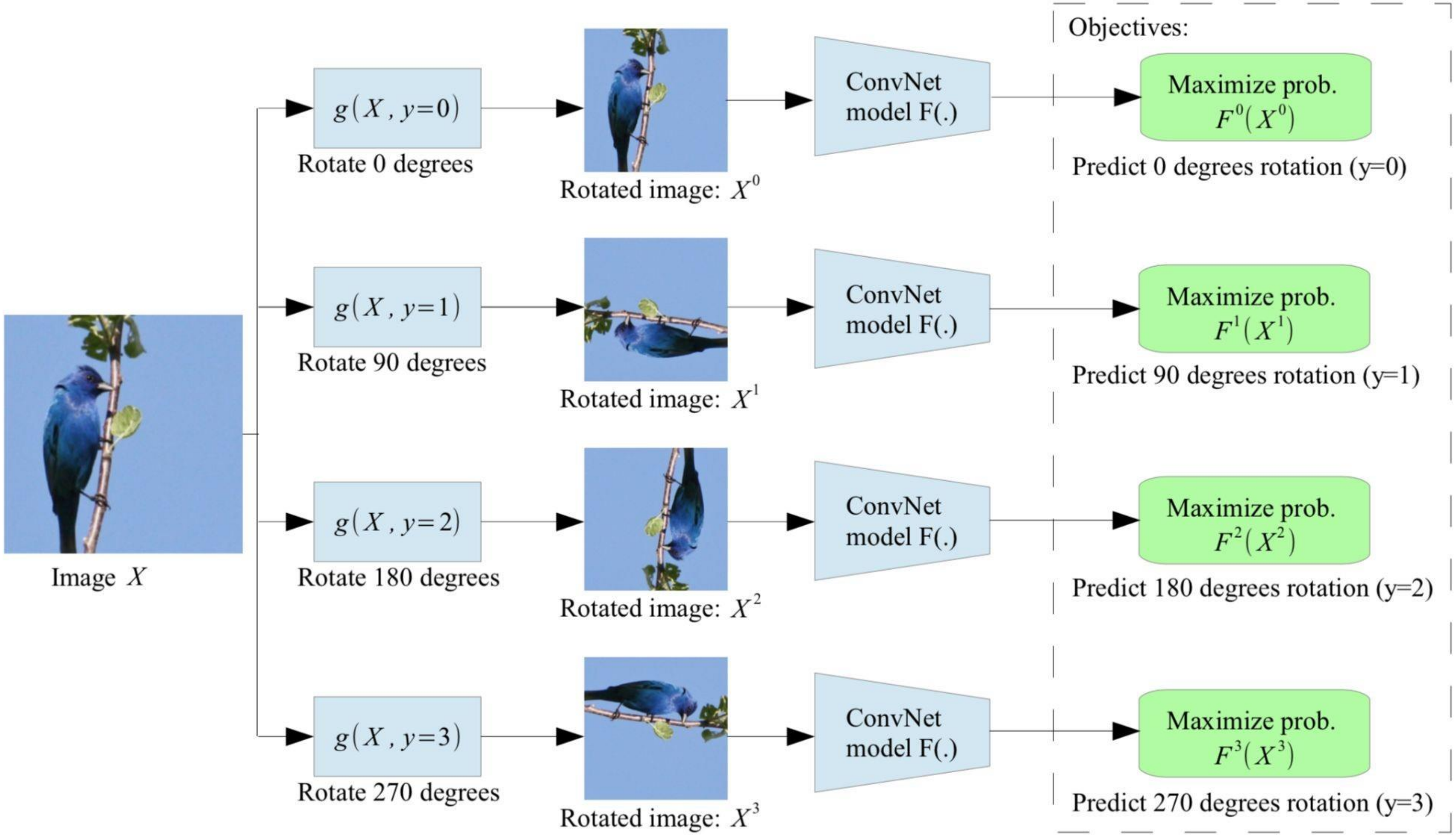


Self-supervised learning by rotating the entire input images.

The model learns to predict which rotation is applied (4-way classification)

(Image source: [Gidaris et al. 2018](#))

Pretext task: predict rotations

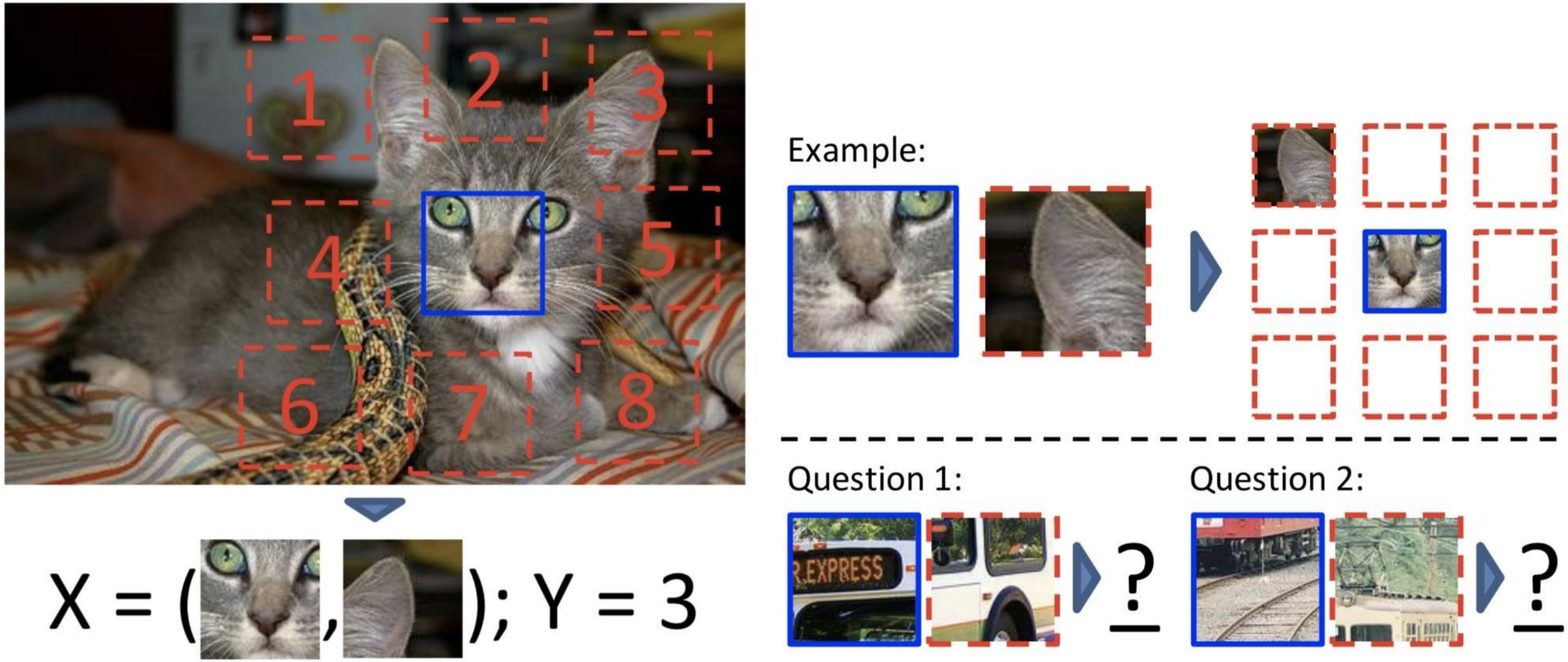


Self-supervised learning by rotating the entire input images.

