Lecture 12

Representation Learning

Sara Beery, Kaiming He, Mina Konaković Luković, Vincent Sitzmann

Overview

- Why should we care about Representation Learning?
- Transfer learning
- Unsupervised learning
- Self-supervised learning

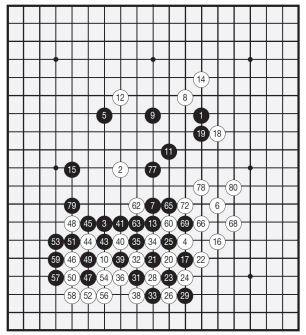
Why should we care about Representation Learning?

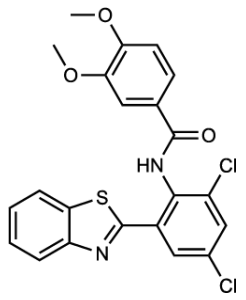
Deep Learning is Representation Learning

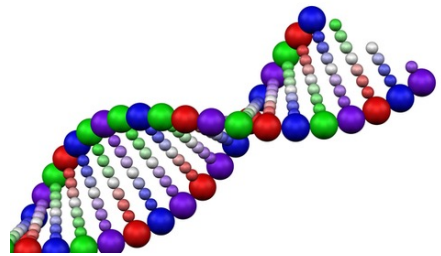
- Representation Learning: worth a conference name :)
  - International Conference on Learning Representations (ICLR)
- Represent raw data to solve complex problems
  - compression, abstraction, conceptualization
- Raw data in different forms
  - pixels, words, waves, states, molecules, DNA, …

Current approaches to object recognition make essential use of machine learning methods. To improve their performance, we can collect larger datasets, learn more powerful models, and use better techniques for preventing overfitting. Until recently, datasets of labeled images were relatively small — on the order of tens of thousands of images (e.g., NORB [16], Caltech-101/256 [8, 9], and CIFAR-10/100 [12]). Simple recognition tasks can be solved quite well with datasets of this size, especially if they are augmented with label-preserving transformations. For example, the current-best error rate on the MNIST digit-recognition task (<0.3%) approaches human performance [4]. But objects in realistic settings exhibit considerable variability, so to learn to recognize them it is necessary to use much larger training sets. And indeed, the shortcomings of small image datasets have been widely recognized (e.g., Pinto et al. [21]), but it has only recently become possible to collect labeled datasets with millions of images. The new larger datasets include LabelMe [23], which consists of hundreds of thousands of fully-segmented images, and ImageNet [6], which consists of over 15 million labeled high-resolution images in over 22,000 categories.

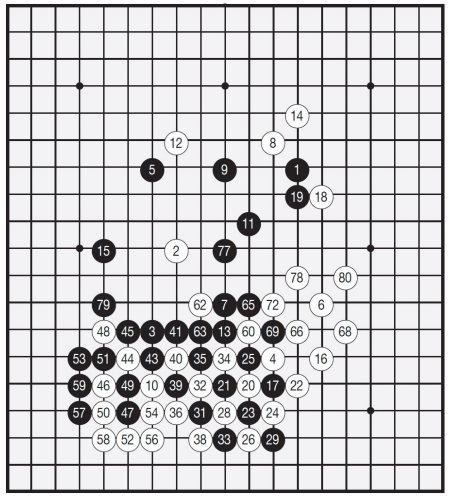
To learn about thousands of objects from millions of images, we need a model with a large learning capacity. However, the immense complexity of the object recognition task means that this problem cannot be specified even by a dataset as large as ImageNet, so our model should also have lots of prior knowledge to compensate for all the data we don't have. Convolutional neural networks (CNNs) constitute one such class of models [16, 11, 13, 18, 15, 22, 26]. Their capacity can be controlled by varying their depth and breadth, and they also make strong and mostly correct assumptions about the nature of images (namely, stationarity of statistics and locality of pixel dependencies). Thus, compared to standard feedforward neural networks with similarly-sized layers, CNNs have much fewer connections and parameters and so they are easier to train, while their theoretically-best performance is likely to be only slightly worse.







Deep Learning is Representation Learning

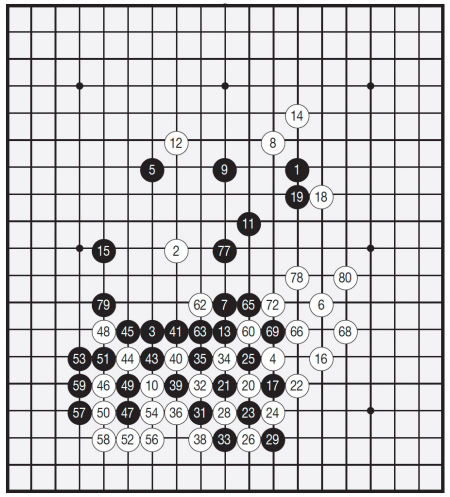


3^{361} states?

Game of Go

- an exponentially large number of states?
- infeasible to enumerate, memorize, or search

Deep Learning is Representation Learning



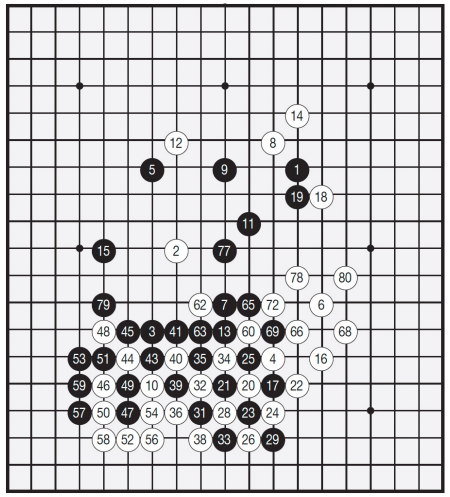
3^{361} states?

Bad representation

$256^{3x500x500}$?

- Image space has exponentially more states than Go.

Deep Learning is Representation Learning

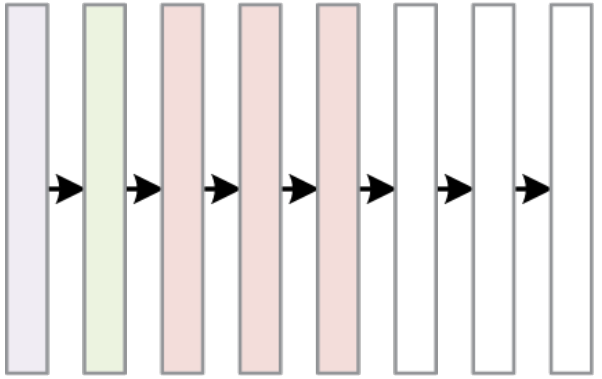


3^{361} states?

$256^{3x500x500}$?

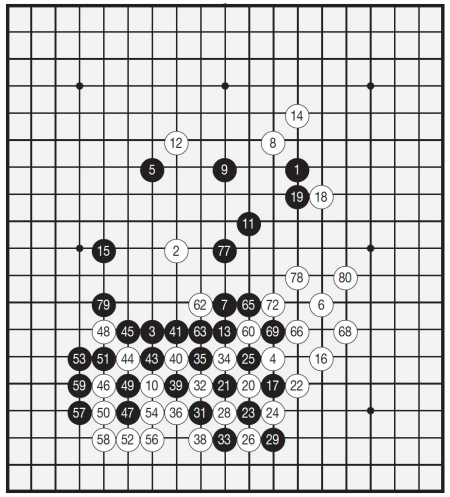
- Image recognition is solved in representation space.

Good representation

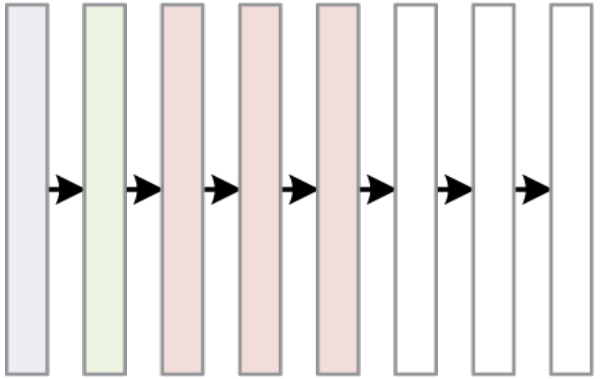


Deep Learning is Representation Learning

- Go playing can be solved in representation space.

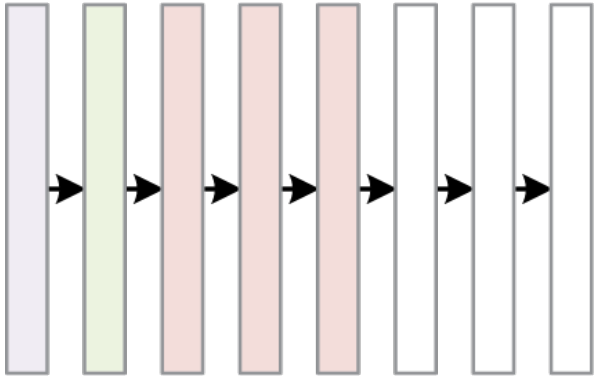


3^{361} states?



Good representation

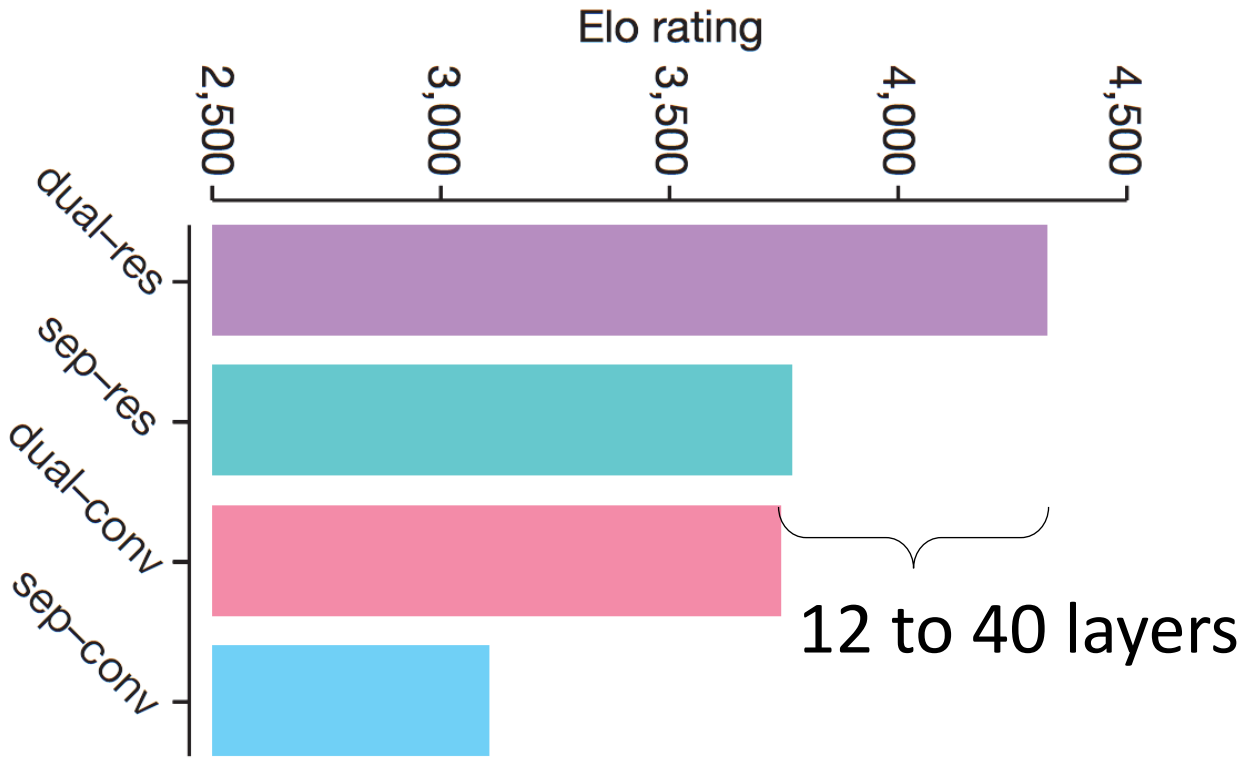
$256^{3x500x500}$?



Deep Learning is Representation Learning

AlphaGo: better representation leads to higher rating

- outperform best human players
- without human knowledge

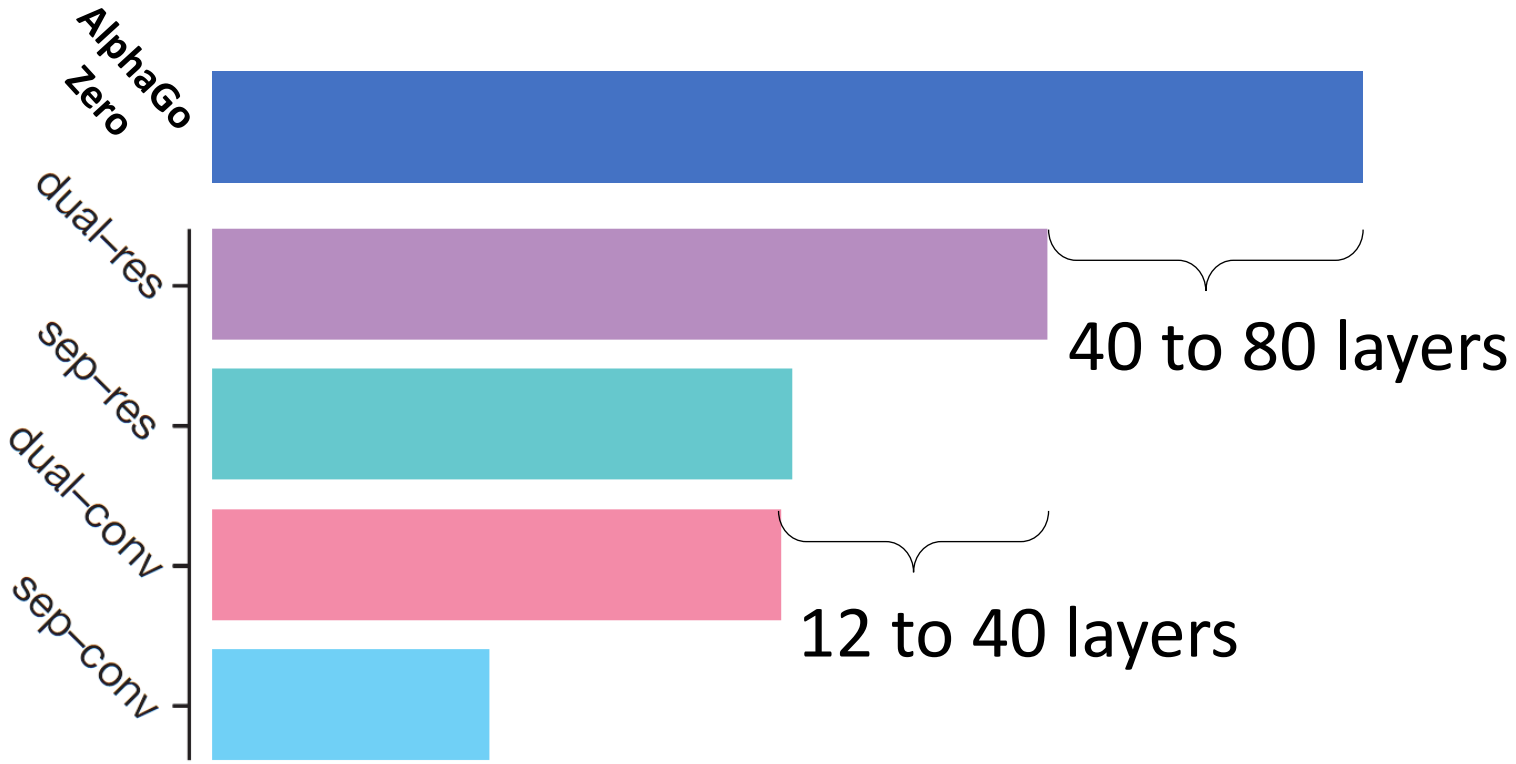


Silver, et al., Mastering the game of Go without human knowledge, Nature 2017

Deep Learning is Representation Learning

AlphaGo: better representation leads to higher rating

- outperform best human players
- without human knowledge



Silver, et al., Mastering the game of Go without human knowledge, Nature 2017

How to Represent Images?

➡ class

➡ edge ➡ class

➡ edge ➡ orientation ➡ class

➡ edge ➡ orientation ➡ histogram ➡ class

➡ edge ➡ orientation ➡ histogram ➡ clusters ➡ class

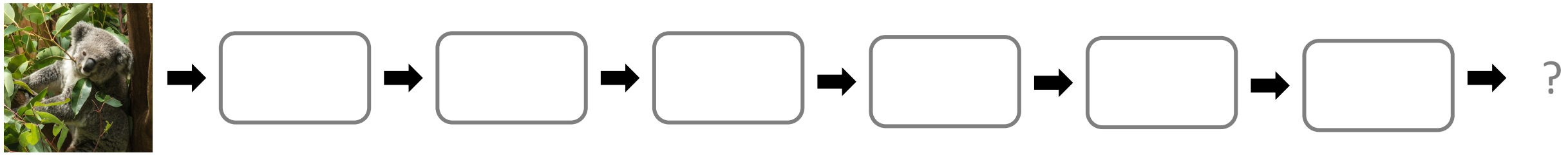
- Domain knowledge required
- OK-ish for low-level representations
- Extremely difficult to design high-level representations

deeper

Lowe 1999, Sivic & Zisserman 2003, Dalal & Triggs 2005, Grauman & Darrell 2005, Lazebnik et al 2006, Perronnin & Dance 2007, Yang et al 2009, Jégou et al 2010, …

How to Represent Images w/ Deep Learning?