The model learns to predict which rotation is applied (4-way classification)

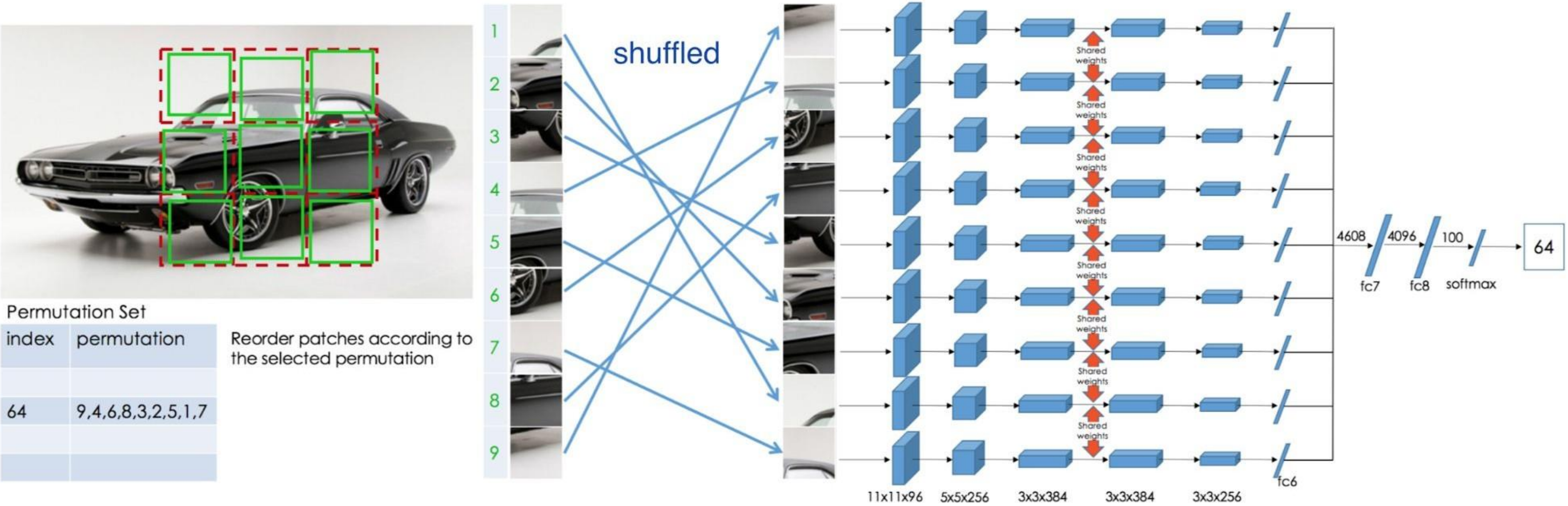
(Image source: Gidaris et al. 2018)

Pretext task: predict relative patch locations



(Image source: Doersch et al., 2015)

Pretext task: solving “jigsaw puzzles”

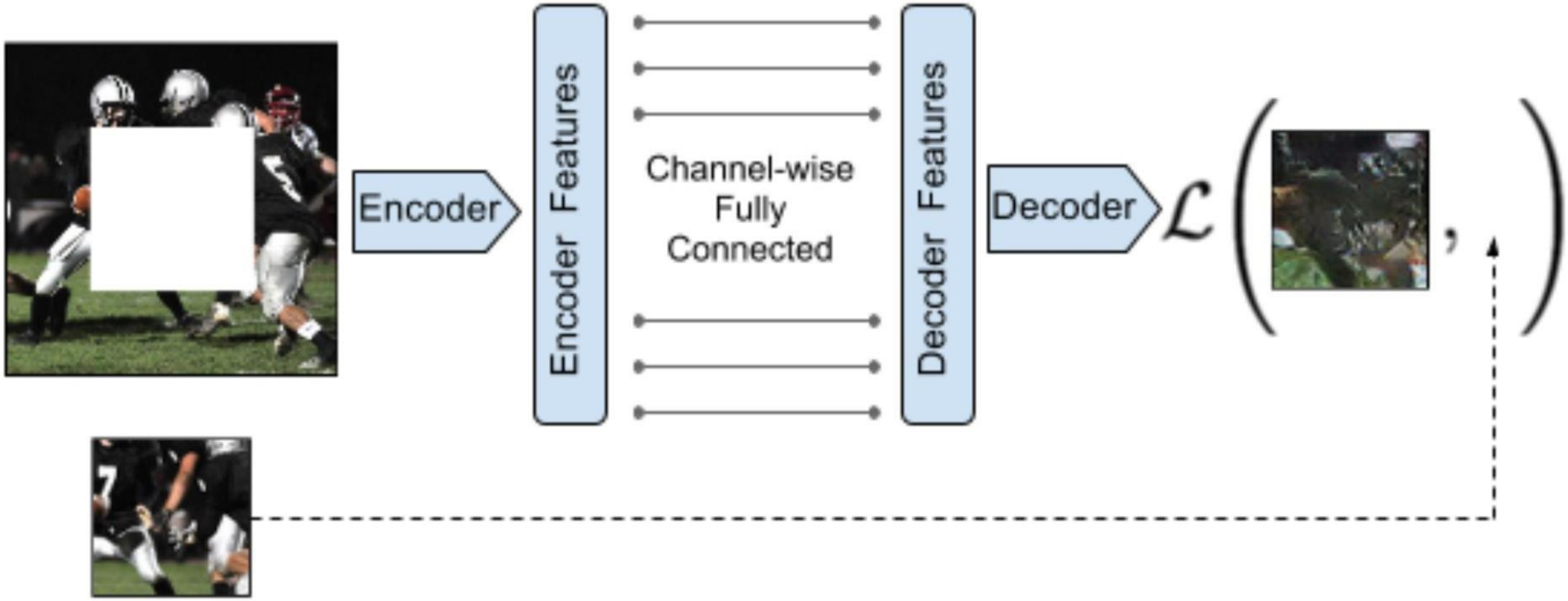


(Image source: [Noroozi & Favaro, 2016](#))

Pretext task: predict missing pixels (inpainting)

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

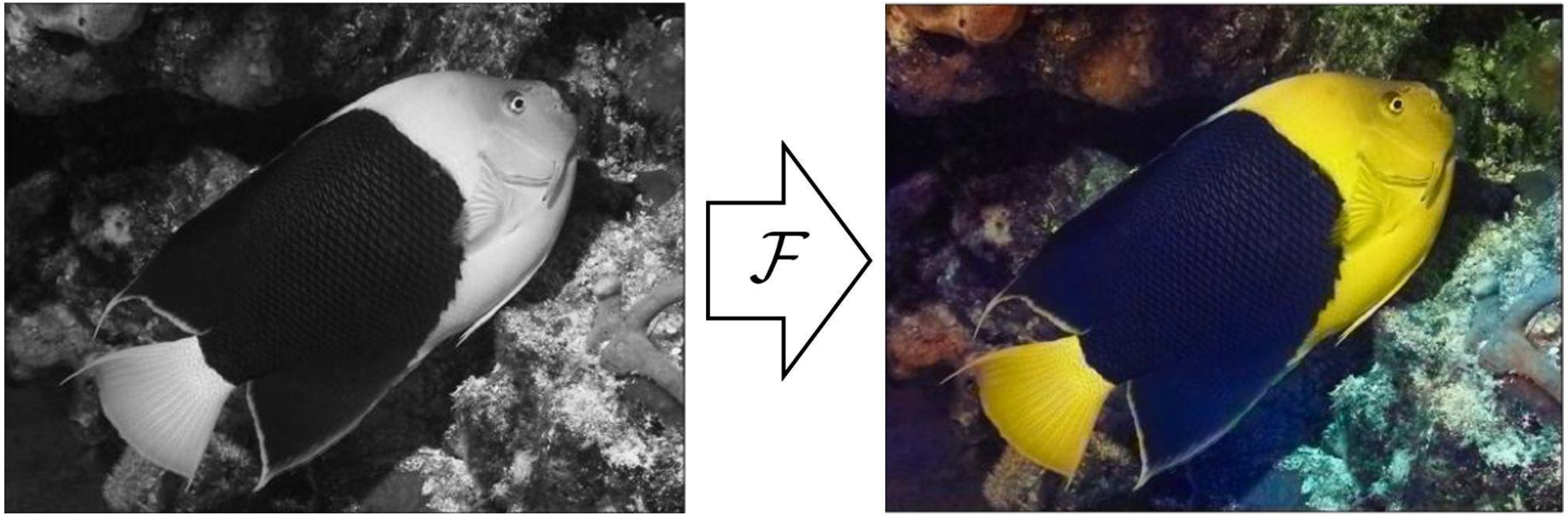
Learning to inpaint by reconstruction



Learning to reconstruct the missing pixels

Source: [Pathak et al., 2016]

Pretext task: image coloring

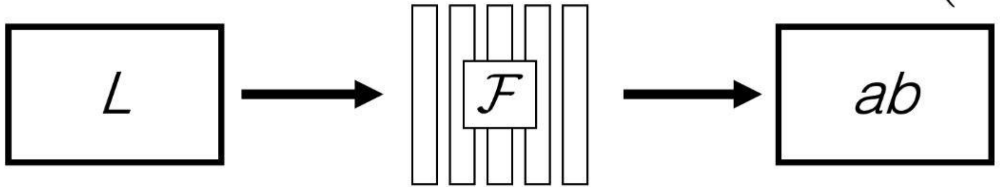


Grayscale image: *L* channel

$$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$$

Concatenate (*L*,*ab*) channels

$$(\mathbf{X}, \widehat{\mathbf{Y}})$$



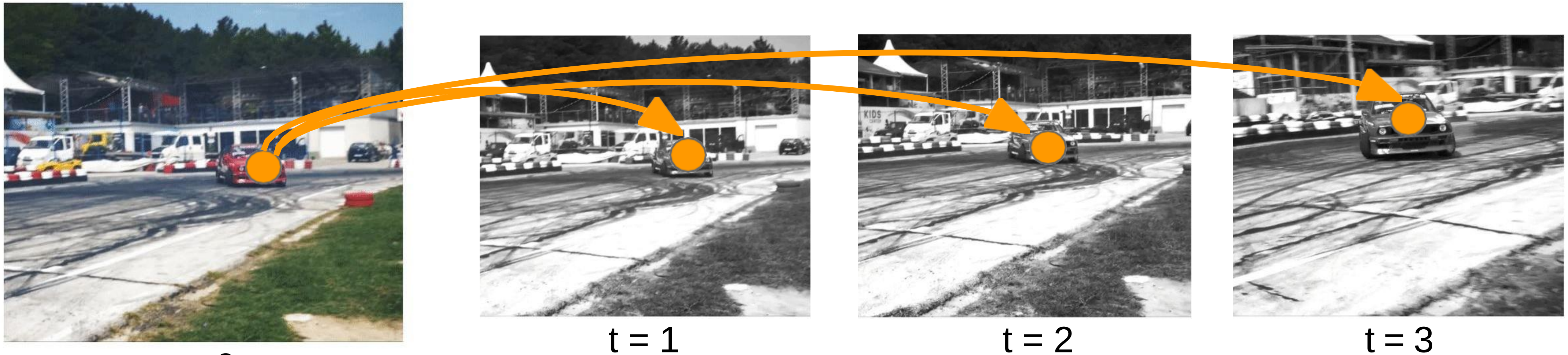
Source: Richard Zhang / Phillip Isola

Pretext task: video coloring

Idea: model the *temporal coherence* of colors in videos

reference frame

how should I color these frames?