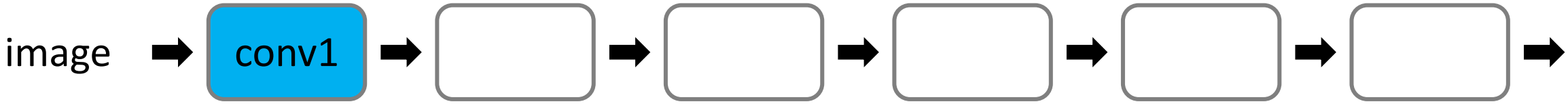
general modules (instead of specialized features)

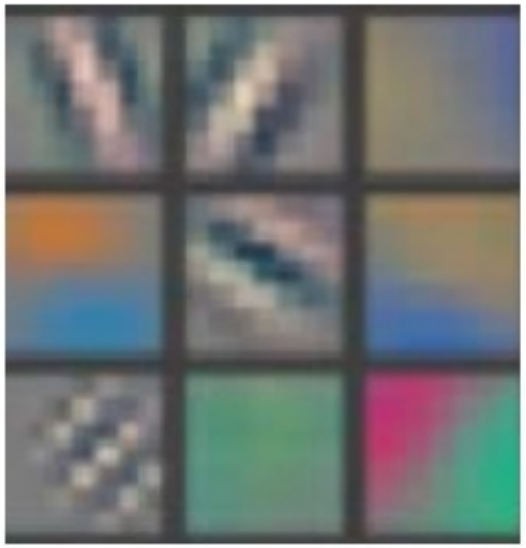


compose simple modules into complex functions

- build multiple levels of abstractions
- learn by back-prop
- learn from data
- reduce domain knowledge and feature engineering

Multiple Levels of Representations



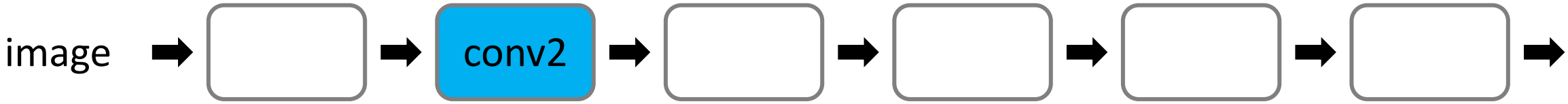


features

stimuli
(patches with the highest
1-hot activations)

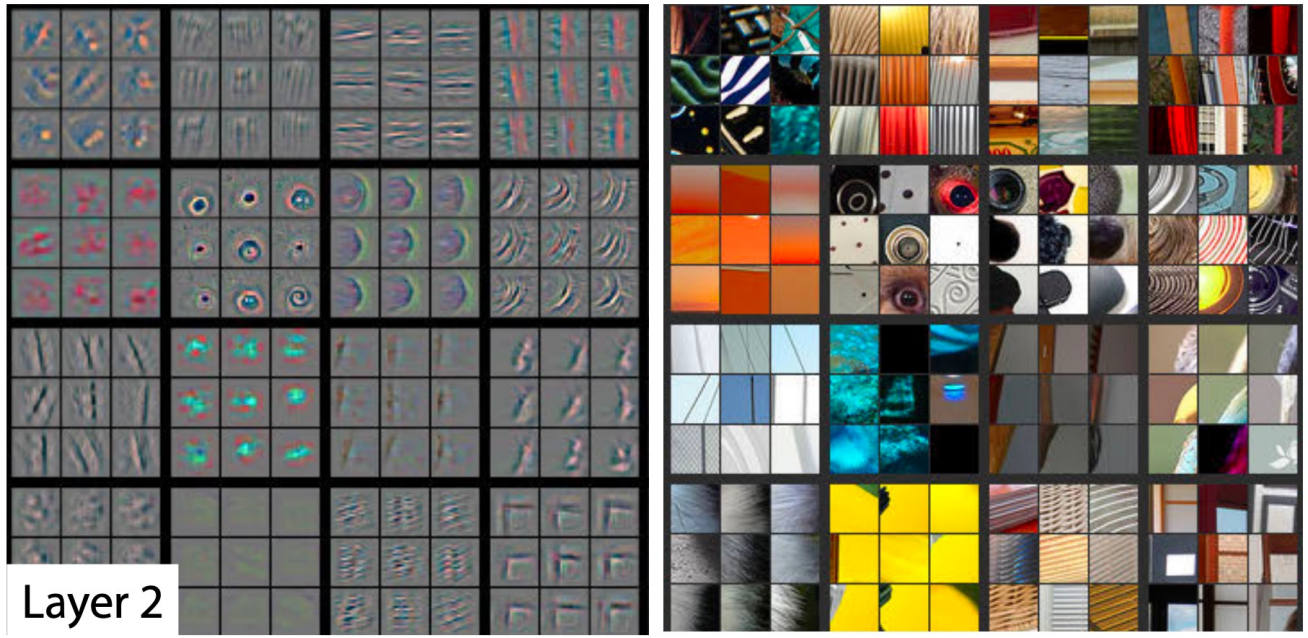
"Visualizing and Understanding Convolutional Networks", Zeiler & Fergus. ECCV 2014

Multiple Levels of Representations



features

stimuli

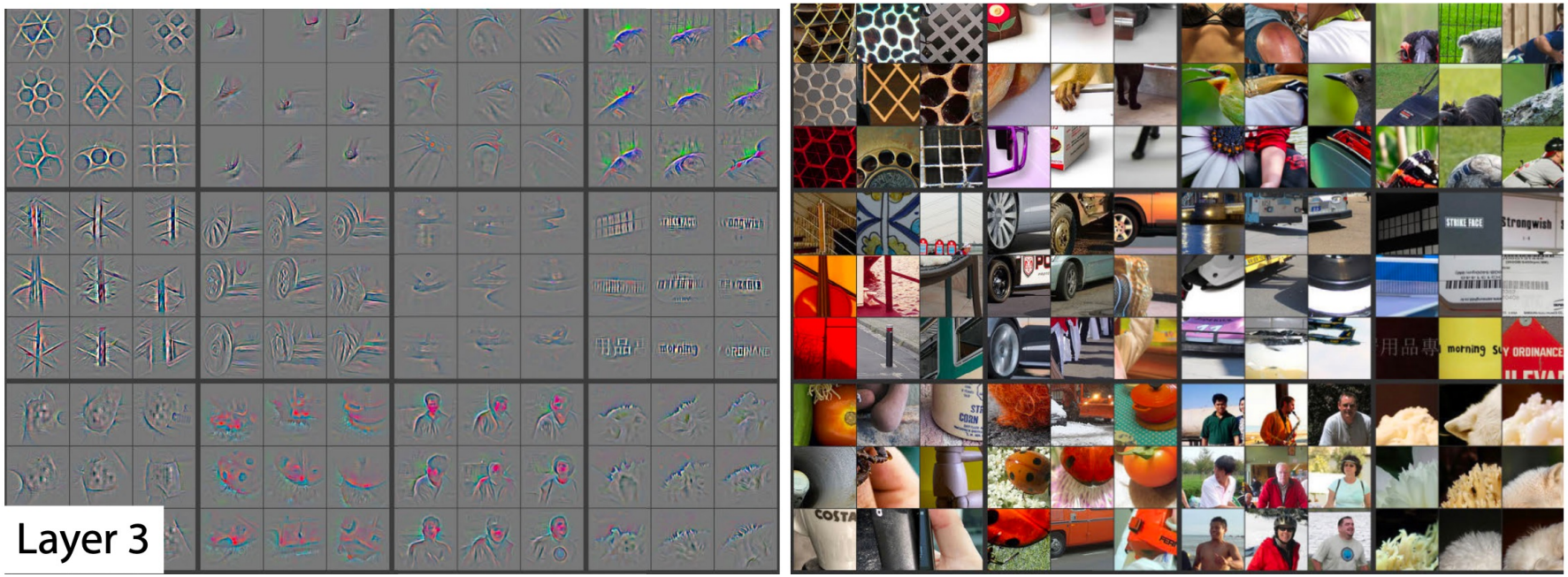


"Visualizing and Understanding Convolutional Networks", Zeiler & Fergus. ECCV 2014

Multiple Levels of Representations

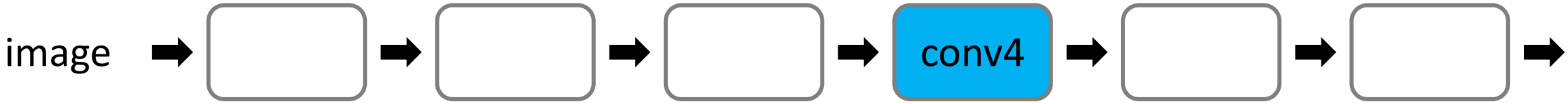
features

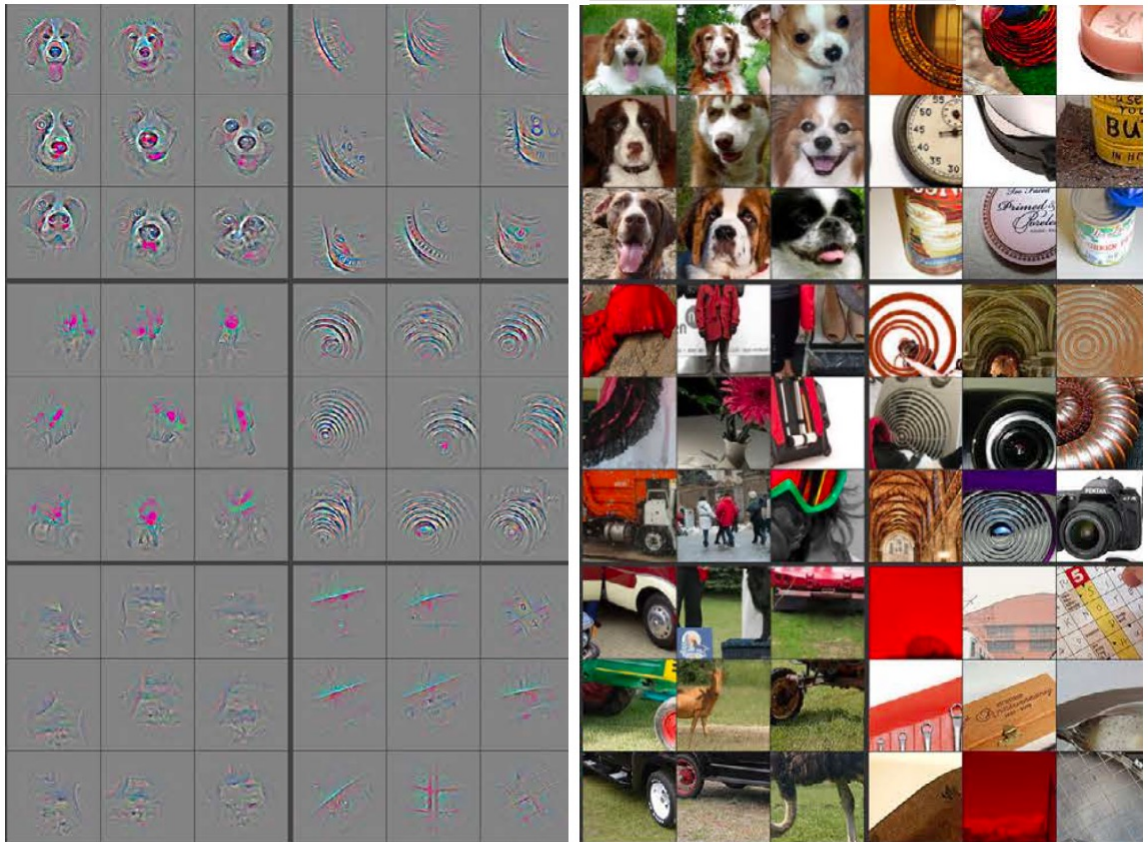
stimuli

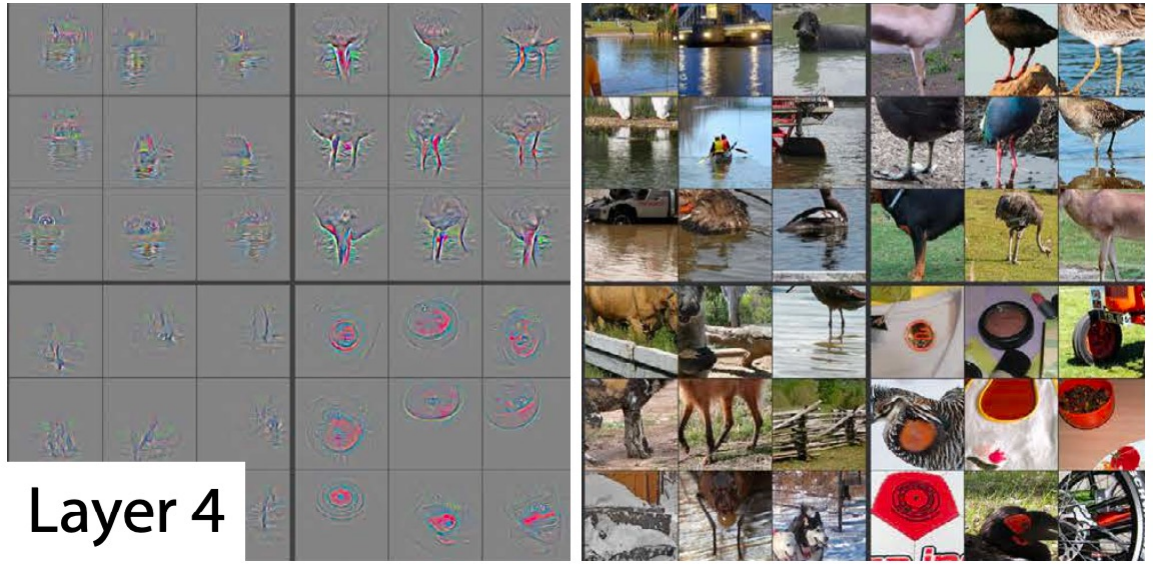


"Visualizing and Understanding Convolutional Networks", Zeiler & Fergus. ECCV 2014

Multiple Levels of Representations

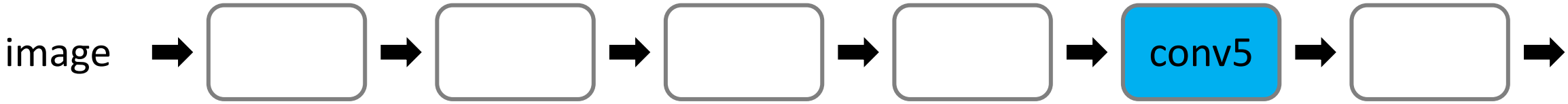


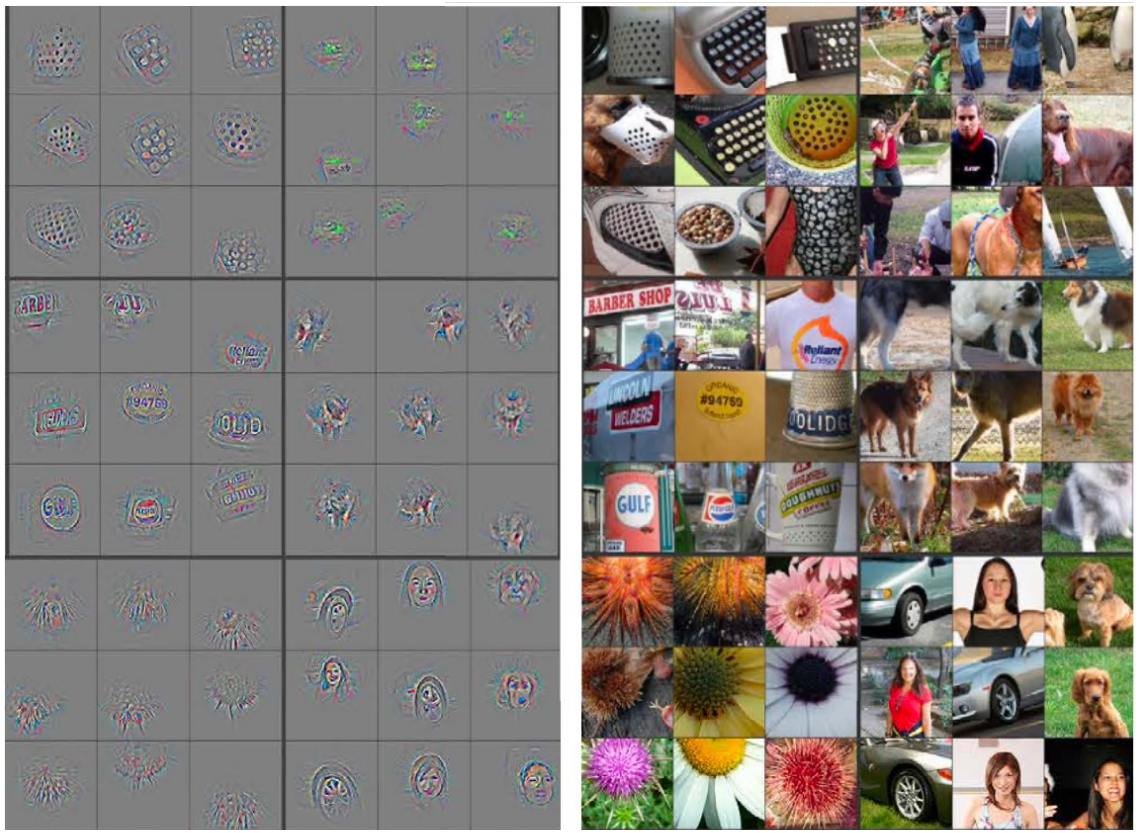


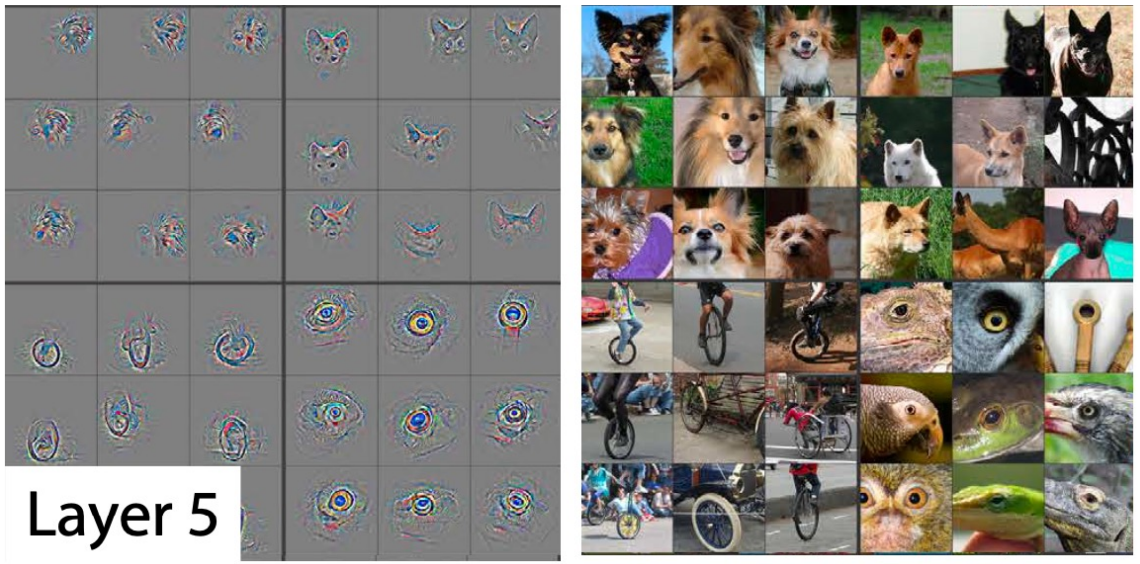


"Visualizing and Understanding Convolutional Networks", Zeiler & Fergus. ECCV 2014

Multiple Levels of Representations



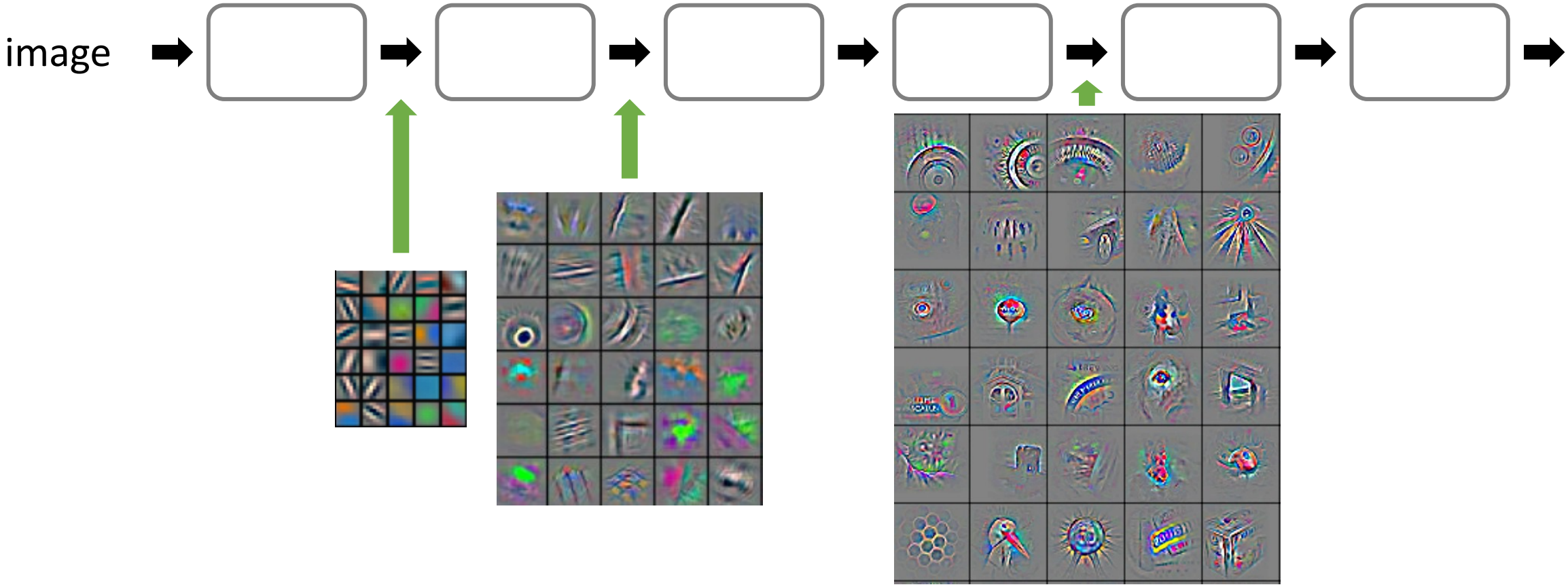




Deeper layers have “higher-level” features.

“Visualizing and Understanding Convolutional Networks”, Zeiler & Fergus. ECCV 2014

Multiple Levels of Representations



Deeper layers have “higher-level” features.