t = 0

t = 1

t = 2

t = 3

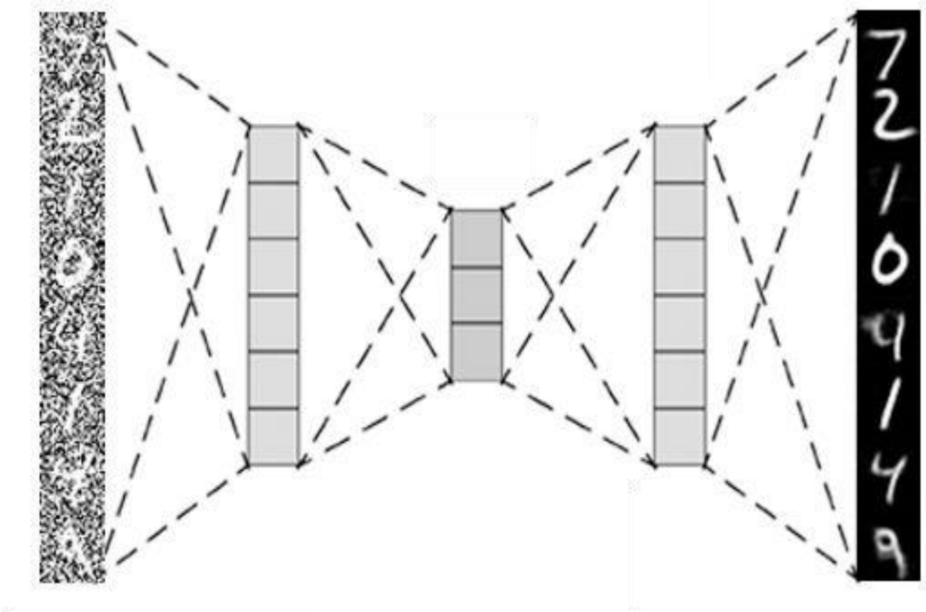
…

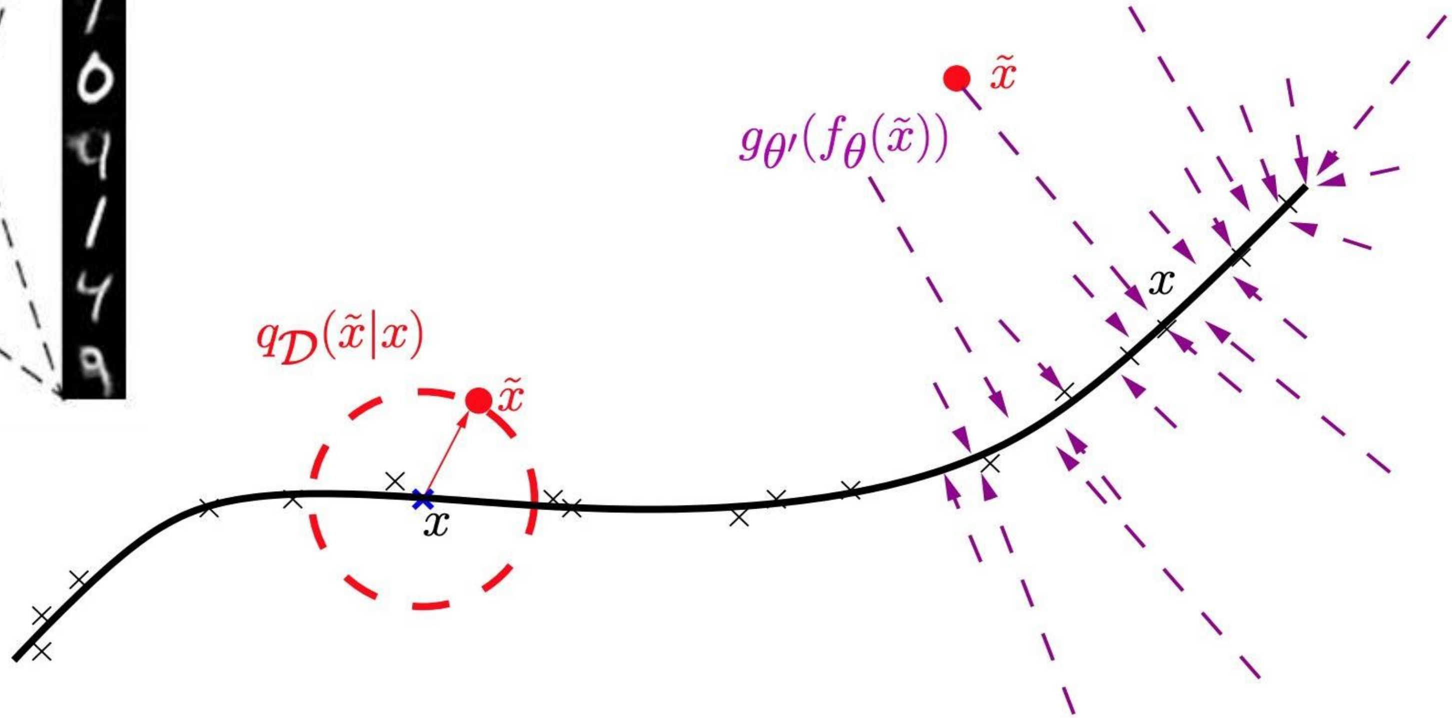
Hypothesis: learning to color video frames should allow model to learn to track regions or objects without labels!

Source: [Vondrick et al., 2018]

Autoencoder: Self-Supervised Learning-Vision in Early Days

Denoising Autoencoder (Vincent et al. 2008)





(Vincent et al. 2008)

Pretext tasks from image transformations

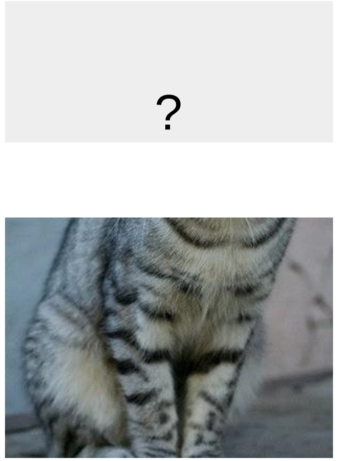
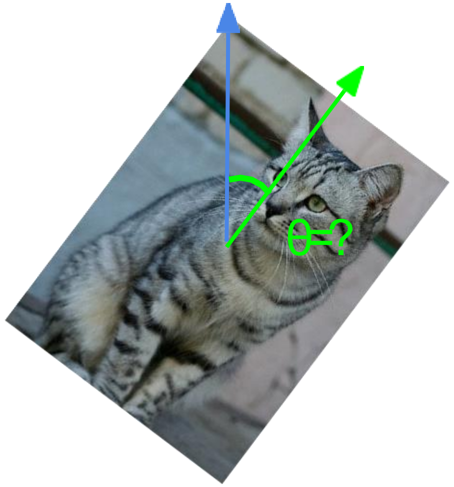


image completion



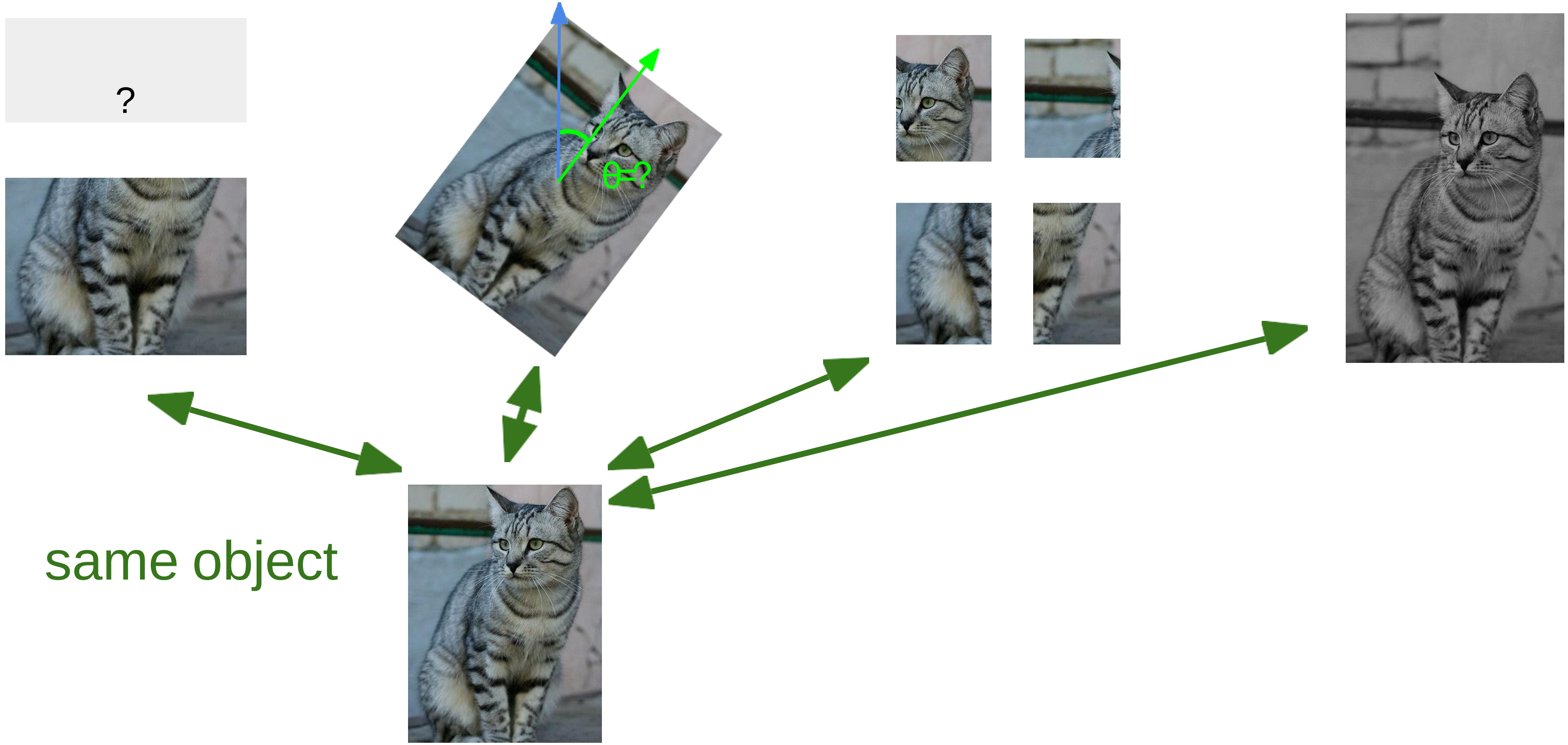
rotation prediction

“jigsaw puzzle”