“Visualizing and Understanding Convolutional Networks”, Zeiler & Fergus. ECCV 2014

Deep Representations are Transferrable!

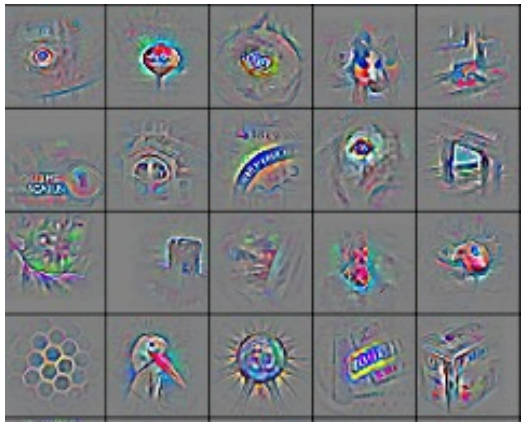
The single most important discovery in DL revolution

Transfer learning:

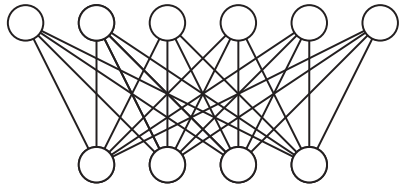
- pre-train on large-scale data
- fine-tune on small-scale data

- enable DL for small datasets
- revolutionize computer vision

- data: engine for general representation
- GPT: a similar principle



pre-train



fine-tune

"DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", Donahue et al. arXiv 2013
"Visualizing and Understanding Convolutional Networks", Zeiler & Fergus. arXiv 2013
"CNN Features off-the-shelf: an Astounding Baseline for Recognition", Razavian. arXiv 2014

Transfer learning

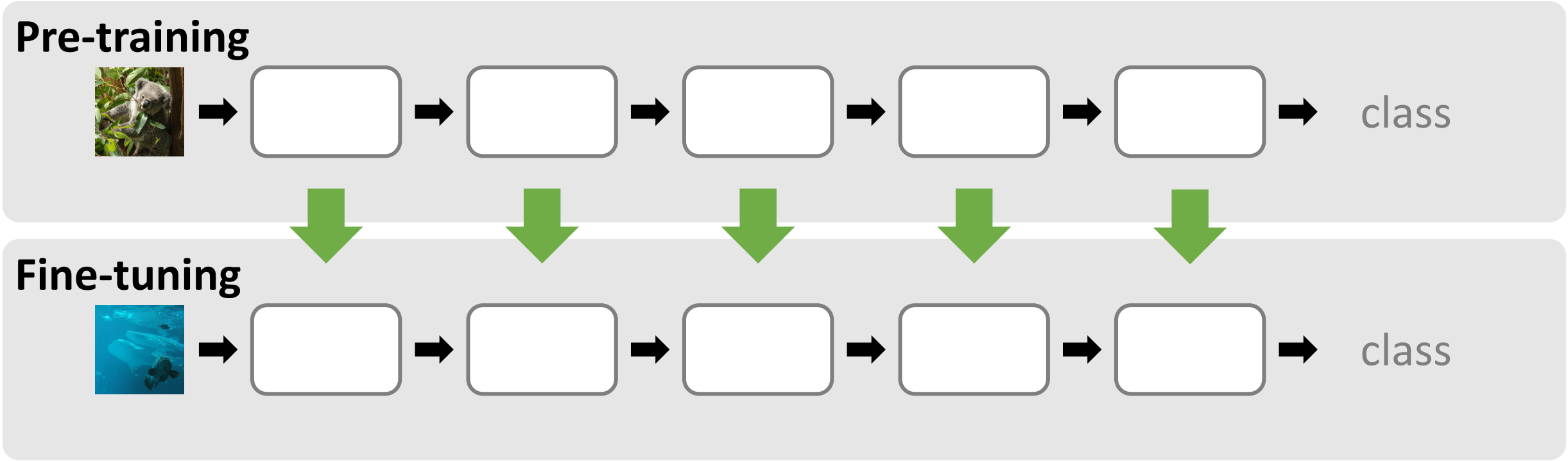
Transfer learning: Pre-training & Fine-tuning

Pre-training

Pre-training:

- to learn **general** representations
- on **large-scale** data
- train for a **long** time
- with **large** models

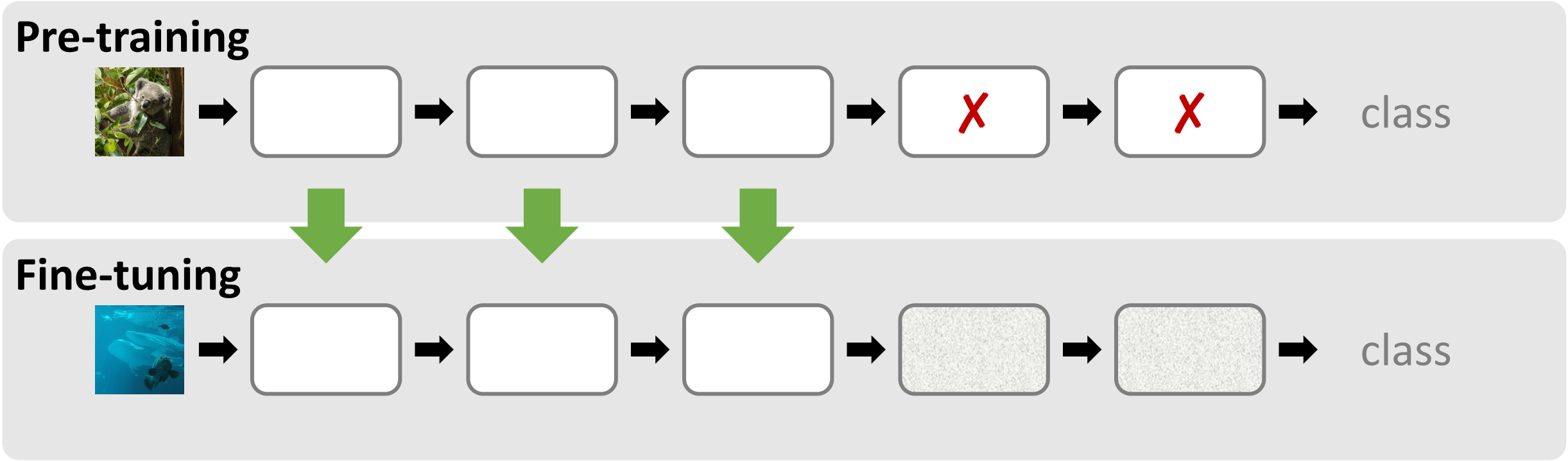
Transfer learning: Pre-training & Fine-tuning



Fine-tuning:

- transfer weights to **specific** tasks
- on **small-scale** data
- train for a **short** time, **lower** learning rate
- enable **large** models with lower risk of overfitting

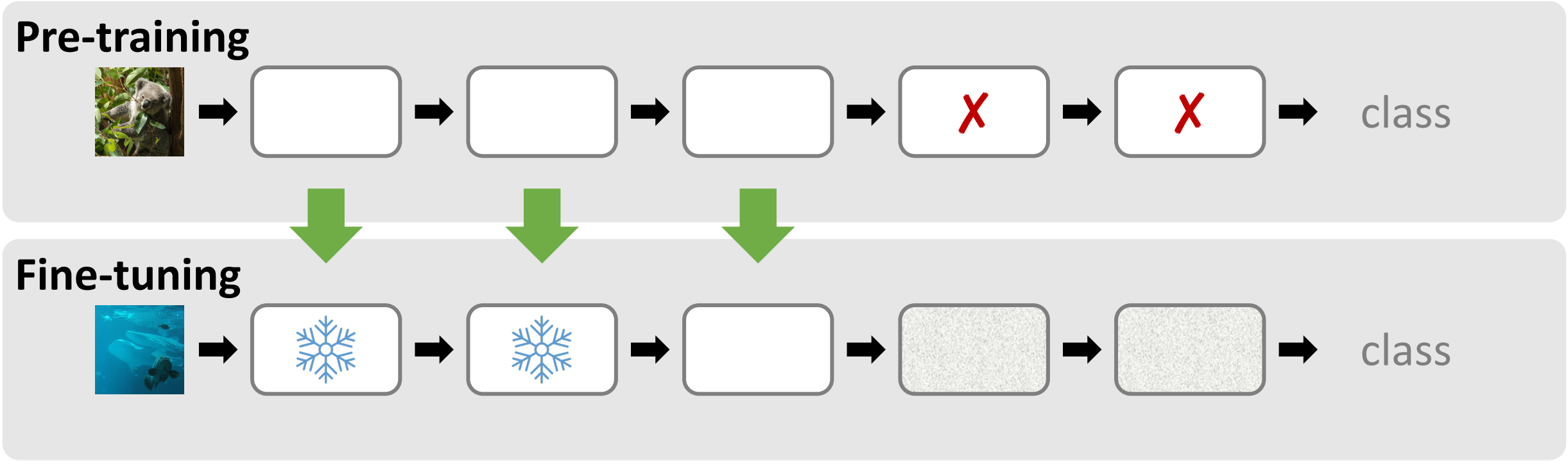
Transfer learning: Pre-training & Fine-tuning



Partial transfer

- pre-train and target domains may differ
- highest-level features are too adapted to pre-training
- randomly initialize new layers

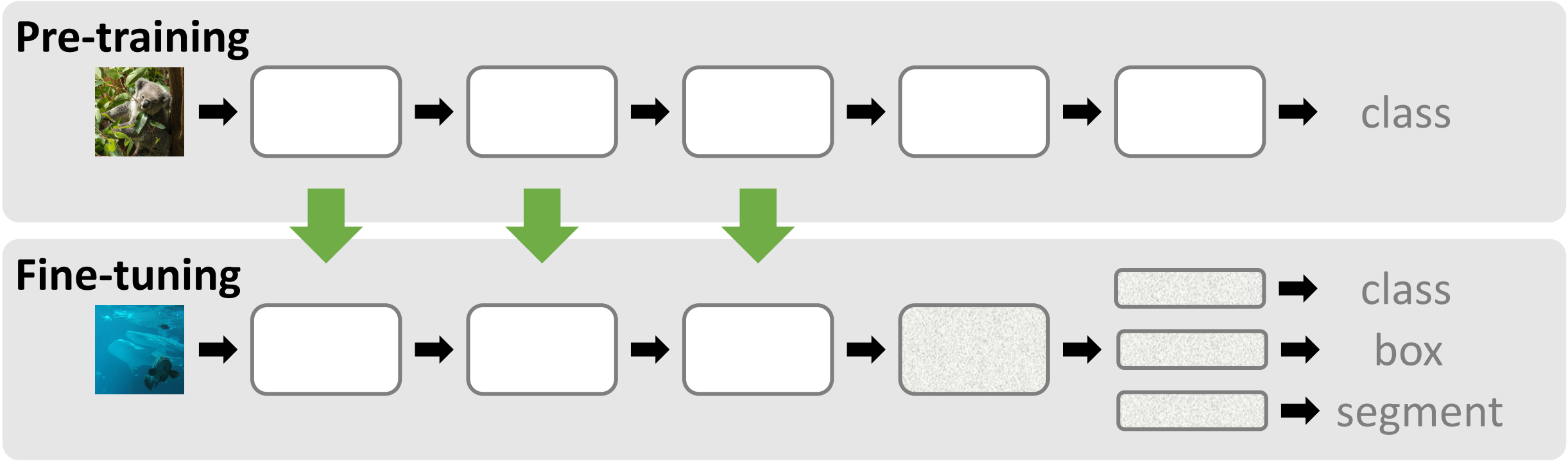
Transfer learning: Pre-training & Fine-tuning



Frozen weights

- freeze some/all pre-trained weights
- reduce overfitting if data is too little
- save memory, speed up training

Transfer learning: Pre-training & Fine-tuning



Network surgery

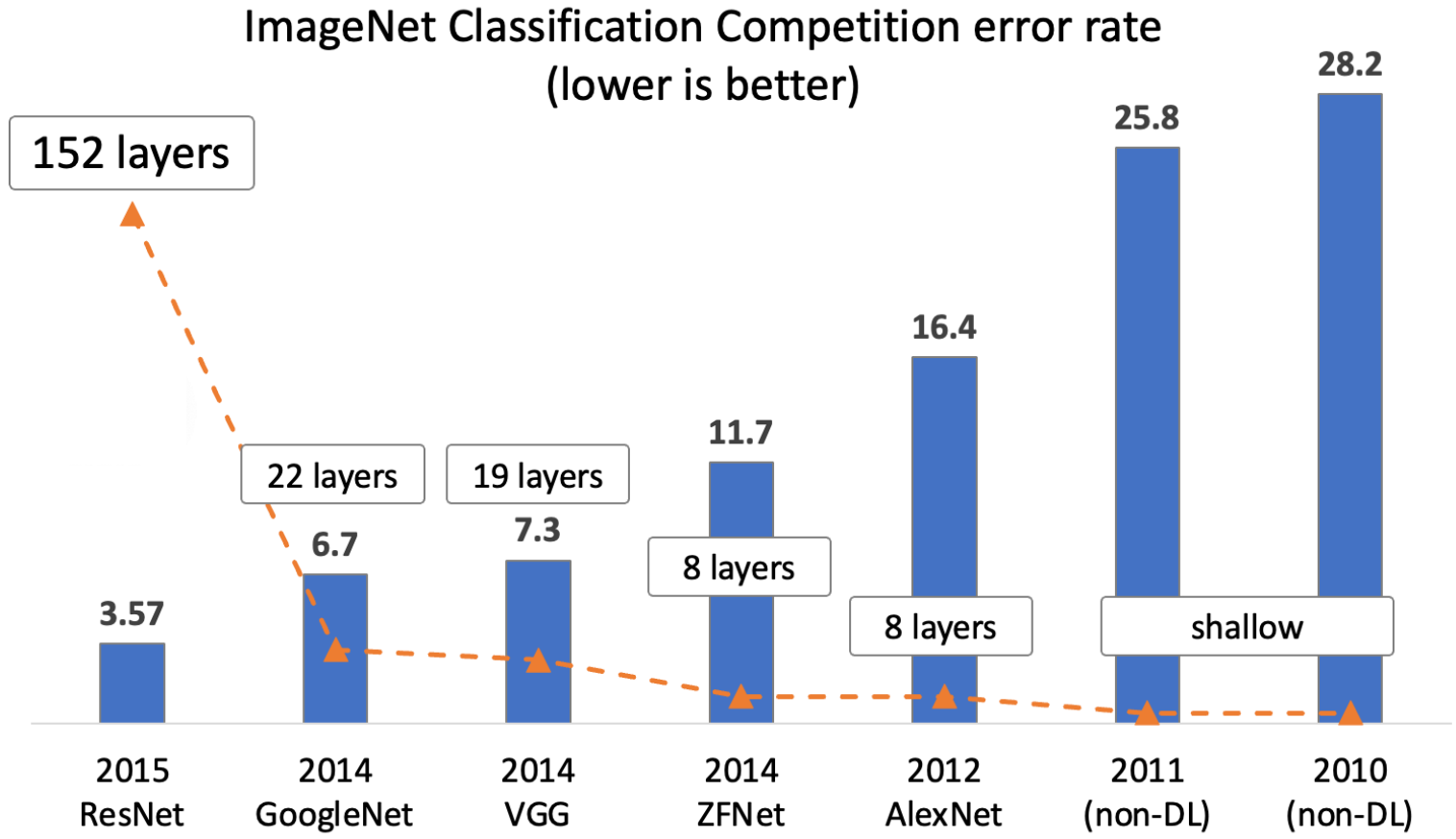
- re-purpose the model for other tasks (detect, segment)
- general features + task-specific predictions

How can we (pre-)train good representations?

- **Model** (network architecture)
  - scaling: deep, wide, large

How can we (pre-)train good representations?

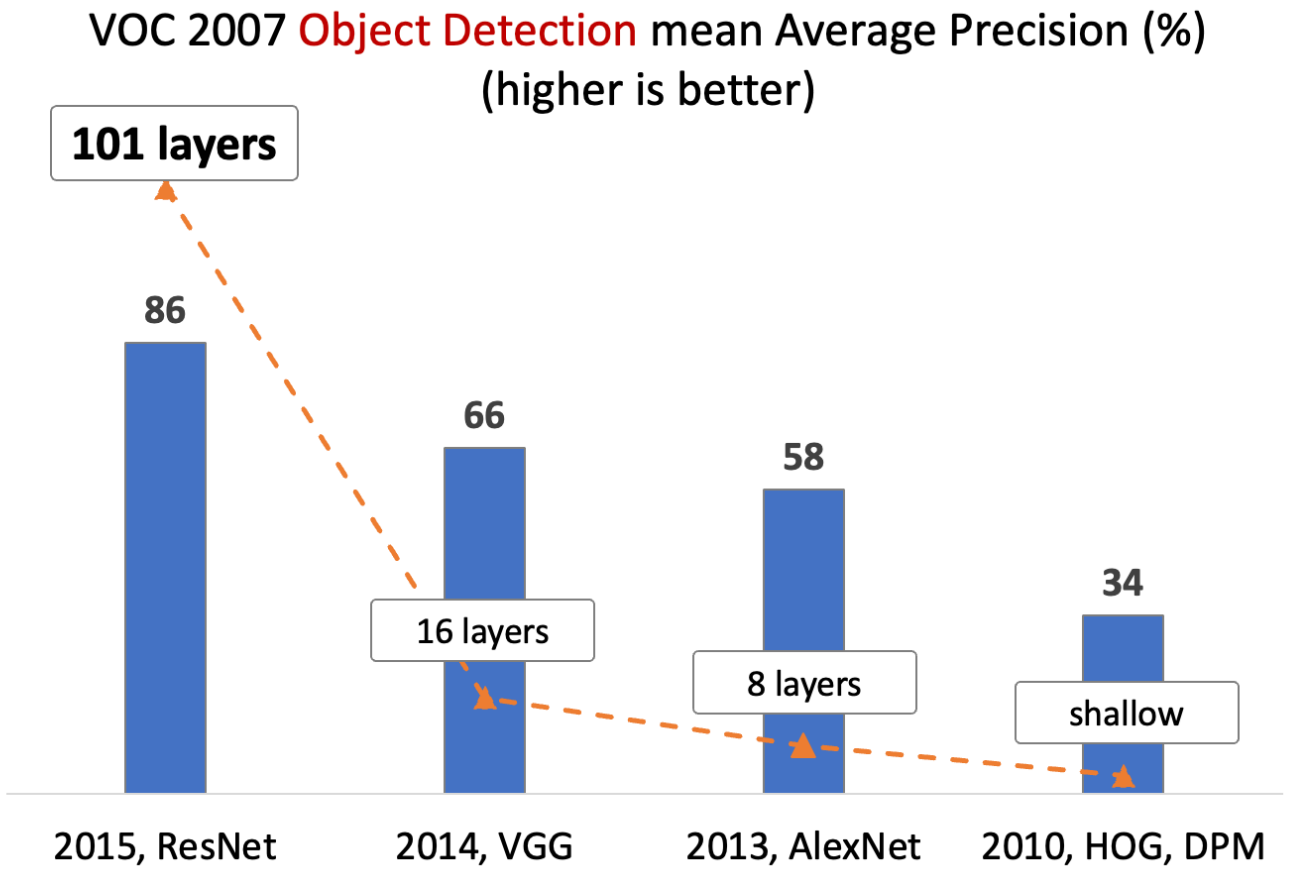
- **Model** (network architecture)
  - scaling: deep, wide, large



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

How can we (pre-)train good representations?

- **Model** (network architecture)
  - scaling: deep, wide, large



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

How can we (pre-)train good representations?

- **Model** (network architecture)
  - scaling: deep, wide, large
  - inductive bias: convolution, recurrency, attention

How can we (pre-)train good representations?

- **Model** (network architecture)
  - scaling: deep, wide, large
  - inductive bias: convolution, recurrency, attention
- **Data**
  - scaling
  - curating, cleaning, filtering, ...

How can we (pre-)train good representations?