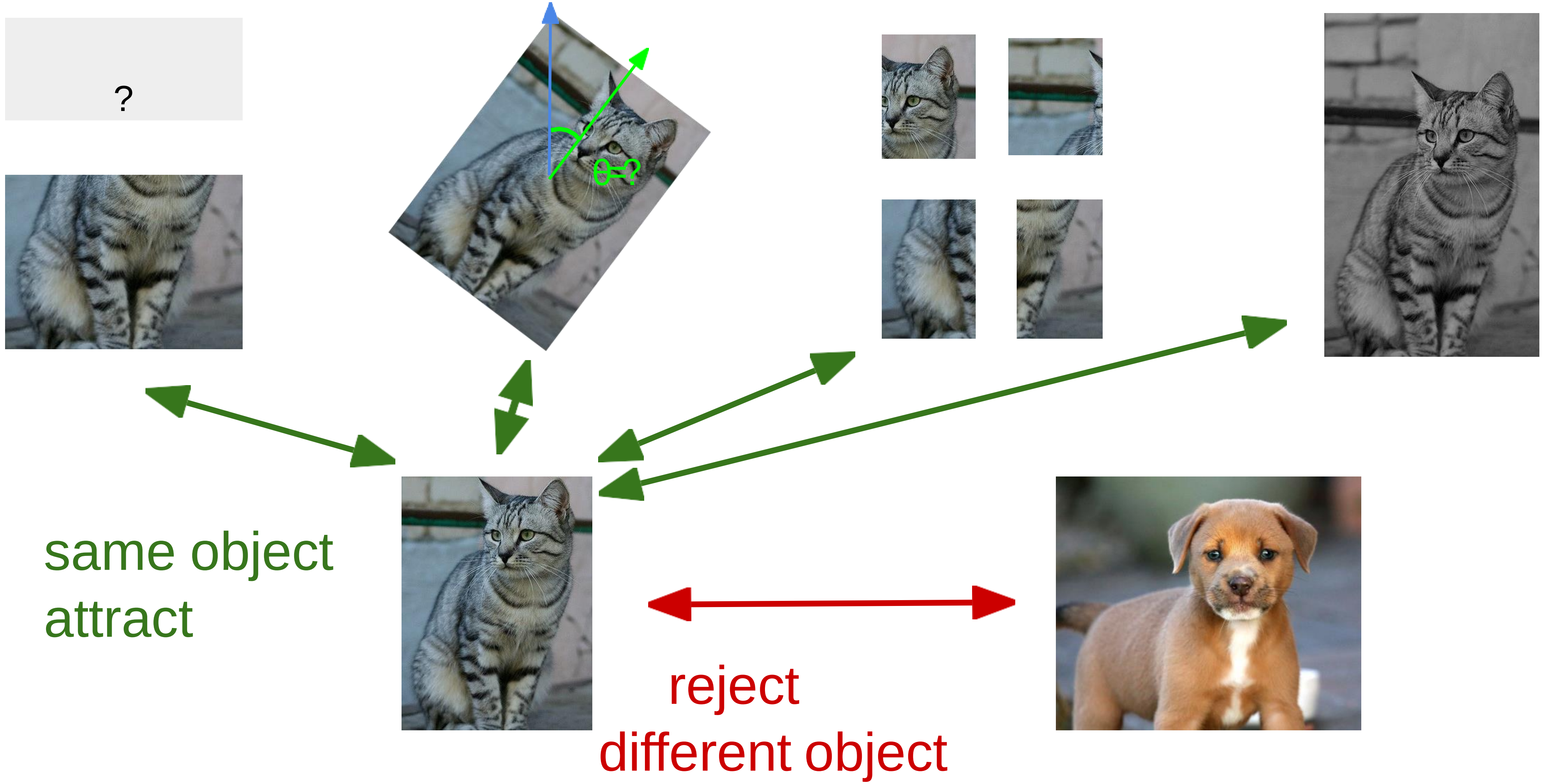
colorization

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

We know samples are same

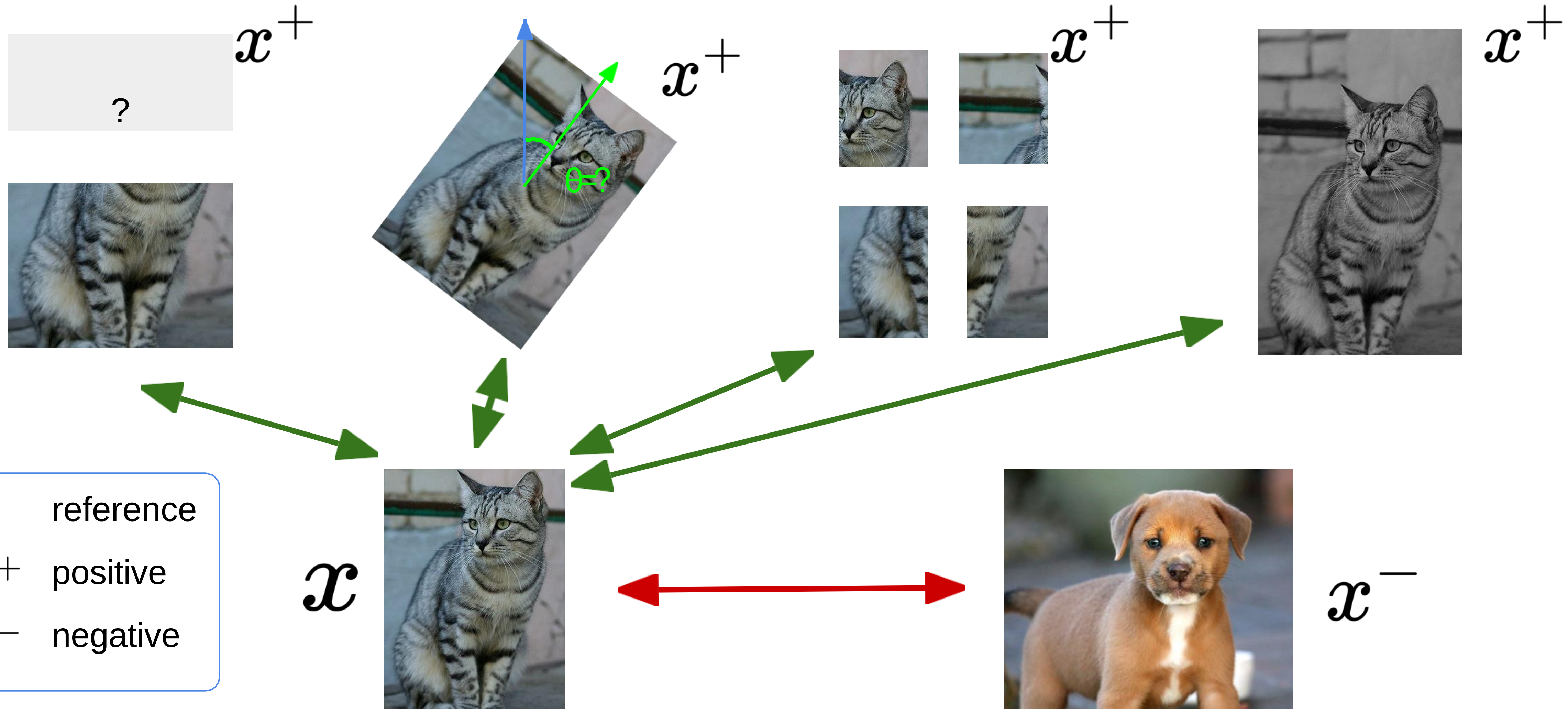


We know samples are same and different



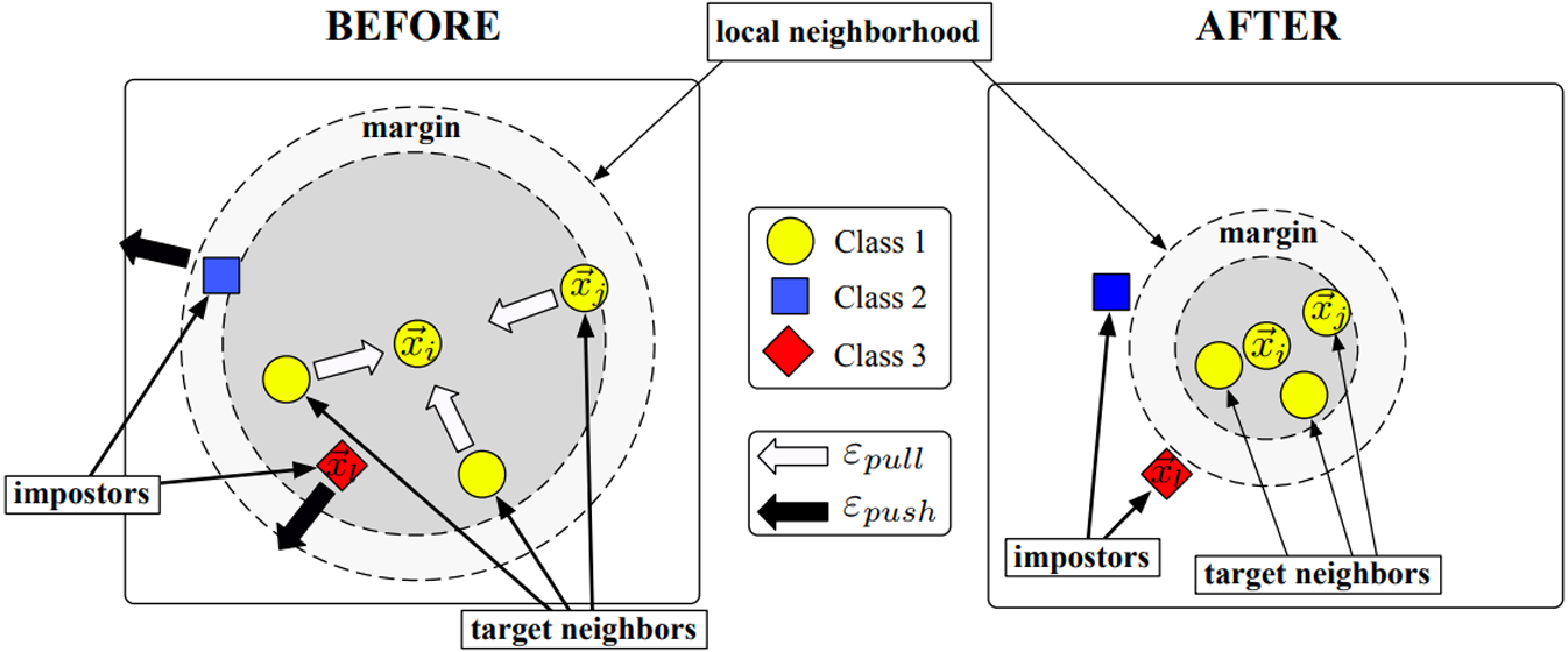
Contrastive Leaning

Contrastive Representation Learning



x	reference
x^+	positive
x^-	negative

Large Margin Nearest Neighbour(LMNN)



Weinberger and Saul, JMLR 2009

Metric Learning

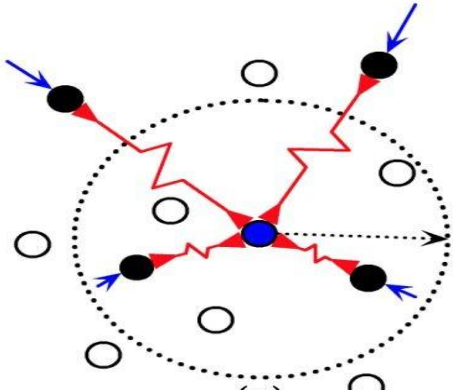
Metric learning (Xing et al. 2002)

$$d_A(x, y) = ||x - y||_A = \sqrt{(x-y)^T A (x-y)}$$

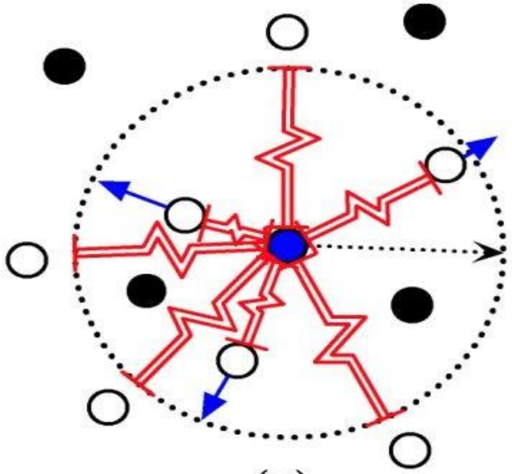
Contrastive Loss (Chopra & Hadsell et al. 2005)

i. If $Y_{ij} = 0$, then update W to decrease $D_W = \|G_W(\vec{X}_i) - G_W(\vec{X}_j)\|_2$

ii. If $Y_{ij} = 1$, then update W to increase $D_W = \|G_W(\vec{X}_i) - G_W(\vec{X}_j)\|_2$

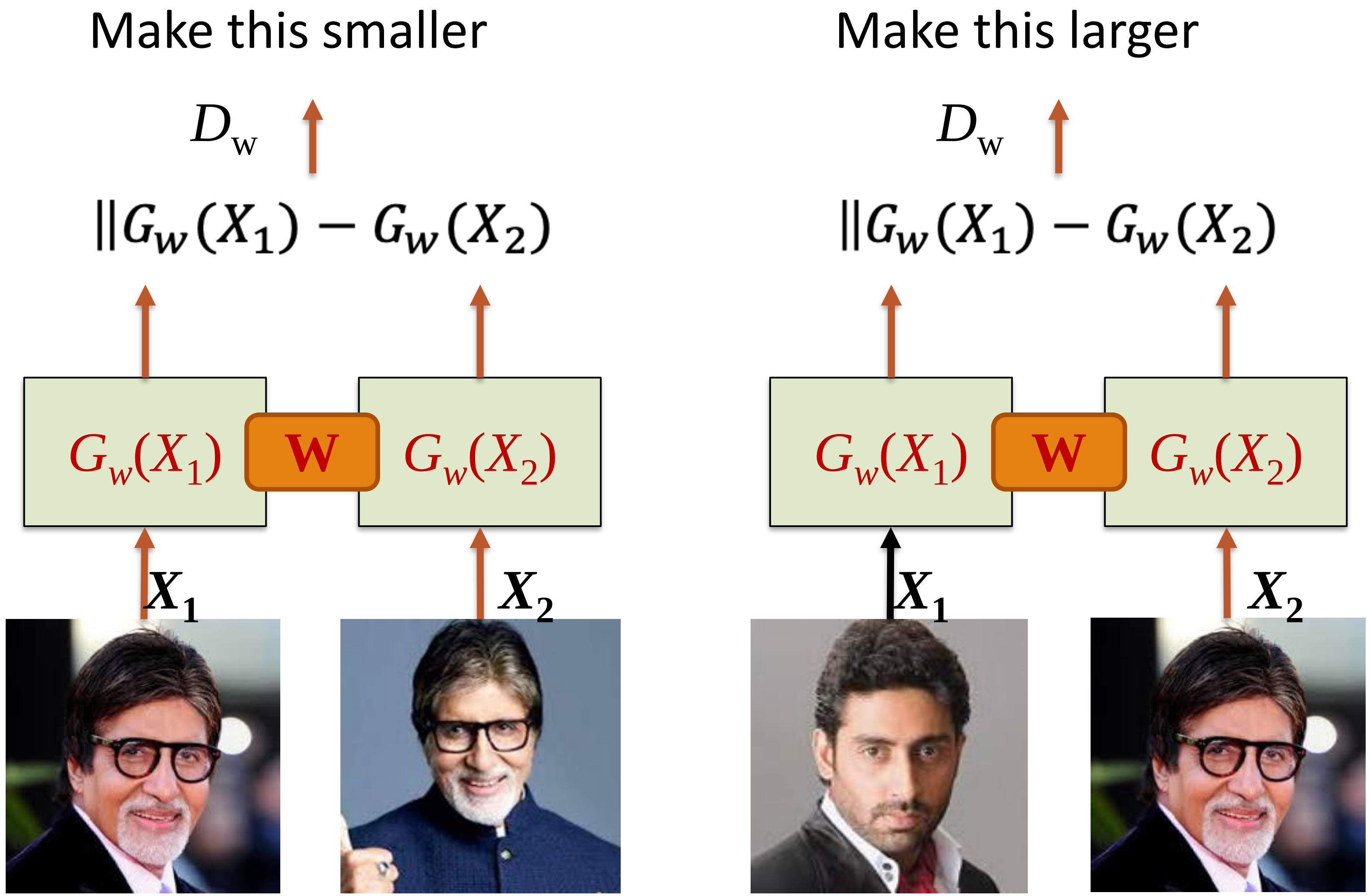


(a)



(c)

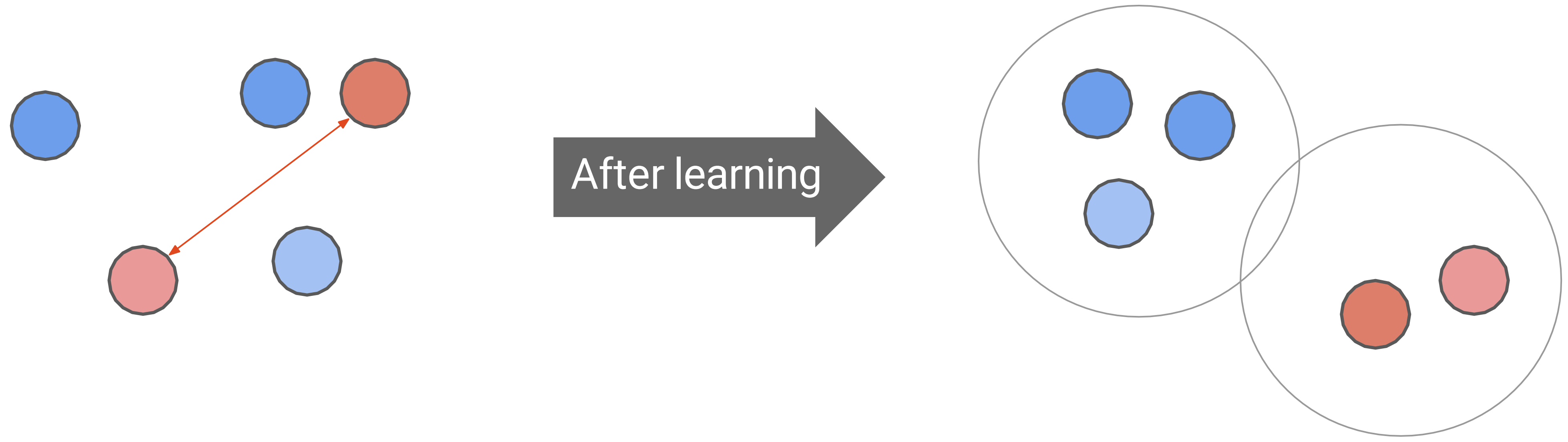
Siamese Architecture/Loss



- Only pair-wise Labels
- Similarity Metric:
  $D_w(X_1, X_2)$
- Have shared weights
- Training in batches

Contrastive Learning

The goal of contrastive representation learning is to learn such an embedding space in which *similar* sample pairs stay *close* to each other while *dissimilar* ones are *far apart*.