- **Model** (network architecture)
  - scaling: deep, wide, large
  - inductive bias: convolution, recurrency, attention
- **Data**
  - scaling
  - curating, cleaning, filtering, ...
- **Learning objective**
  - supervised
  - unsupervised
  - self-supervised

Pre-training Objective

- **Supervised Learning**
  - w/ human annotated labels (e.g., classes)

- **Unsupervised Learning**
  - w/o human annotated labels

- **Self-supervised Learning**
  - labels induced by data “itself”
  - modern “unsupervised learning” in camouflage

} remaining of this lecture

Unsupervised Learning

Unsupervised Learning

- Conceptually, "**unsupervised learning**" involves any learning method **without** using human labels.

- In the modern literature, "**unsupervised learning**" often refers to a family of classical methods for:
  - **clustering**
  - **dimension reduction**

- As we will show, even classical "**unsupervised learning**" methods are kind of "**self-supervised**".

Unsupervised Learning

Classical (but very useful) unsupervised learning methods:

- K-means clustering
- Principal Component Analysis (PCA)
- Autoencoder

Unsupervised Learning

Classical (but very useful) unsupervised learning methods:

- **K-means clustering**
- Principal Component Analysis (PCA)
- Autoencoder

K-means clustering

Represent data by k centers

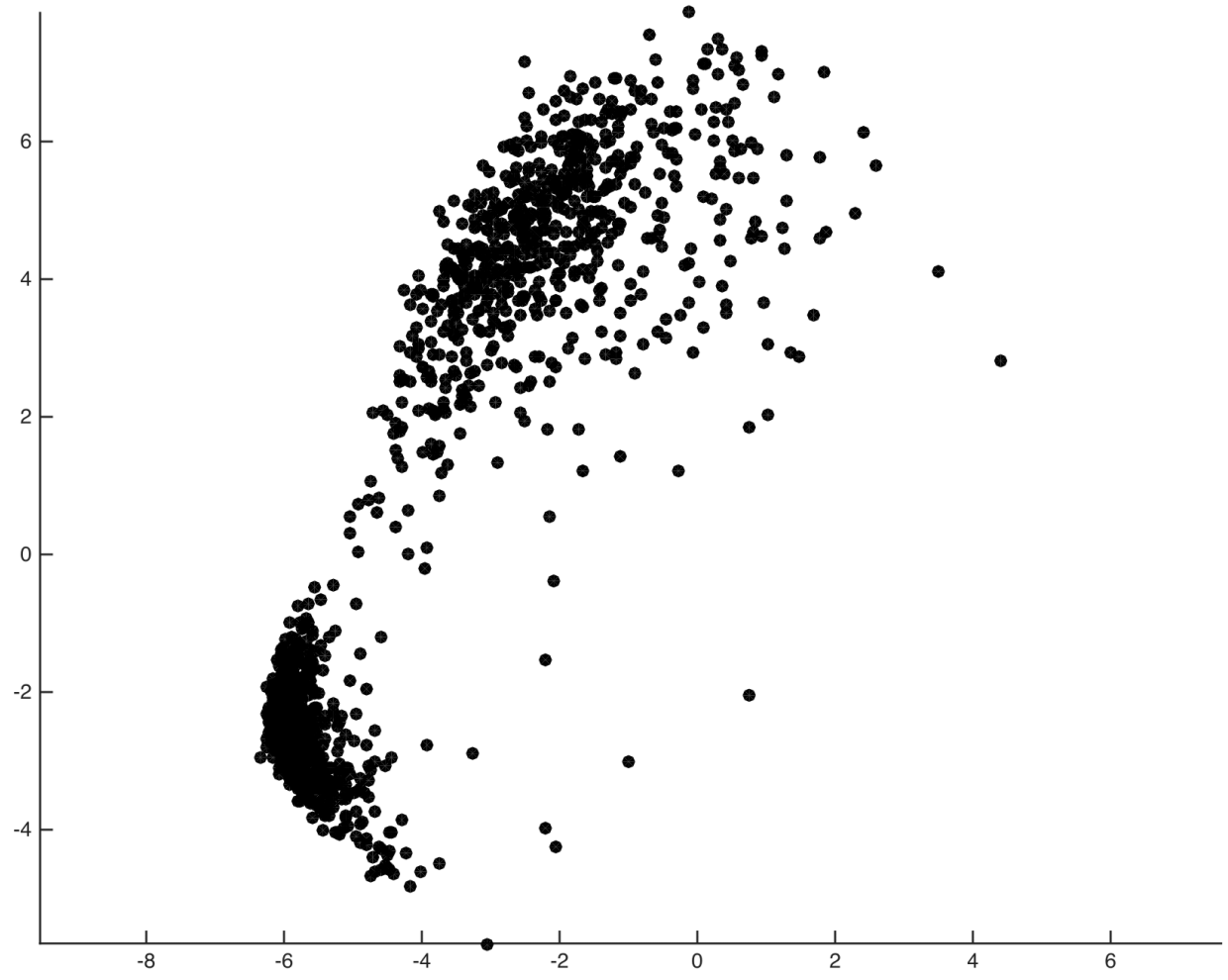


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

K-means clustering

Represent data by k centers

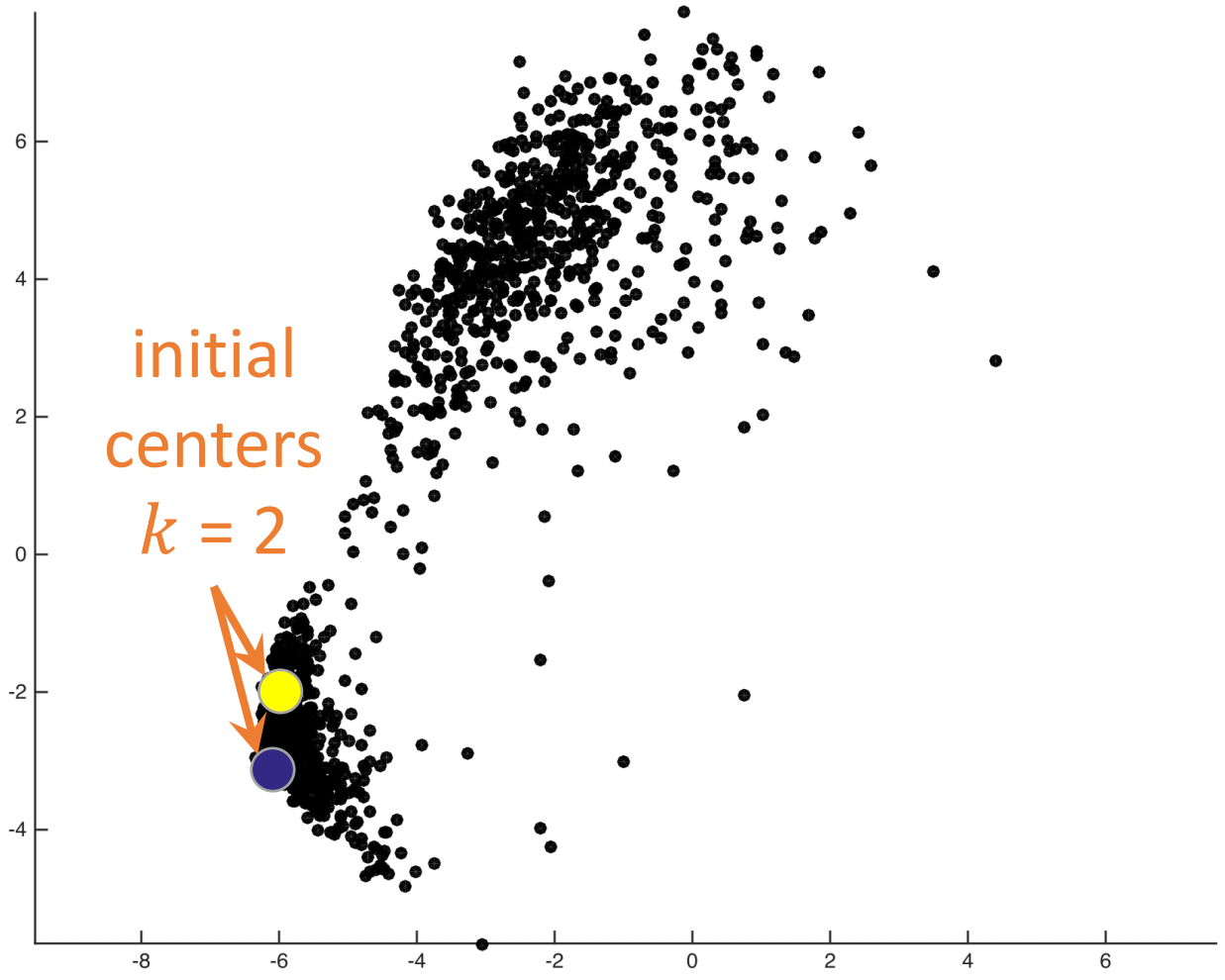


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

K-means clustering

Represent data by k centers

- **Assignment**: each sample is assigned to the nearest center

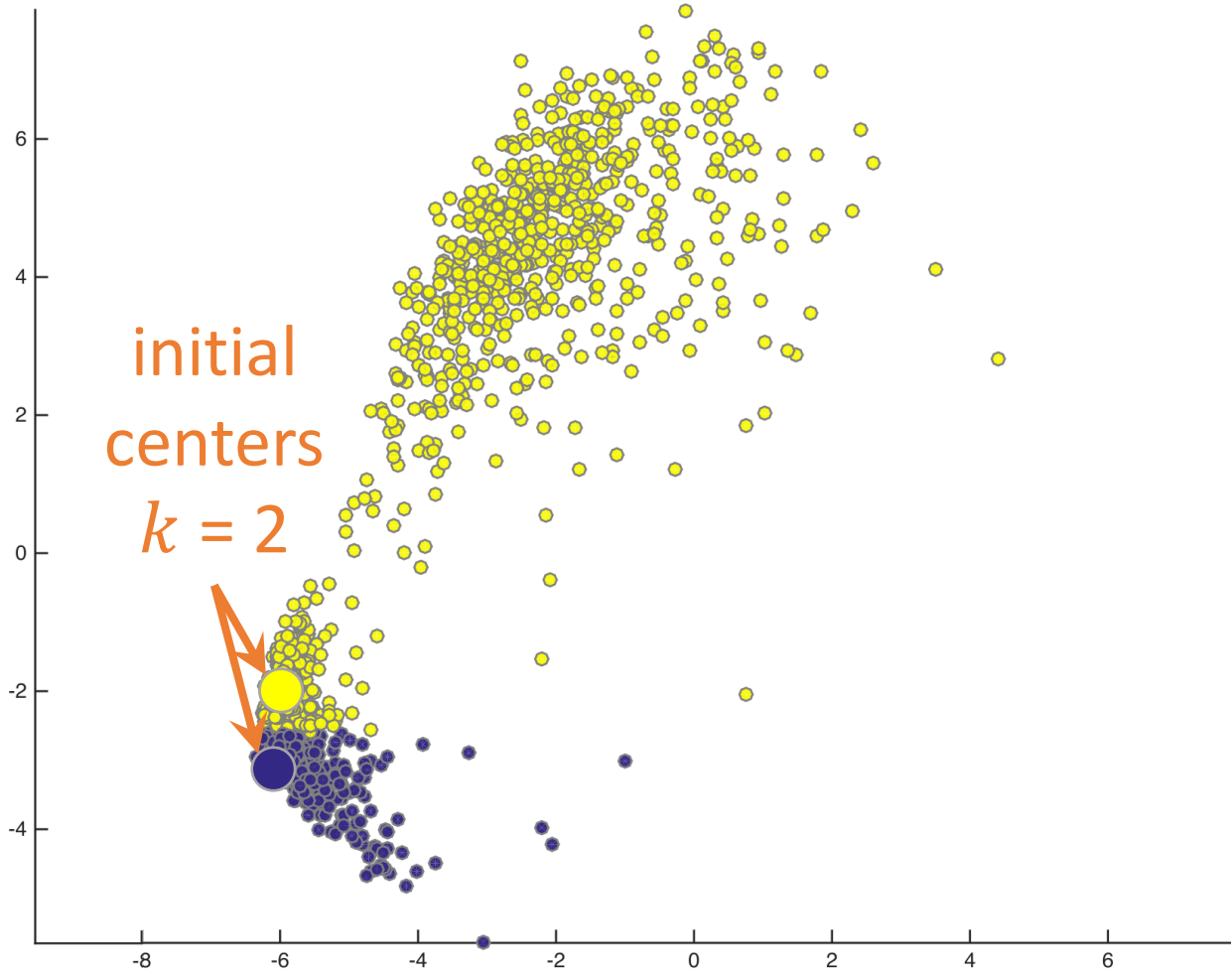


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

K-means clustering

Represent data by k centers

- **Assignment**: each sample is assigned to the nearest center
- **Update**: each center is updated by the mean of samples assigned to it

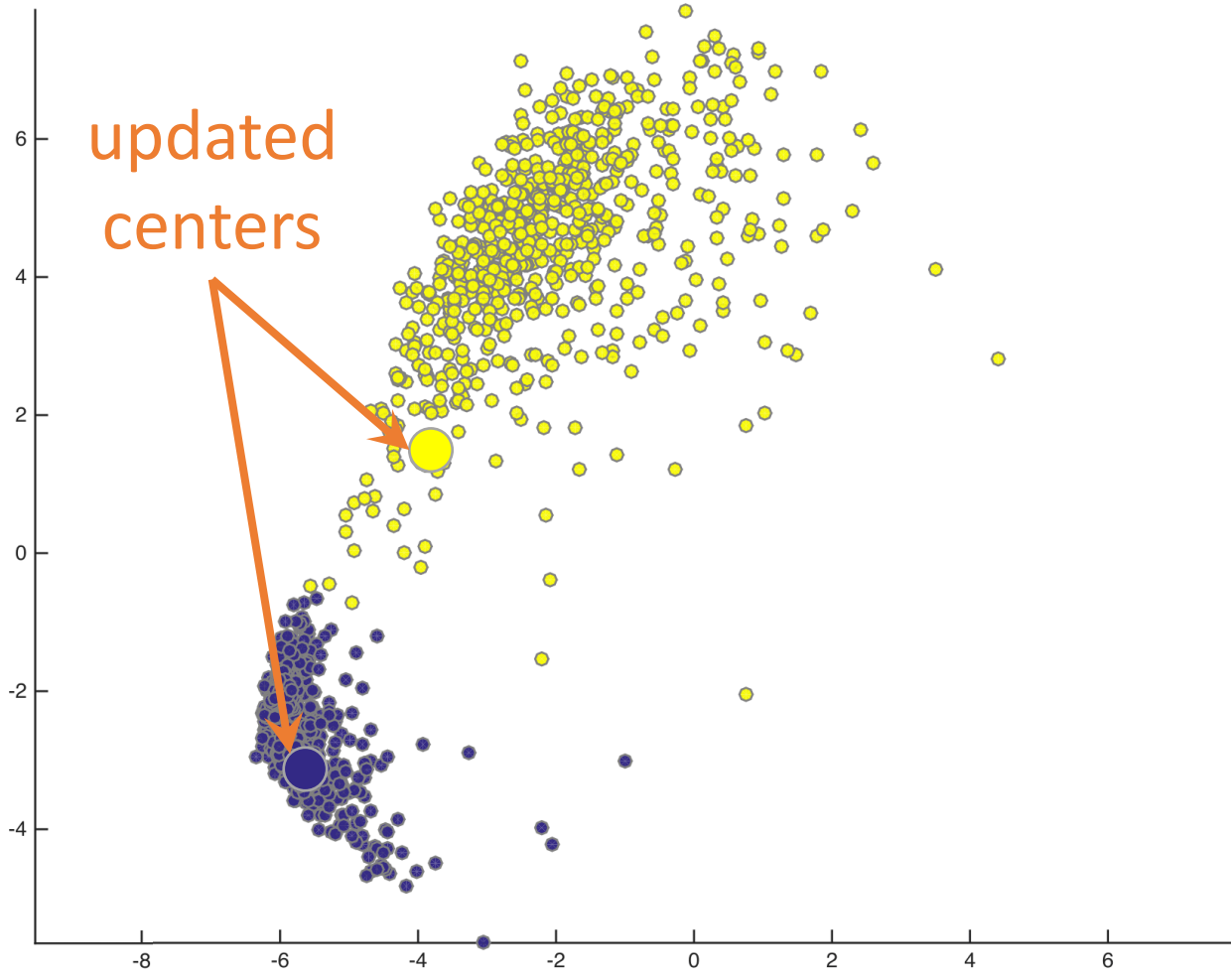


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

K-means clustering

Represent data by k centers

- **Assignment**: each sample is assigned to the nearest center
- **Update**: each center is updated by the mean of samples assigned to it
- Repeat until converge

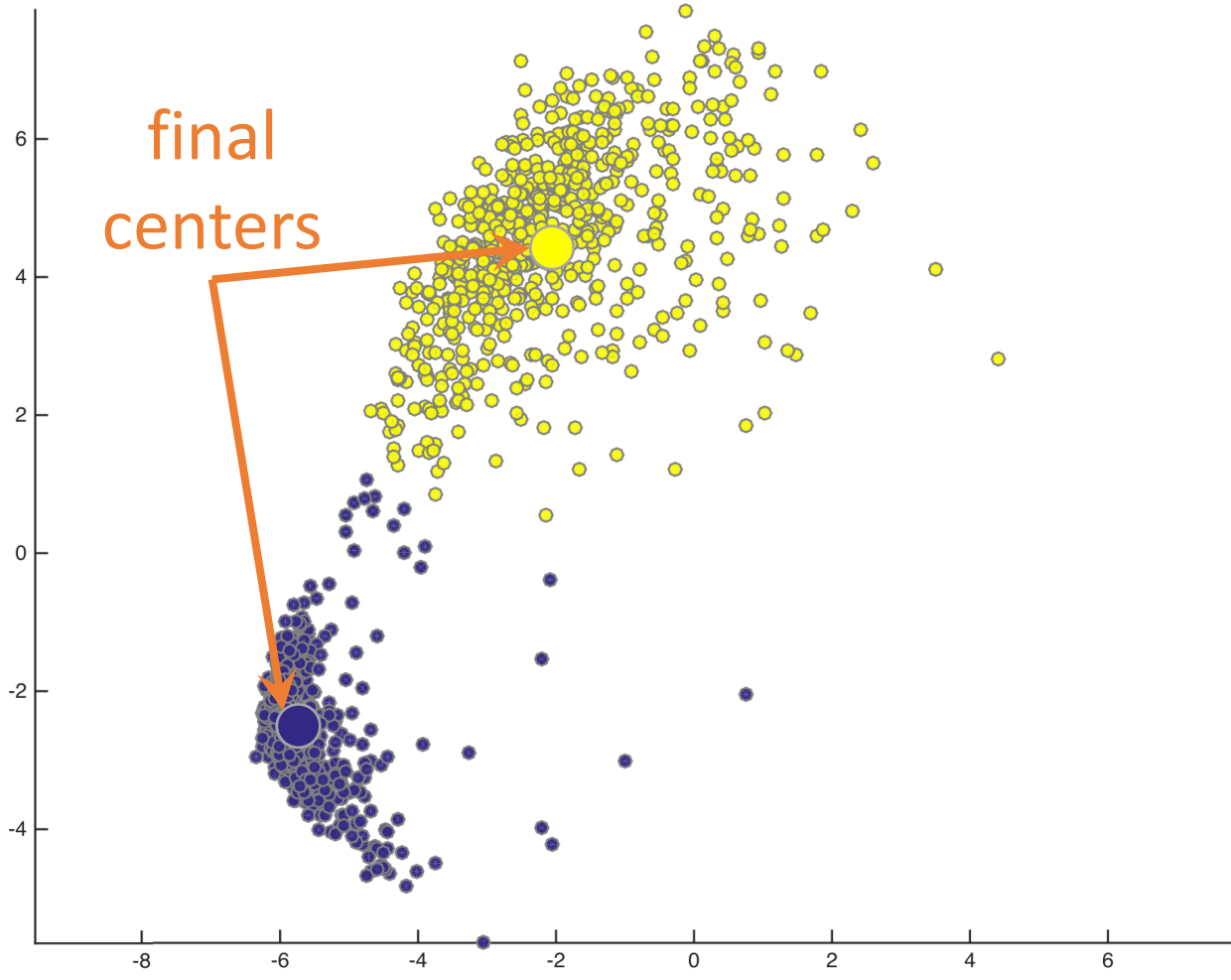


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

K-means clustering

Loss function:

$i(x)$: the cluster index of sample x
$c(i)$: the i-th center

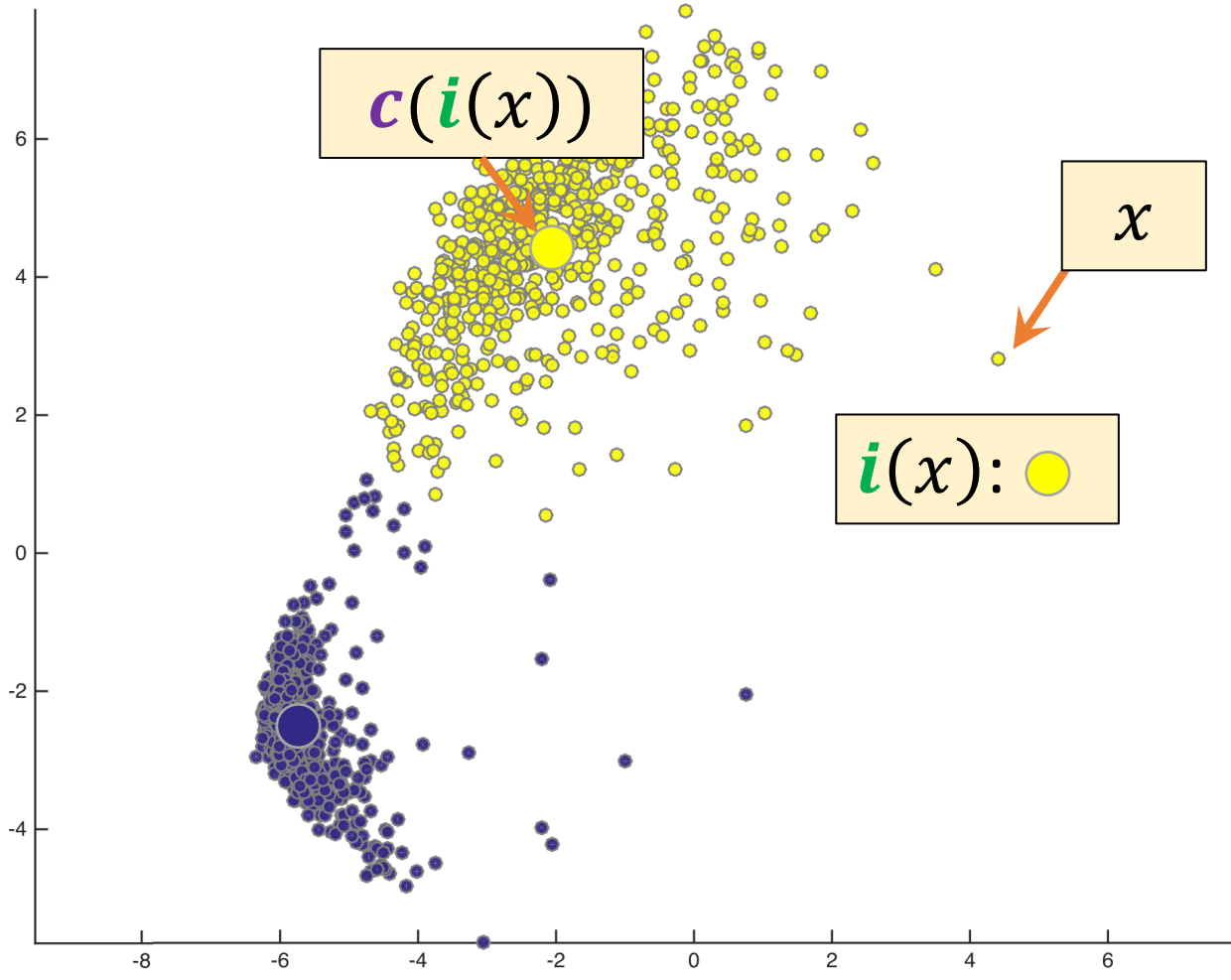


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

K-means clustering

$\boldsymbol{i}(x)$: the cluster index of sample x
$\boldsymbol{c}(i)$: the i-th center

Loss function:

$$\min_{\boldsymbol{c},\boldsymbol{i}} \sum_x \|x - \boldsymbol{c}(\boldsymbol{i}(x))\|^2$$

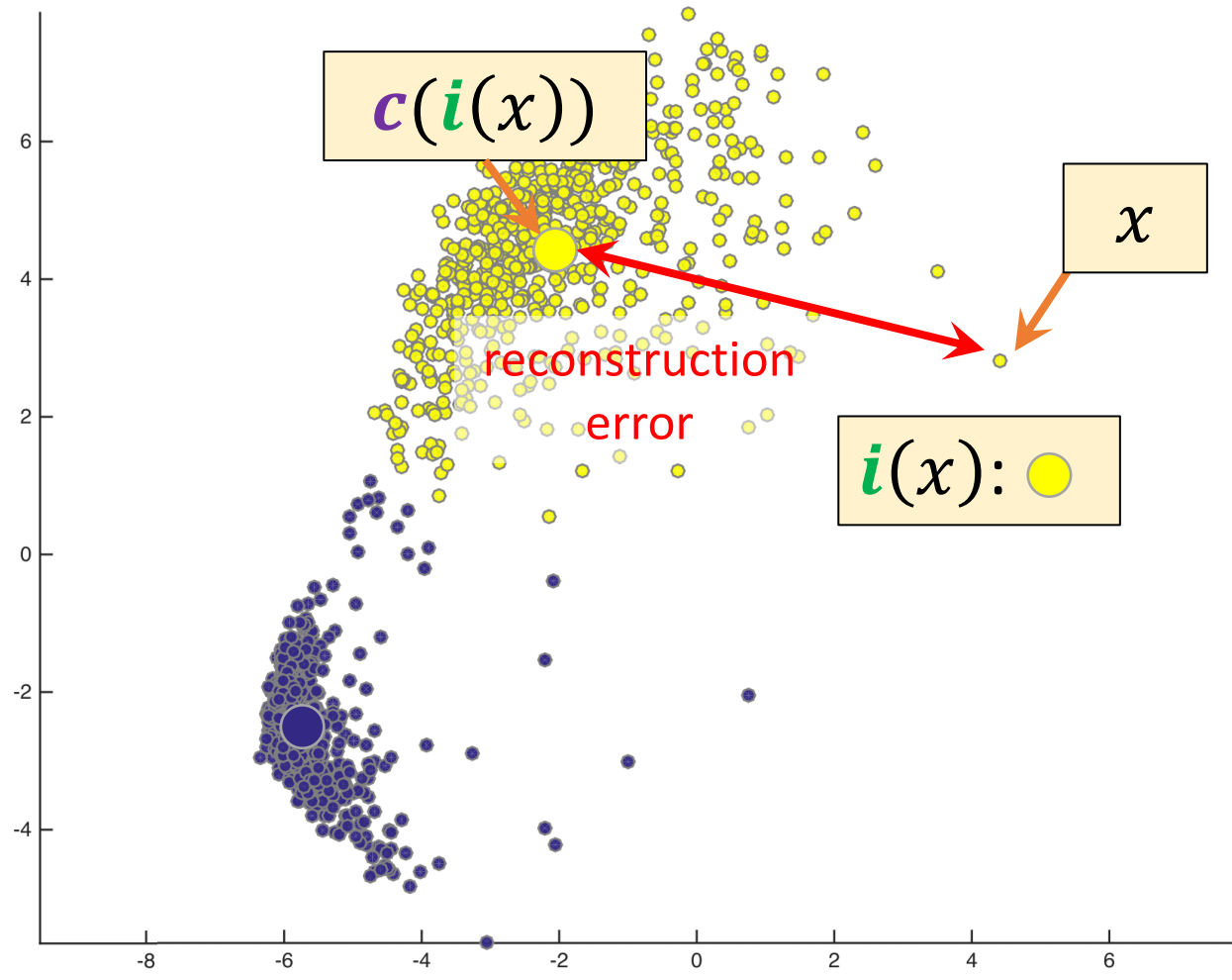


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

K-means clustering

$\boldsymbol{i}(x)$: the cluster index of sample x
$\boldsymbol{c}(i)$: the i-th center

Loss function:

$$\min_{\boldsymbol{c},\boldsymbol{i}} \sum_x \|x - \boldsymbol{c}(\boldsymbol{i}(x))\|^2$$

Assignment: fixed $\boldsymbol{c}$, optimize $\boldsymbol{i}$

$$\boldsymbol{i}(x) = \underset{\boldsymbol{i}}{\operatorname{argmin}} \|x - \boldsymbol{c}(\boldsymbol{i}(x))\|^2$$

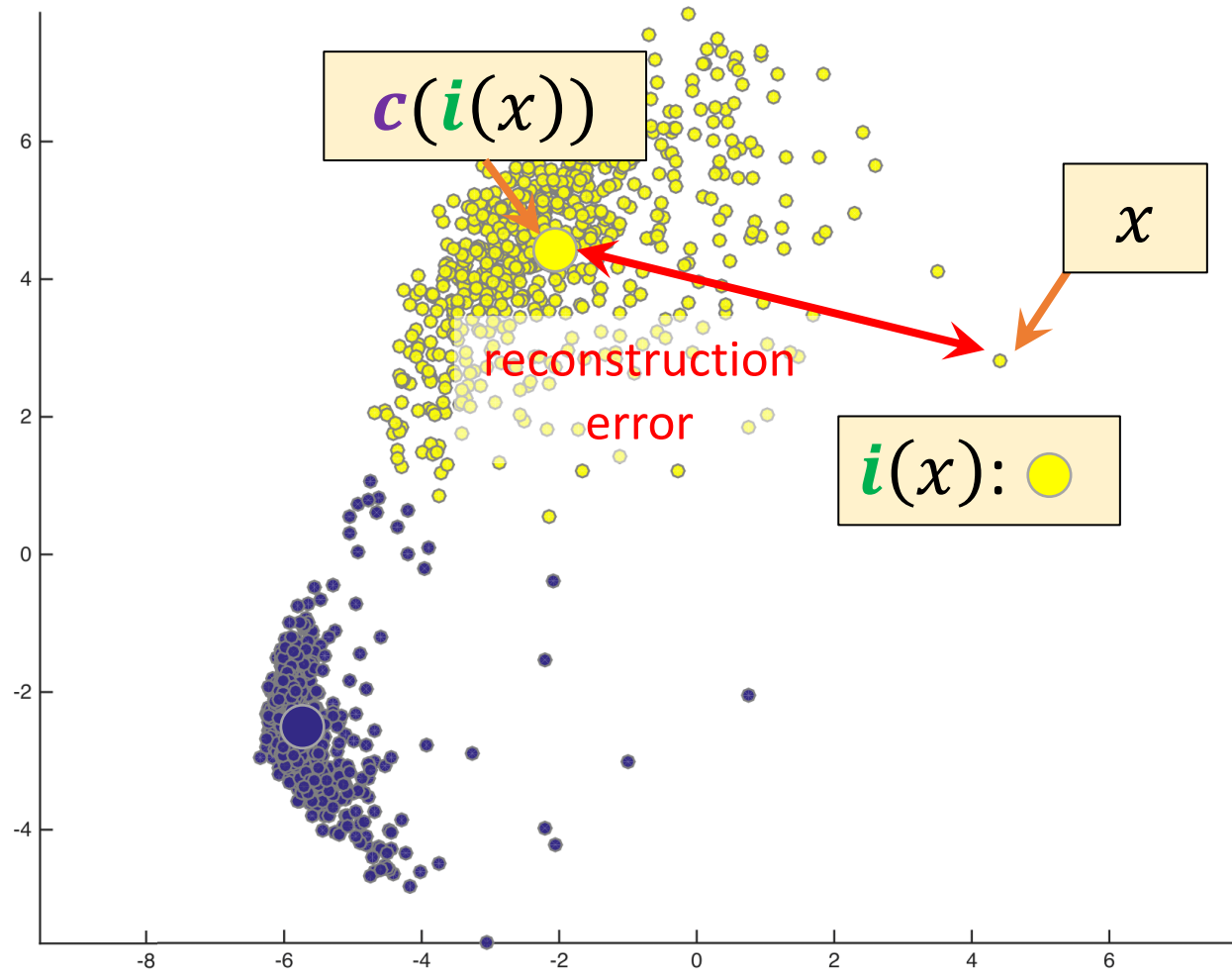


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

K-means clustering

$\boldsymbol{i}(x)$: the cluster index of sample x
$\boldsymbol{c}(i)$: the i-th center

Loss function:

$$\min_{\boldsymbol{c},\boldsymbol{i}} \sum_x \|x - \boldsymbol{c}(\boldsymbol{i}(x))\|^2$$

Assignment: fixed $\boldsymbol{c}$, optimize $\boldsymbol{i}$

Update: fixed $\boldsymbol{i}$, optimize $\boldsymbol{c}$

$$\boldsymbol{c}(j) = \operatorname*{argmin}_{\boldsymbol{c}(j)} \sum_{x:\, \boldsymbol{i}(x)=j} \|x - \boldsymbol{c}(j)\|^2$$

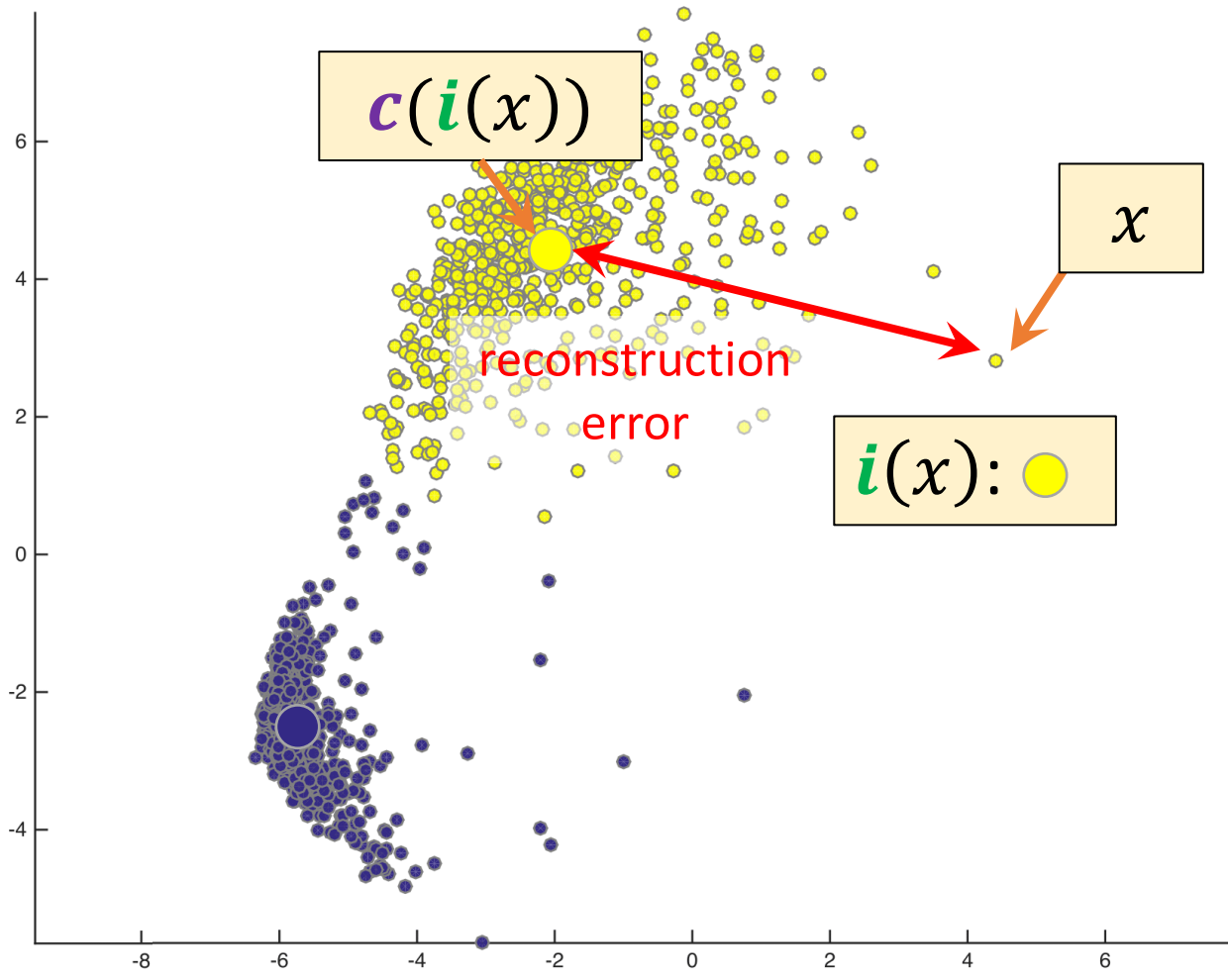


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

K-means clustering

$\boldsymbol{i}(x)$: the cluster index of sample x
$\boldsymbol{c}(i)$: the i-th center

Loss function:

$$\min_{\boldsymbol{c},\boldsymbol{i}} \sum_x \|x - \boldsymbol{c}(\boldsymbol{i}(x))\|^2$$