Contrastive Learning: Inter-Sample Classification

The goal of contrastive representation learning is to learn such an embedding space in which *similar* sample pairs stay *close* to each other while *dissimilar* ones are *far apart*.

Given both similar (“positive”) and dissimilar (“negative”) candidates, to identify which ones are similar to the anchor data point is a classification task.

There are creative ways to construct a set of data point candidates:

- ❑The original input and its distorted version
- ❑Data that captures the same target from different views

Contrastive Learning: Inter-Sample Classification

Common loss functions:

- Contrastive loss (Chopra et al. 2005)
- Triplet loss (Schroff et al. 2015; FaceNet)
- Lifted structured loss (Song et al. 2015)
- Multi-class n-pair loss (Sohn 2016)
- Noise contrastive estimation ("NCE"; Gutmann & Hyvarinen 2010)
- InfoNCE (van den Oord, et al. 2018)
- Soft-nearest neighbors loss (Salakhutdinov & Hinton 2007, Frosst et al. 2019)

Contrastive Learning: Inter-Sample Classification

Contrastive loss (Chopra et al. 2005): Works with labelled dataset.

Encodes data into an embedding vector such that examples from the same class have similar embeddings and samples from different classes have different ones.

Given two labeled data pairs $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$:

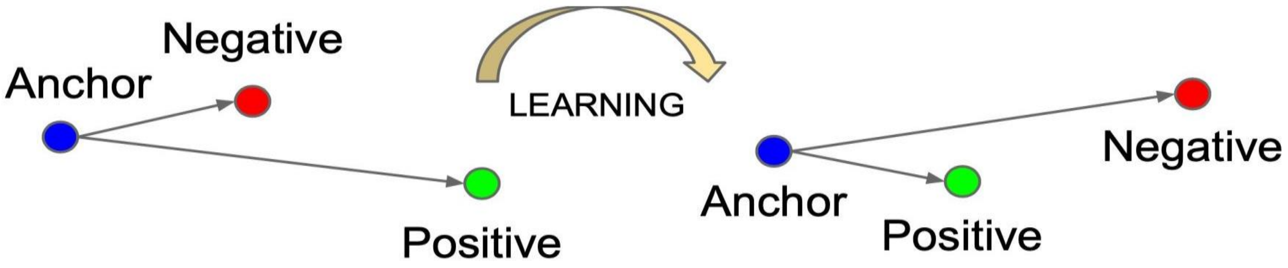
$$\mathcal{L}_{\text{cont}}(\mathbf{x}_i, \mathbf{x}_j, \theta) = \mathbb{1}[y_i = y_j] \underbrace{\boxed{\|f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}_j)\|_2^2}}_{\text{minimize}} + \mathbb{1}[y_i \neq y_j] \max(0, \epsilon - \underbrace{\boxed{\|f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}_j)\|_2}}_{\text{maximize}})^2$$

Contrastive Learning: Inter-Sample Classification

Triplet loss (Schroff et al. 2015): learns to minimize the distance between the anchor **x** and positive **x+** and maximize the distance between the anchor **x** and negative **x-** at the same time.

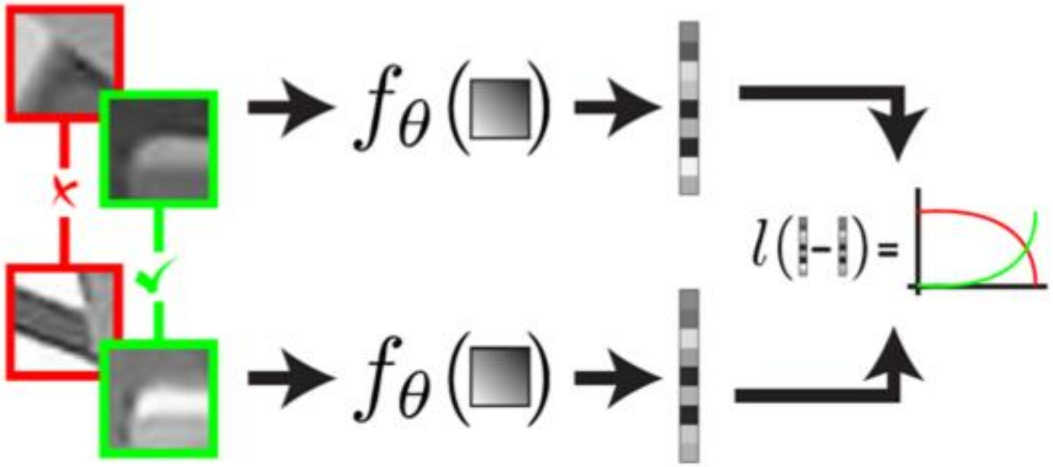
Given a triplet input $(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-)$,

$$\mathcal{L}_{\text{triplet}}(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-) = \sum_{\mathbf{x} \in \mathcal{X}} \max\left(0, \|f(\mathbf{x}) - f(\mathbf{x}^+)\|_2^2 - \|f(\mathbf{x}) - f(\mathbf{x}^-)\|_2^2 + \epsilon\right)$$



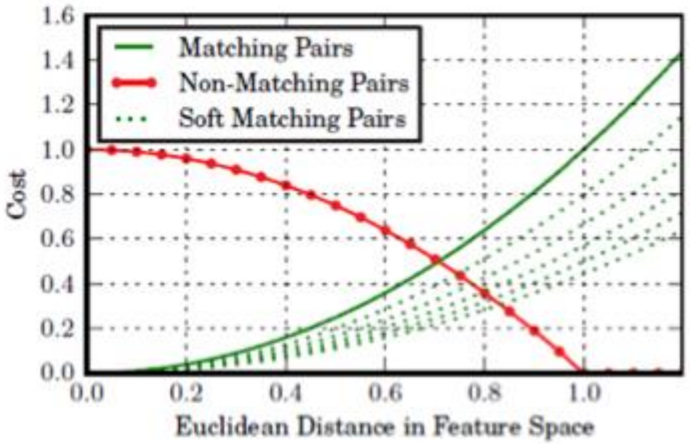
(Schroff et al. 2015)

Application: Learning to Match Siamese Network



Using the contrastive cost function

$$l_{\boldsymbol{\theta}}\left(\mathbf{y}_i, \mathbf{y}_j\right) = \begin{cases} s_{ij} d_{ij}^2, & \text{if matching} \\ \max\left(1.0 - d_{ij}^2,\ 0\right), & \text{if non-matching} \end{cases}$$



IROS 2014

Person identification

CUHK03 Data set

True positive

True negative

Ahmed, E., Jones, M. and Marks, T.K., 2015. An improved deep learning architecture for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3908-3916).

Summary

- Fully Supervised Learning is not practical in all situations
  - Availability of the annotation
  - Knowledge of the task ahead of time
- Self Supervised Learning provided an “unsupervised” learning
  - Pretraining
- Two main ways
  - Pretext Tasks
  - Contrastive Learning
- Success stories
  - Many in language
  - Impressive results in vision competing with Fully Supervised

Thanks!!

Questions?