Assignment: fixed $\boldsymbol{c}$, optimize $\boldsymbol{i}$

Update: fixed $\boldsymbol{i}$, optimize $\boldsymbol{c}$

Repeat until converge

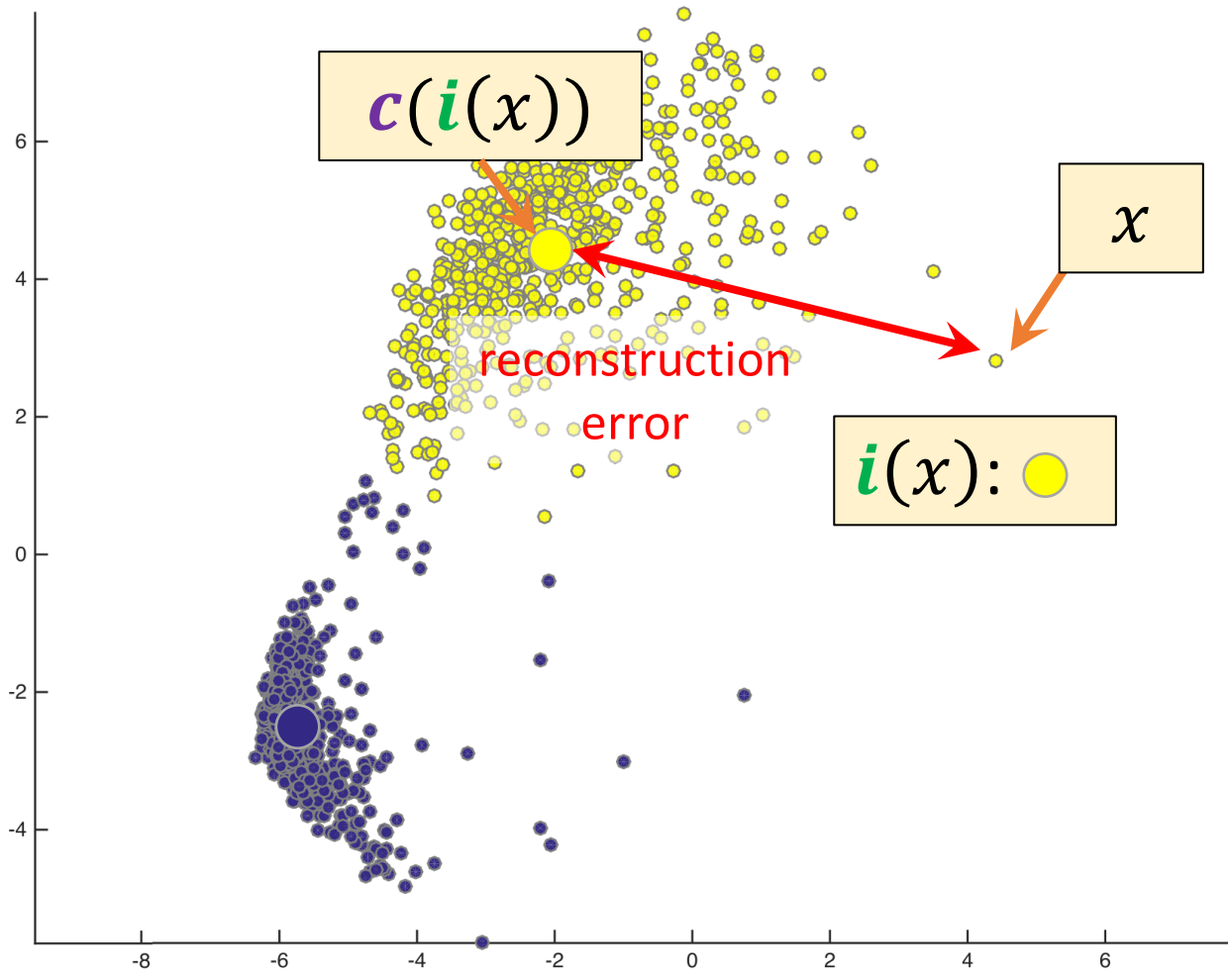


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

K-means clustering

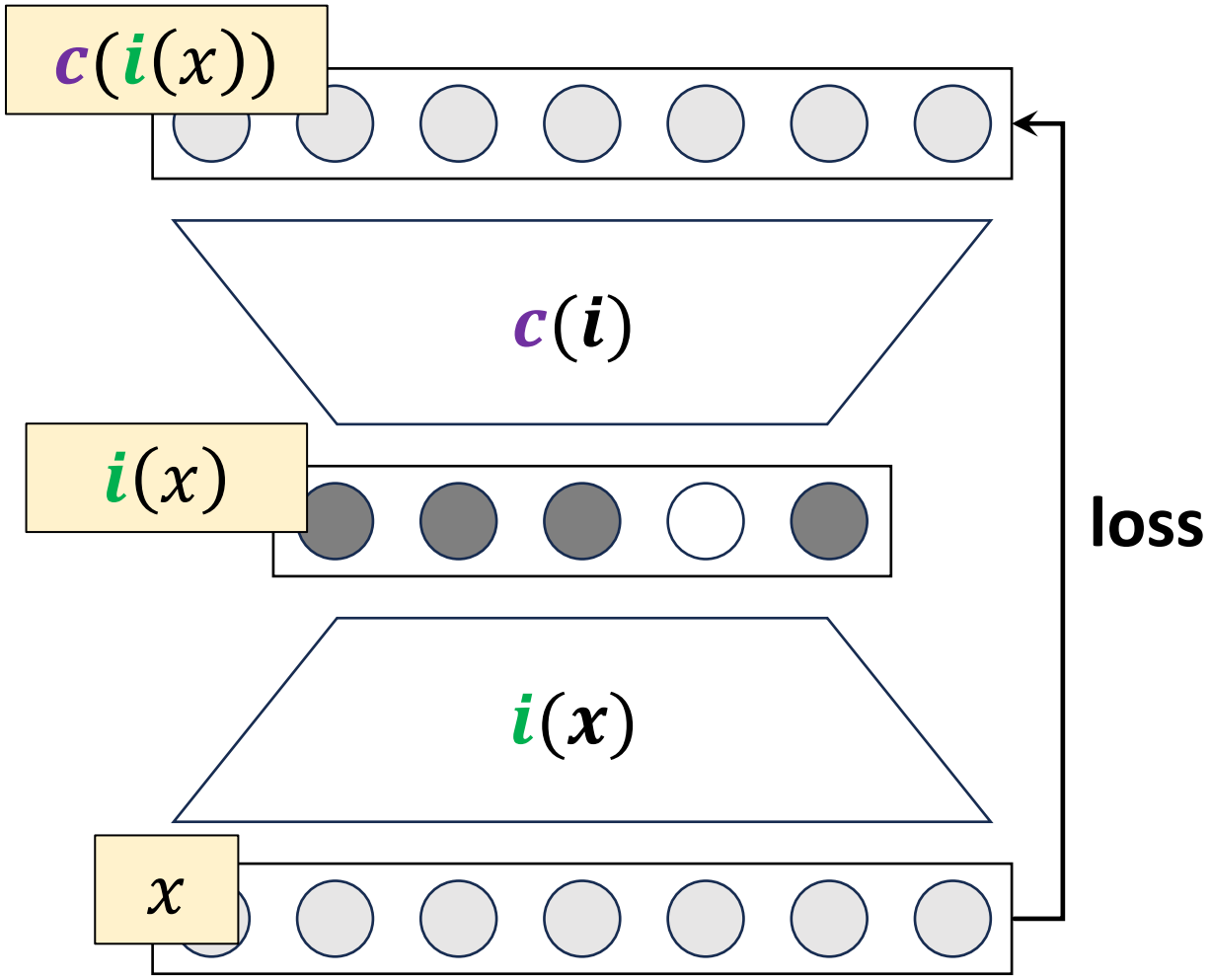
Loss function:

$$\min_{\boldsymbol{c},\boldsymbol{i}} \sum_{x} \|x - \boldsymbol{c}(\boldsymbol{i}(x))\|^2$$

Assignment: fixed $\boldsymbol{c}$, optimize $\boldsymbol{i}$

Update: fixed $\boldsymbol{i}$, optimize $\boldsymbol{c}$

Repeat until converge



K-means clustering

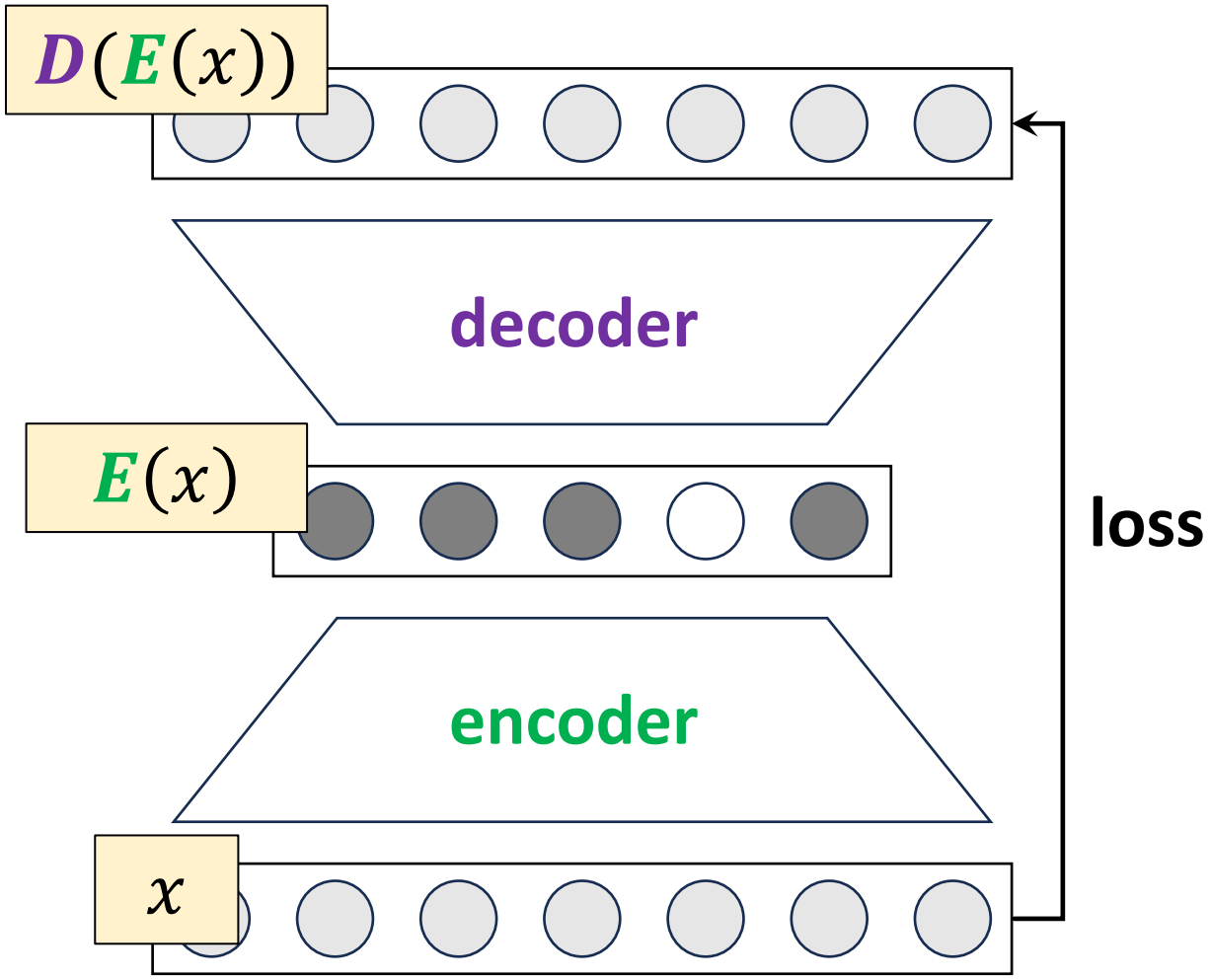
Loss function:

$$\min_{\boldsymbol{D},\boldsymbol{E}} \sum_{x} \|x - \boldsymbol{D}(\boldsymbol{E}(x))\|^2$$

encoder $\boldsymbol{E}$: map x to a (one-hot) code

decoder $\boldsymbol{D}$: map a code to a center

K-means is like Autoencoding.



Example: K-means on MNIST digits

These are not actual digit images.

These are the learned "representations".

Visualized centers with k = 20

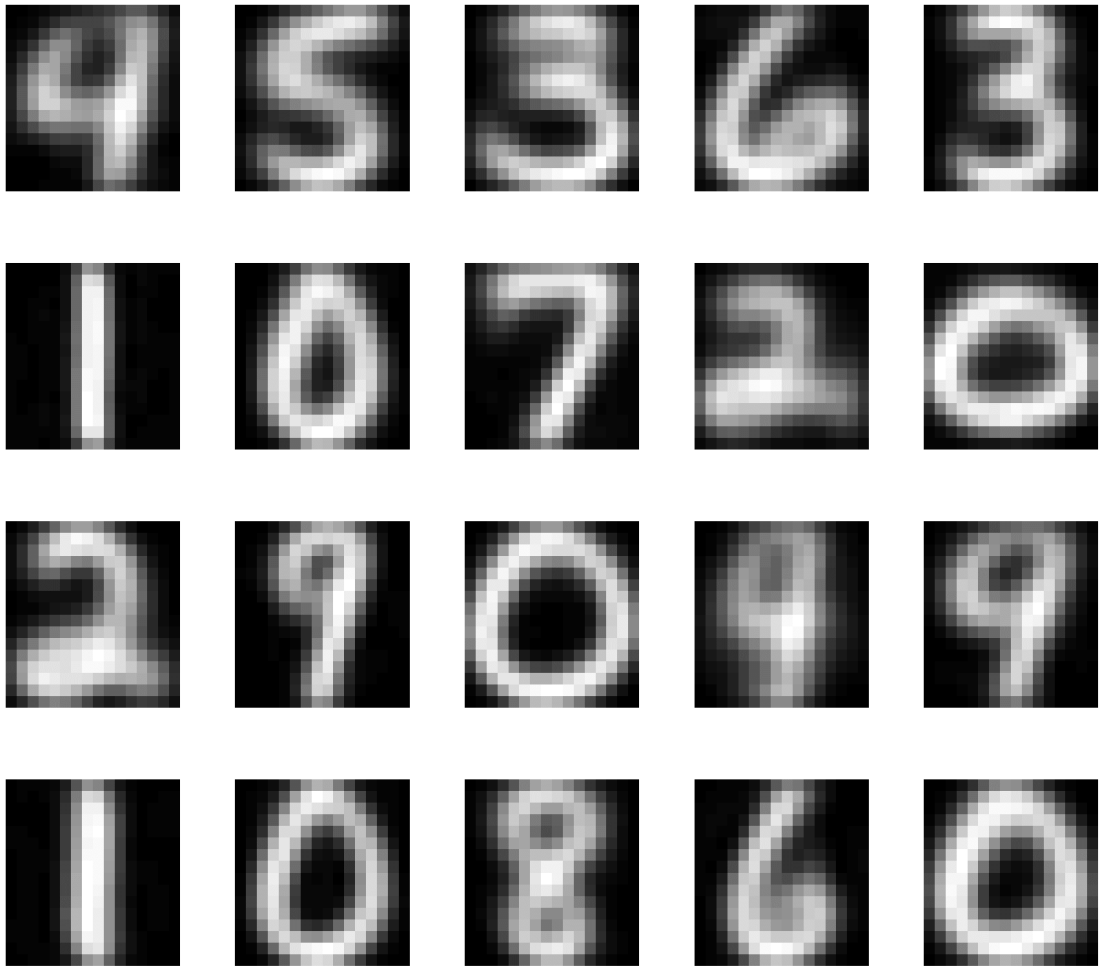
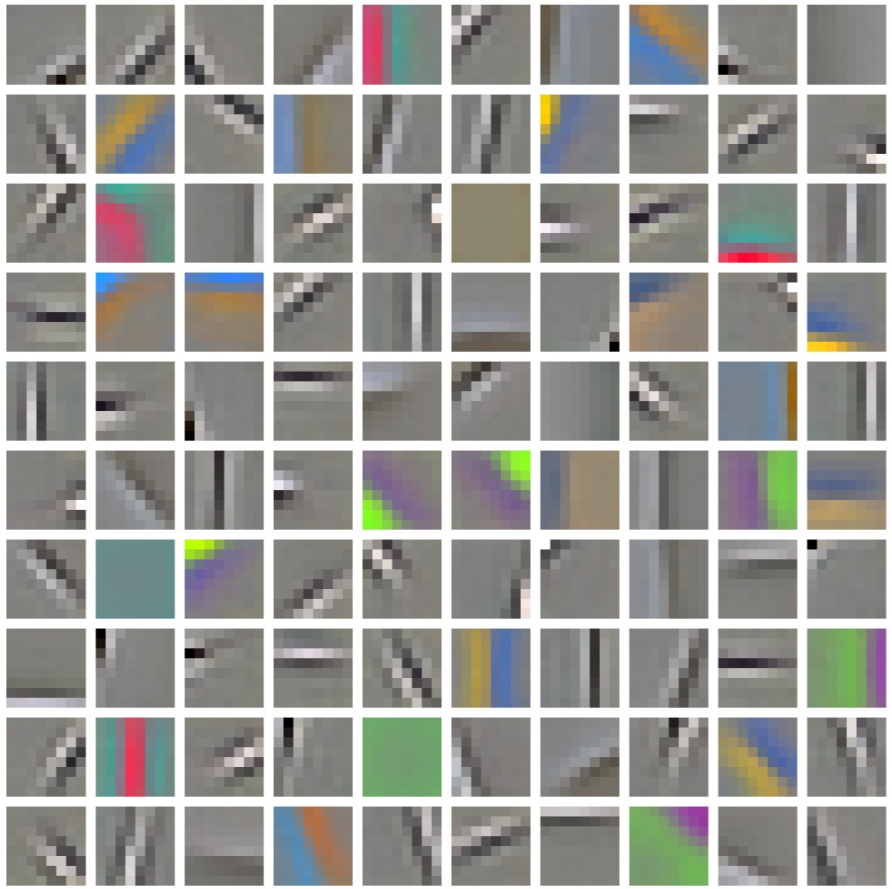
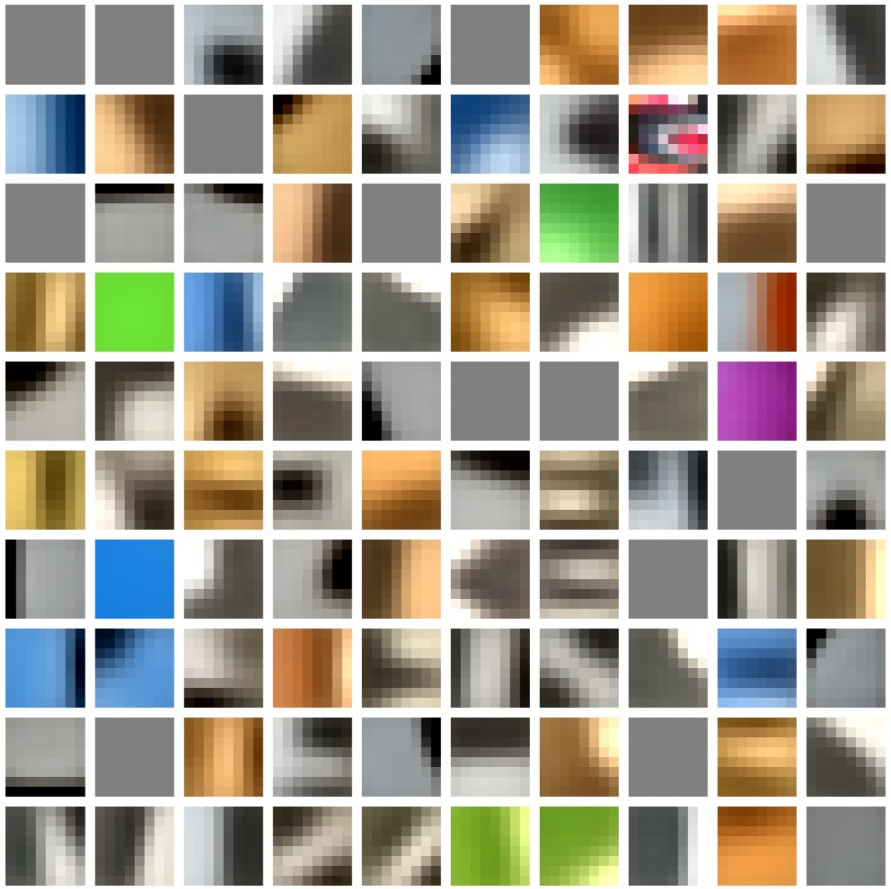


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

Example: K-means on 8×8 image patches



w/ PCA whitening



w/o PCA whitening

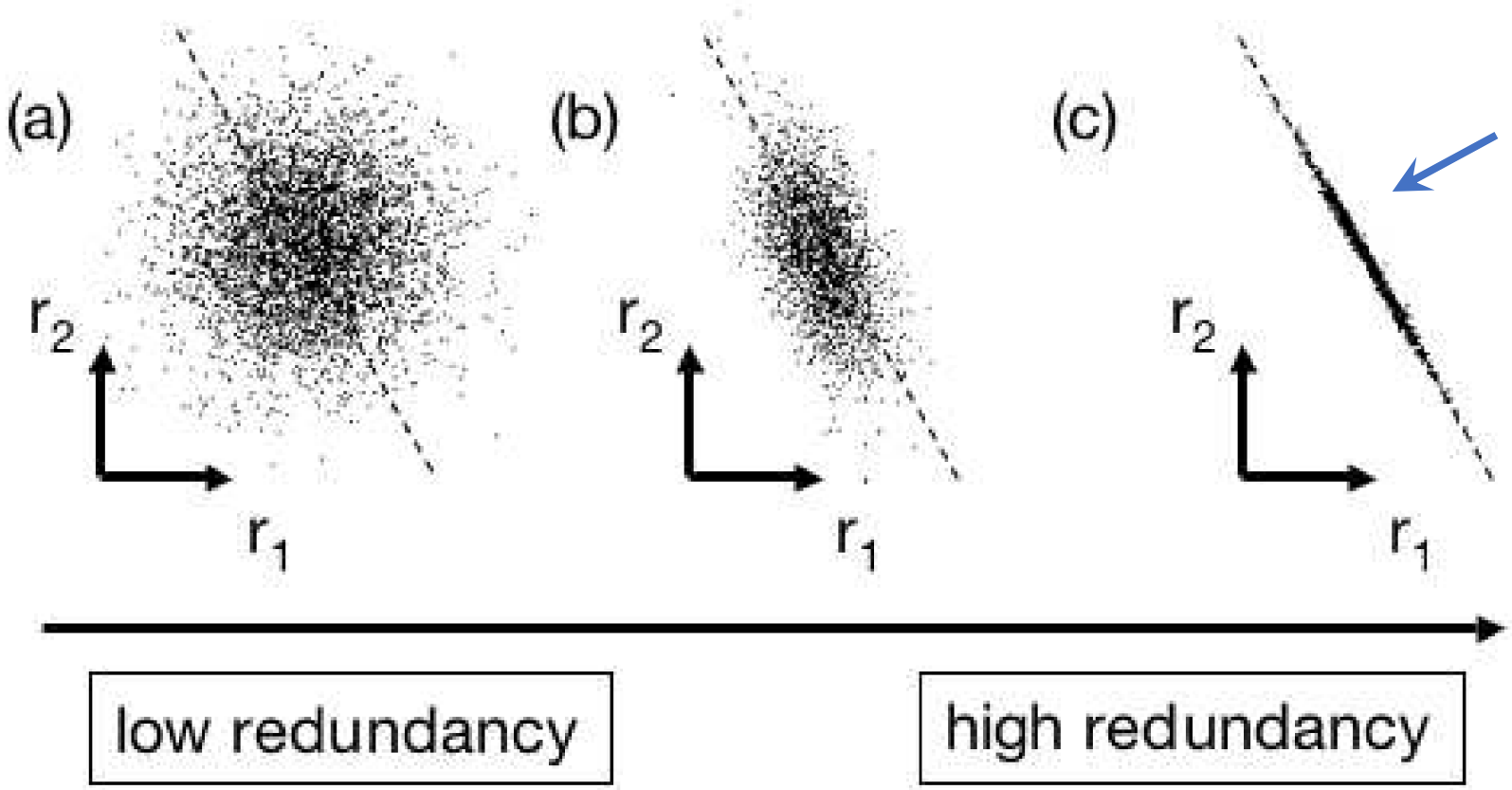
"An Analysis of Single-Layer Networks in Unsupervised Feature Learning", Coates, Lee, Ng. AISTATS 2011

Unsupervised Learning

Classical (but very useful) unsupervised learning methods:

- K-means clustering
- **Principal Component Analysis (PCA)**
- Autoencoder

Principal Component Analysis (PCA)



In this case, one can use r_1 to **represent** r_2 approximately.

So, (r_1 , r_2) is not a good representation.

Figure credit: Jonathon Shlens, "A Tutorial on Principal Component Analysis", 2005

Principal Component Analysis (PCA)

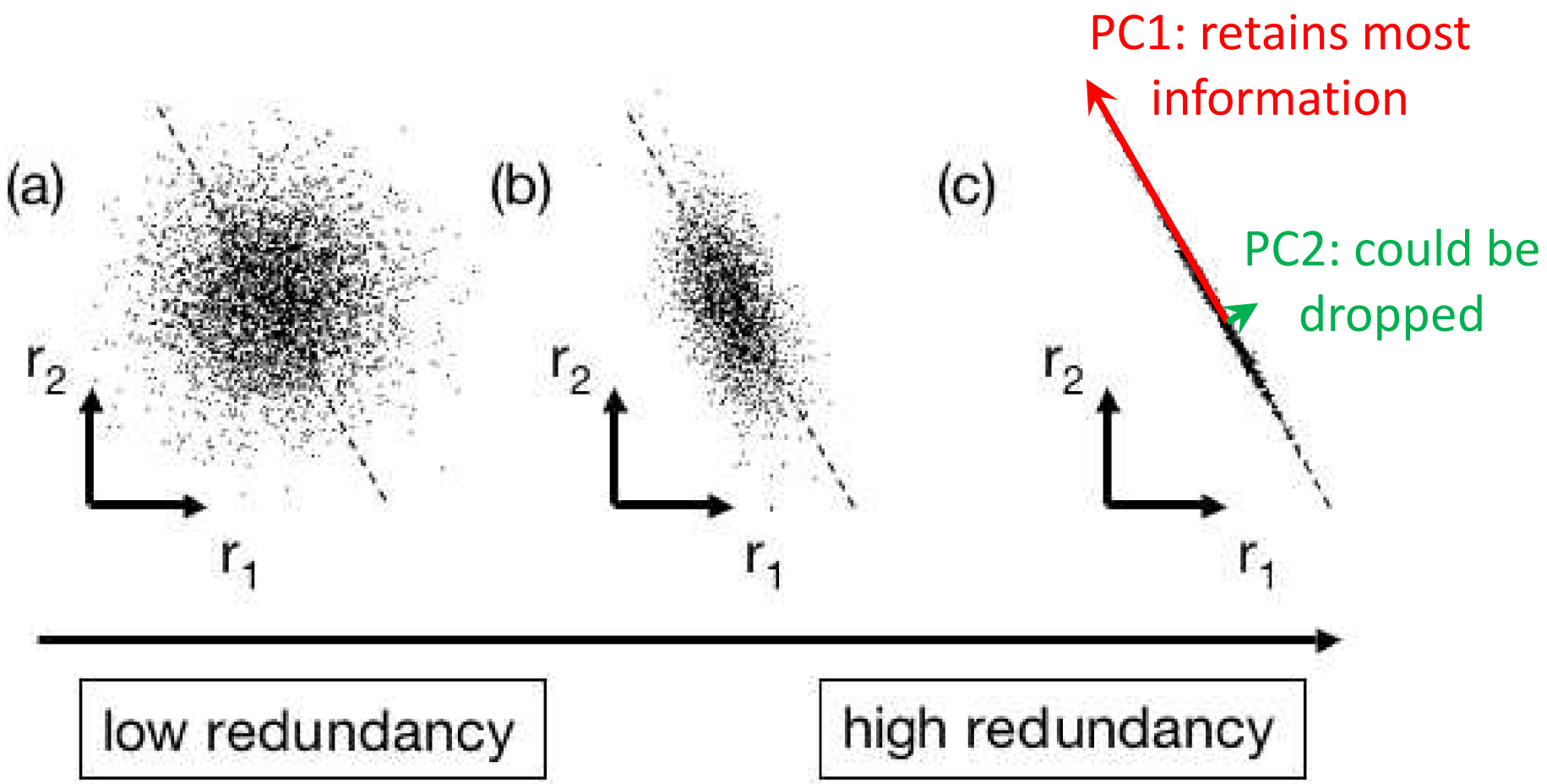


Figure credit: Jonathon Shlens, "A Tutorial on Principal Component Analysis", 2005

Principal Component Analysis (PCA)

Represent data in the space of the largest k Principal Components (PCs)

$$\min_{W} \sum_{x} \|x - W^{\mathrm{T}} W x\|^2$$
$$s.t.\ W W^{\mathrm{T}} = I_{k \times k}$$

x: d-dimensional vector
W: $k \times d$ matrix

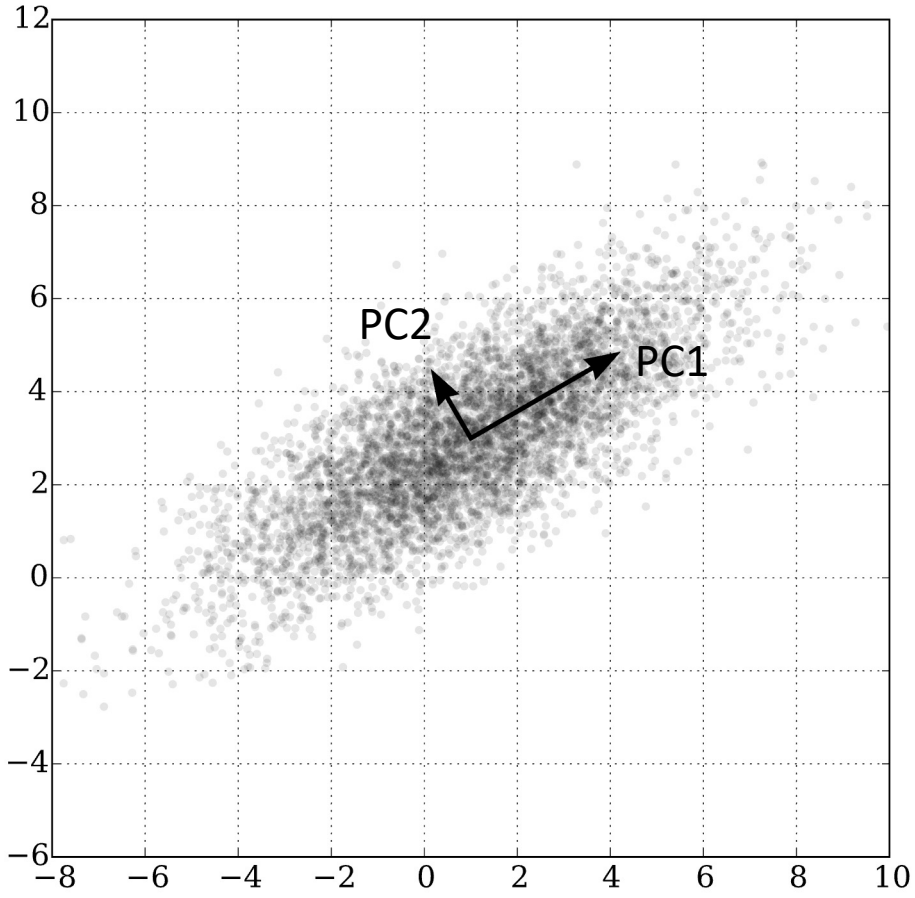


figure credit: https://en.wikipedia.org/wiki/Principal_component_analysis

Principal Component Analysis (PCA)

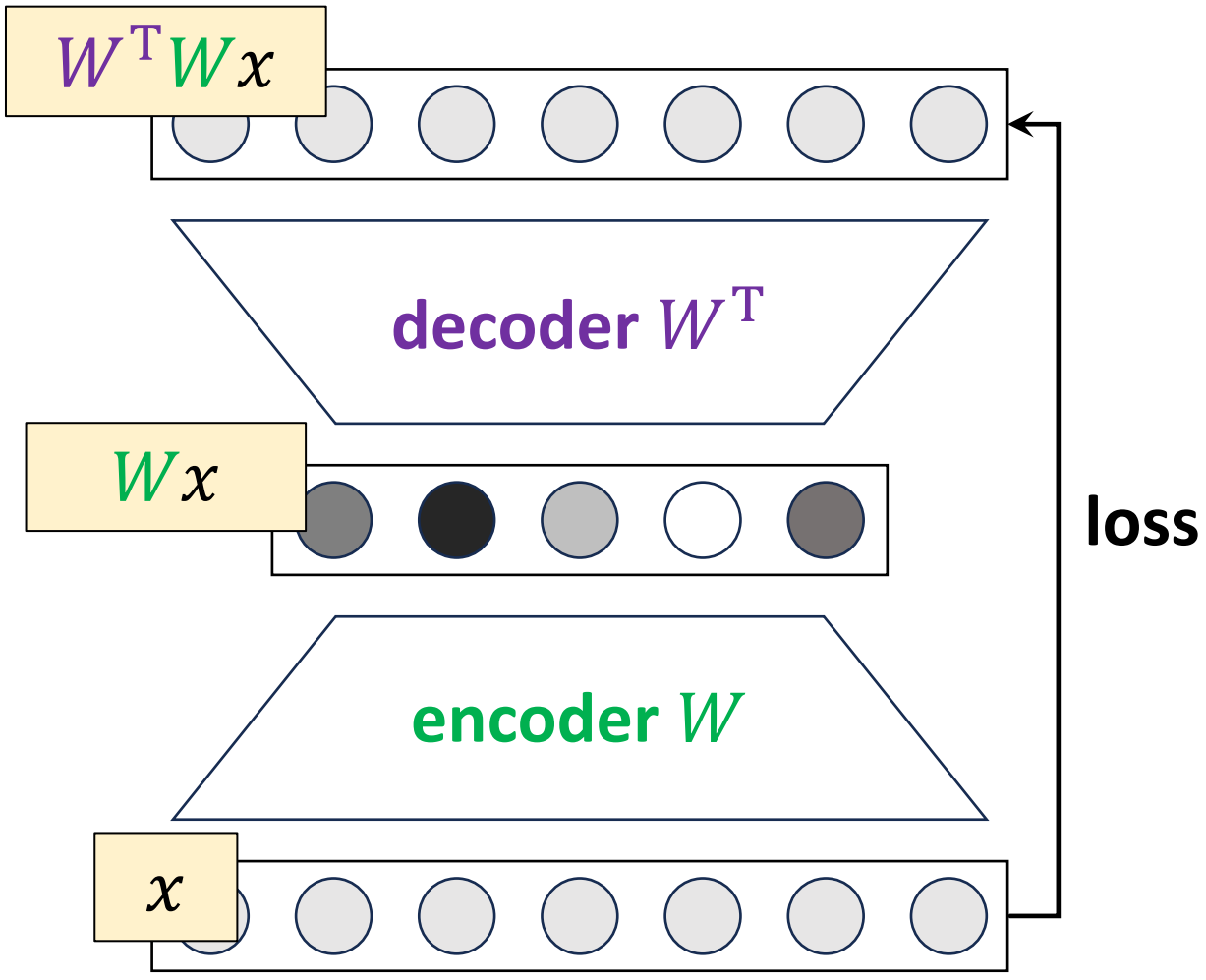
Represent data in the space of the largest k Principal Components (PCs)

$$\min_{W} \sum_{x} \|x - W^{\mathrm{T}} W x\|^2 \quad s.t.\ W W^{\mathrm{T}} = I_{k \times k}$$

encoder W: project onto PCs
decoder W^{T} : project back

PCA is like Autoencoding.

- w/ linear encoder & decoder
- w/ orthogonal constraint



Example: PCA on face data (Eigenfaces)

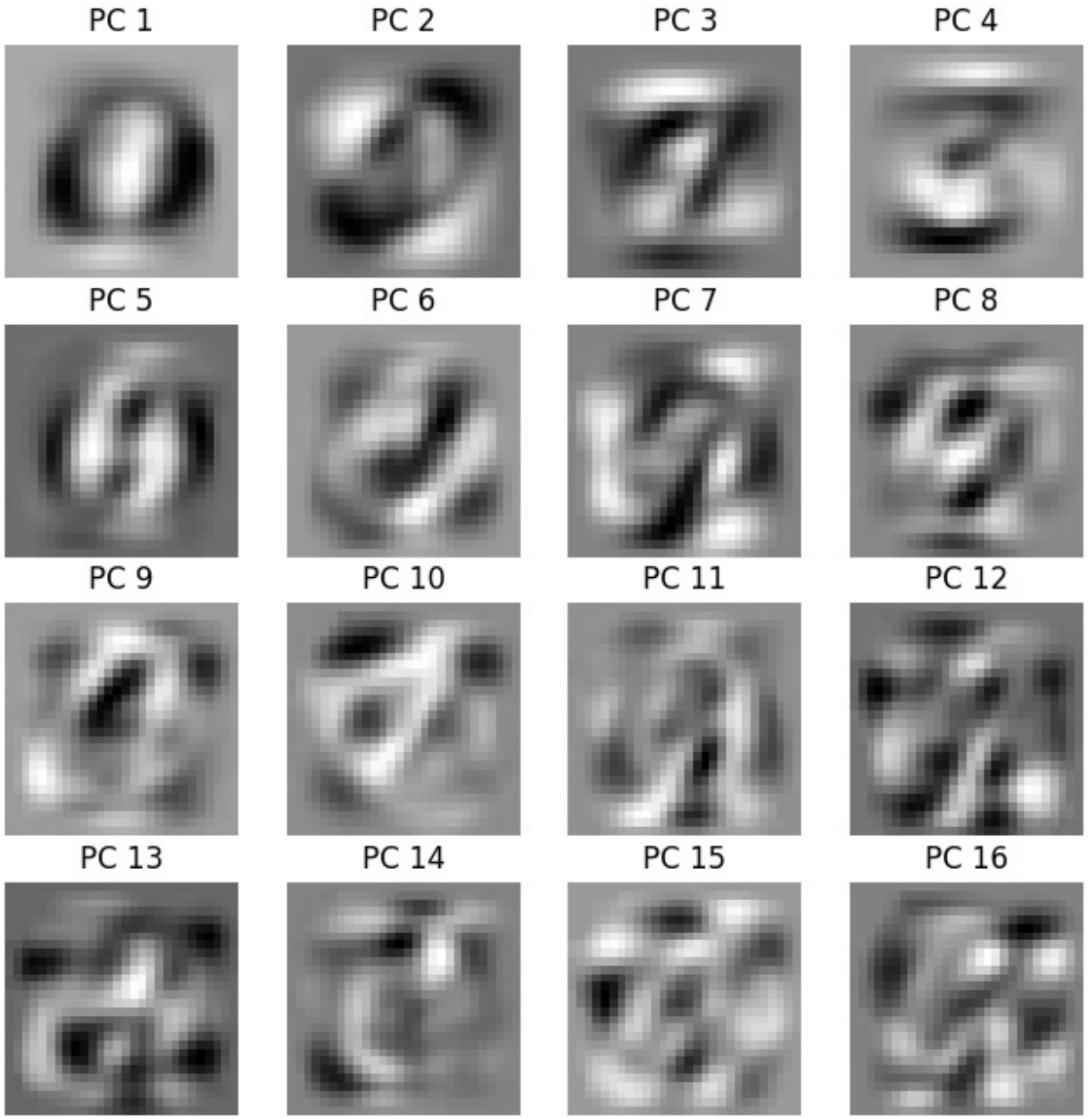
These are not actual face images.

These are the learned "representations".

figure credit: https://towardsdatascience.com/eigenfaces-recovering-humans-from-ghosts-17606c328184

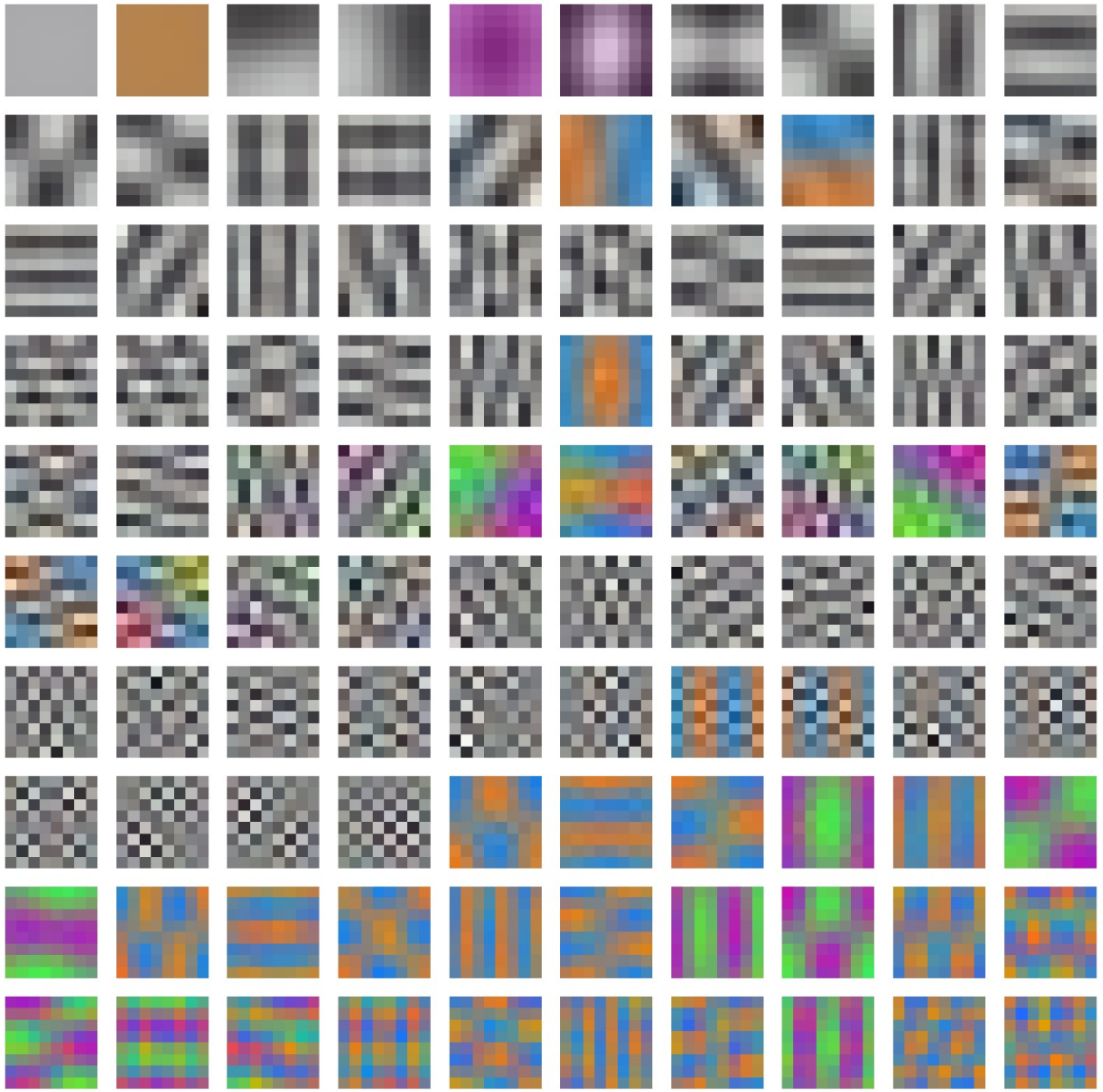
Turk and Pentland. "Face recognition using eigenfaces", CVPR 1991

Example: PCA on MNIST digits



Code from George Shakan: https://blog.georgeshakan.com/2023/08/06/singular-value-decomposition-and-pca/

Example: PCA on 8×8 image patches



Unsupervised Learning

Classical (but very useful) unsupervised learning methods:

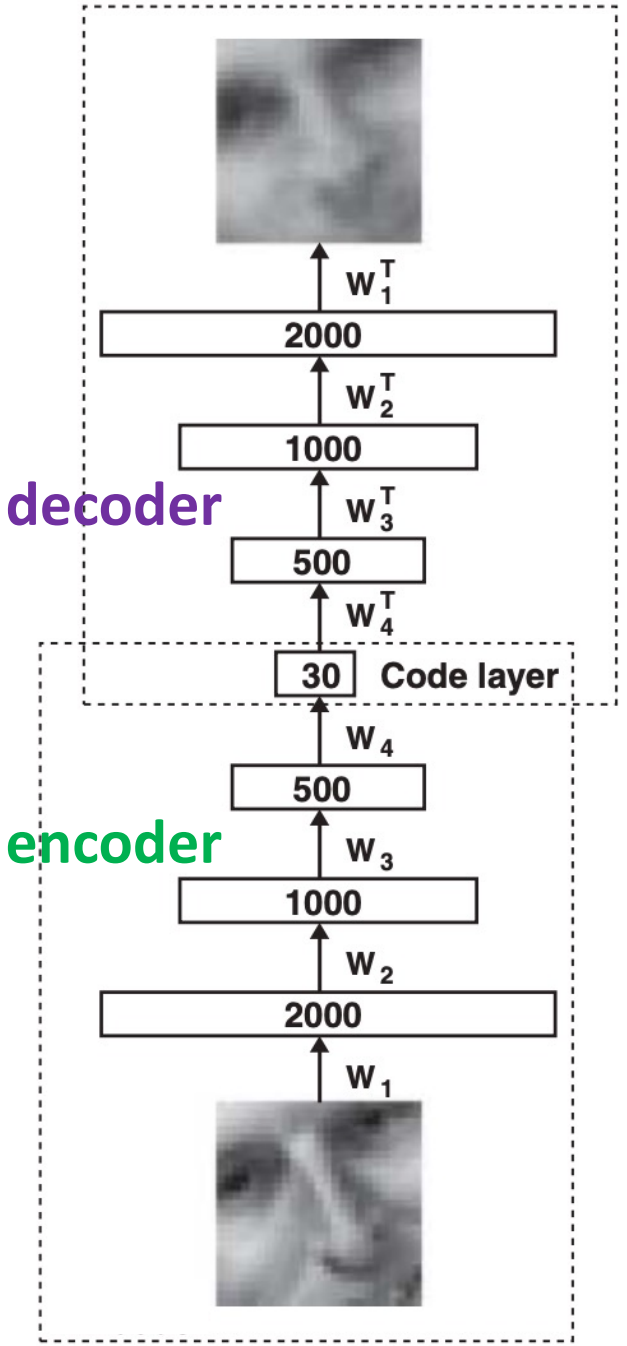
- K-means clustering
- Principal Component Analysis (PCA)
- **Autoencoder**

Autoencoder

Represent data by compressing it into a low-dimensional latent space

- **encoder** & **decoder**: deep neural nets
- optimized by BackProp
- a lower-dimensional latent as a "bottleneck"
- reconstruct the input

$$\min_{D,E} \sum_x \|D(E(x)) - x\|^2$$



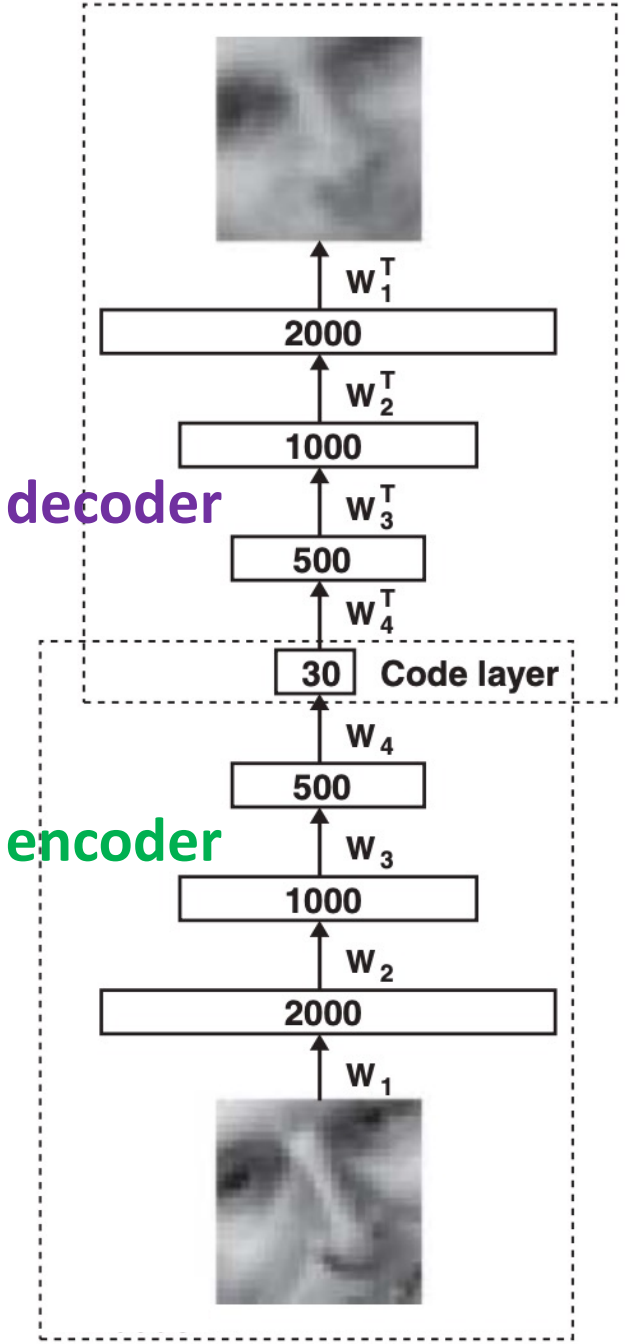
Hinton and Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks", Science 2006

Discussion

- K-means and PCA are like Autoencoding
- Autoencoding is “self-supervised learning”
  - “auto-” == “self-”
  - “encoding” ≈ “learning”

Using data “itself” as the supervision

$$\min_{D,E} \sum_{x} \|D(E(x)) - x\|^2$$



Self-supervised Learning

Self-supervised Learning

- **Self-supervised learning**: Using the data itself as supervision

- **Self-supervised learning** is a form of “unsupervised learning”

- In general, the goal is to:
  - Learn representations from **large-scale** unlabeled data
  - Learn **high-level** semantic representations

Self-supervised Learning

Modern self-supervised learning methods:

- Contrastive Learning
- Predictive Learning

Self-supervised Learning

Modern self-supervised learning methods:

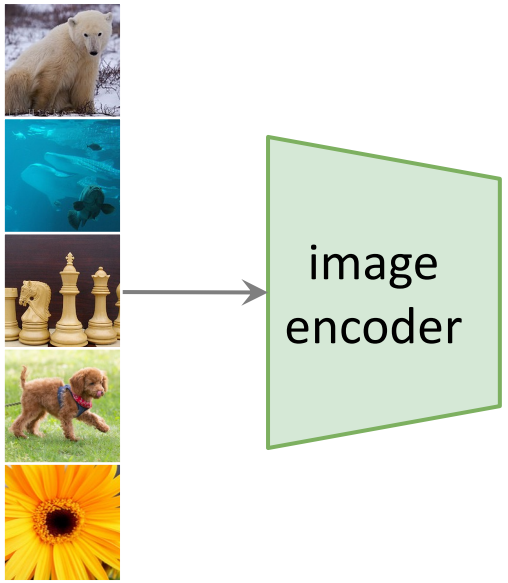
- **Contrastive Learning**
- Predictive Learning

Contrastive Learning

- **Supervised** classification is Contrastive Learning

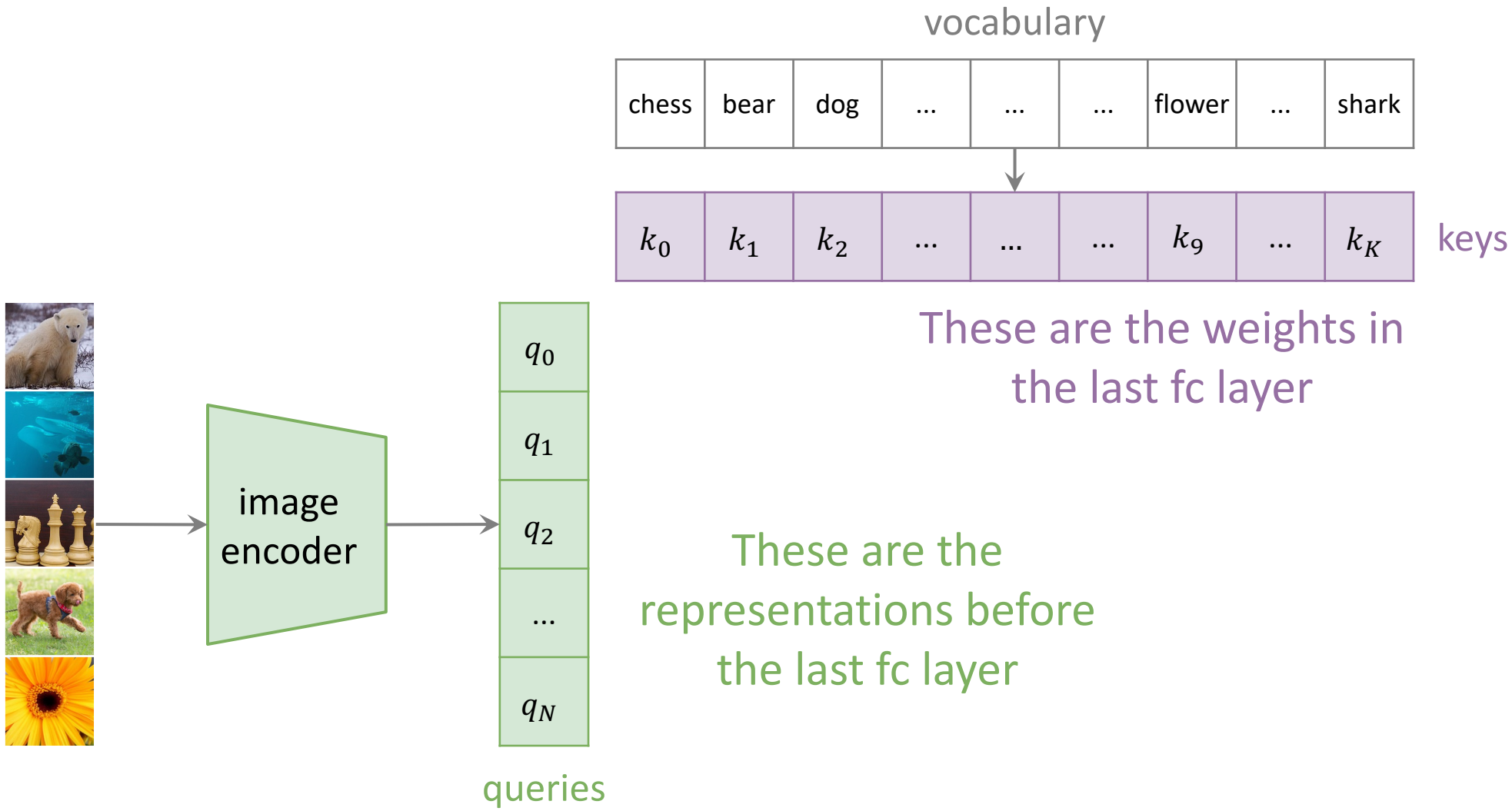
vocabulary

chess	bear	dog	...	...	...	flower	...	shark



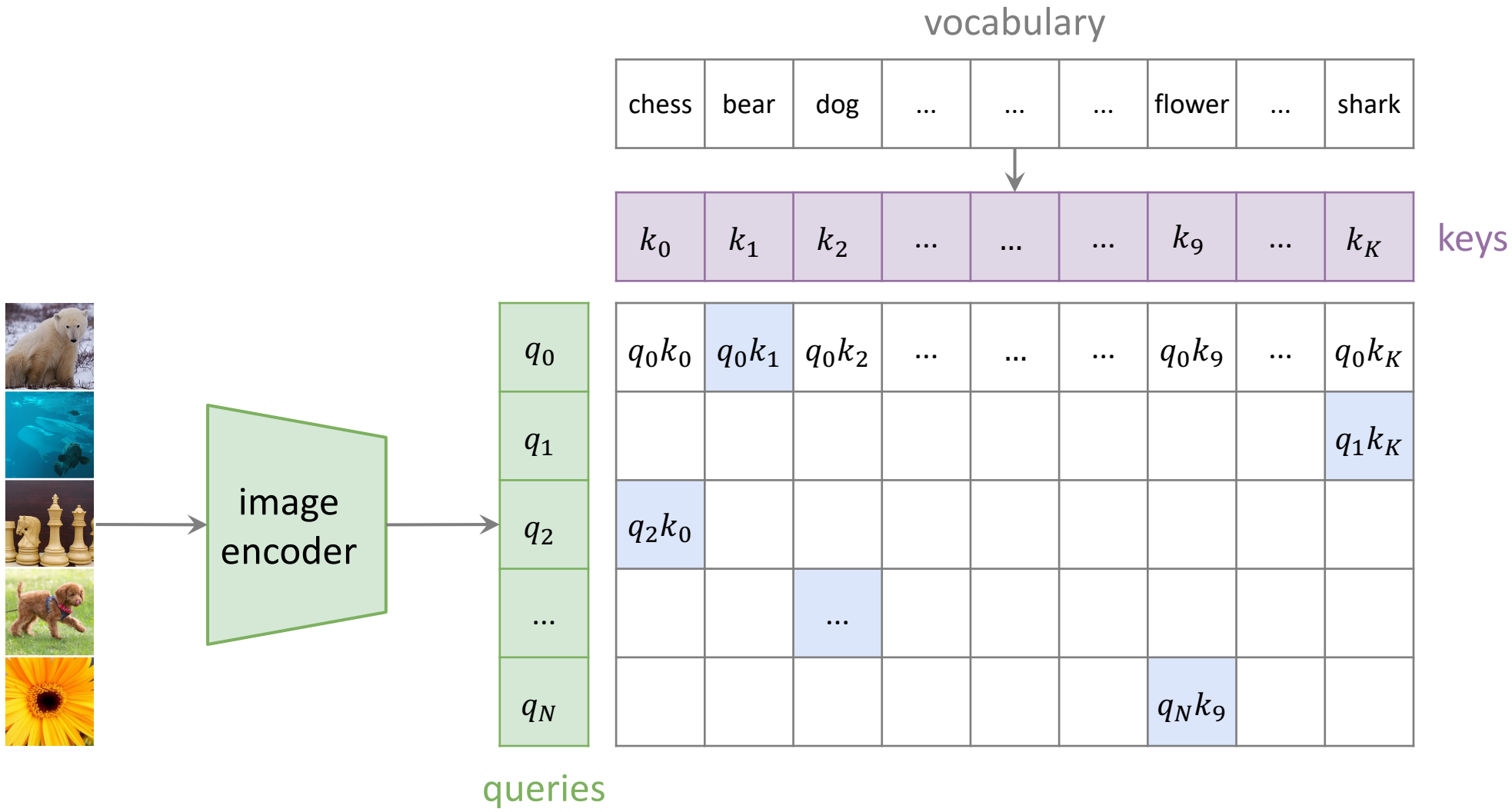
Contrastive Learning

- **Supervised** classification is Contrastive Learning



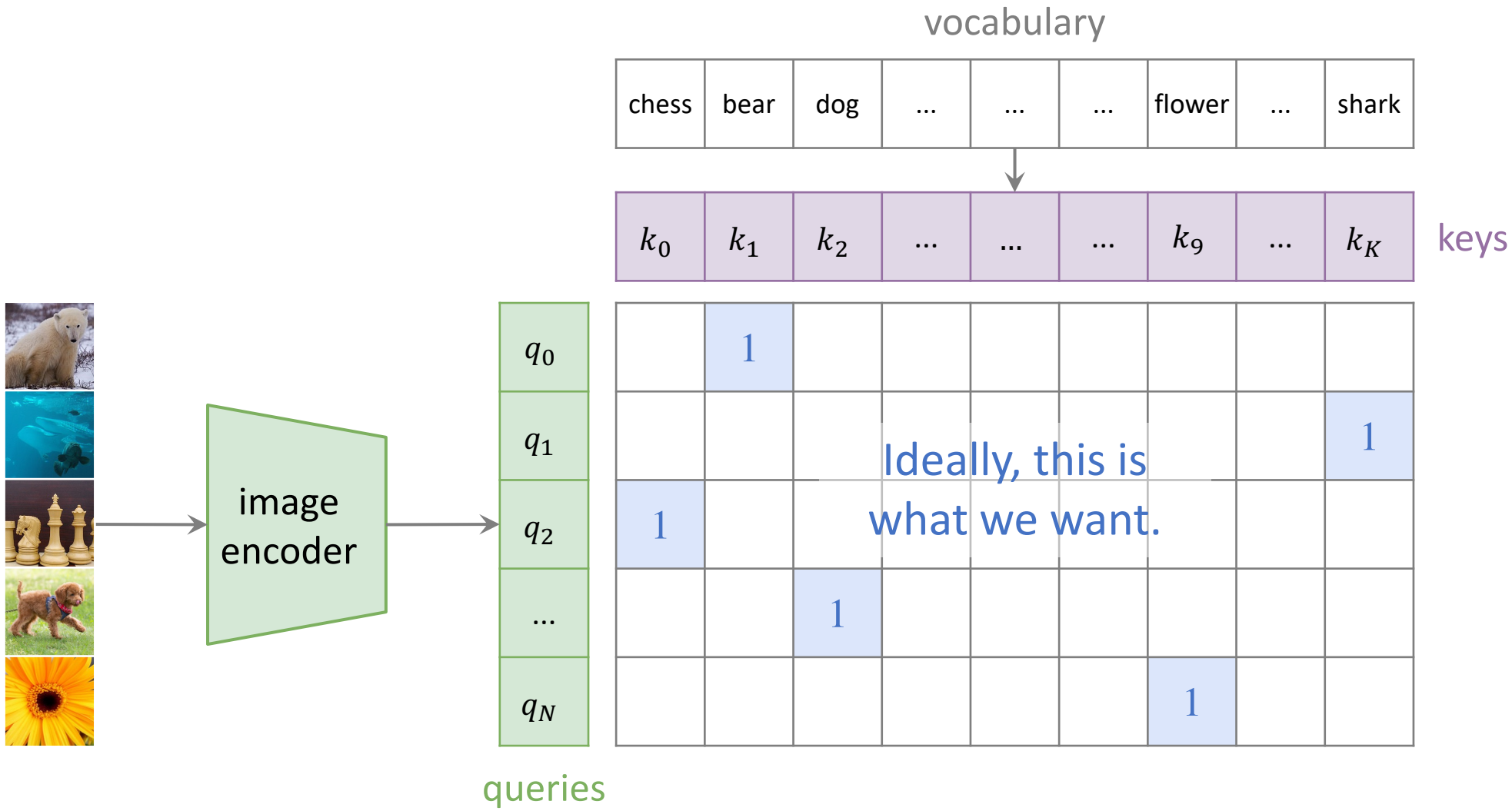
Contrastive Learning

- **Supervised** classification is Contrastive Learning



Contrastive Learning

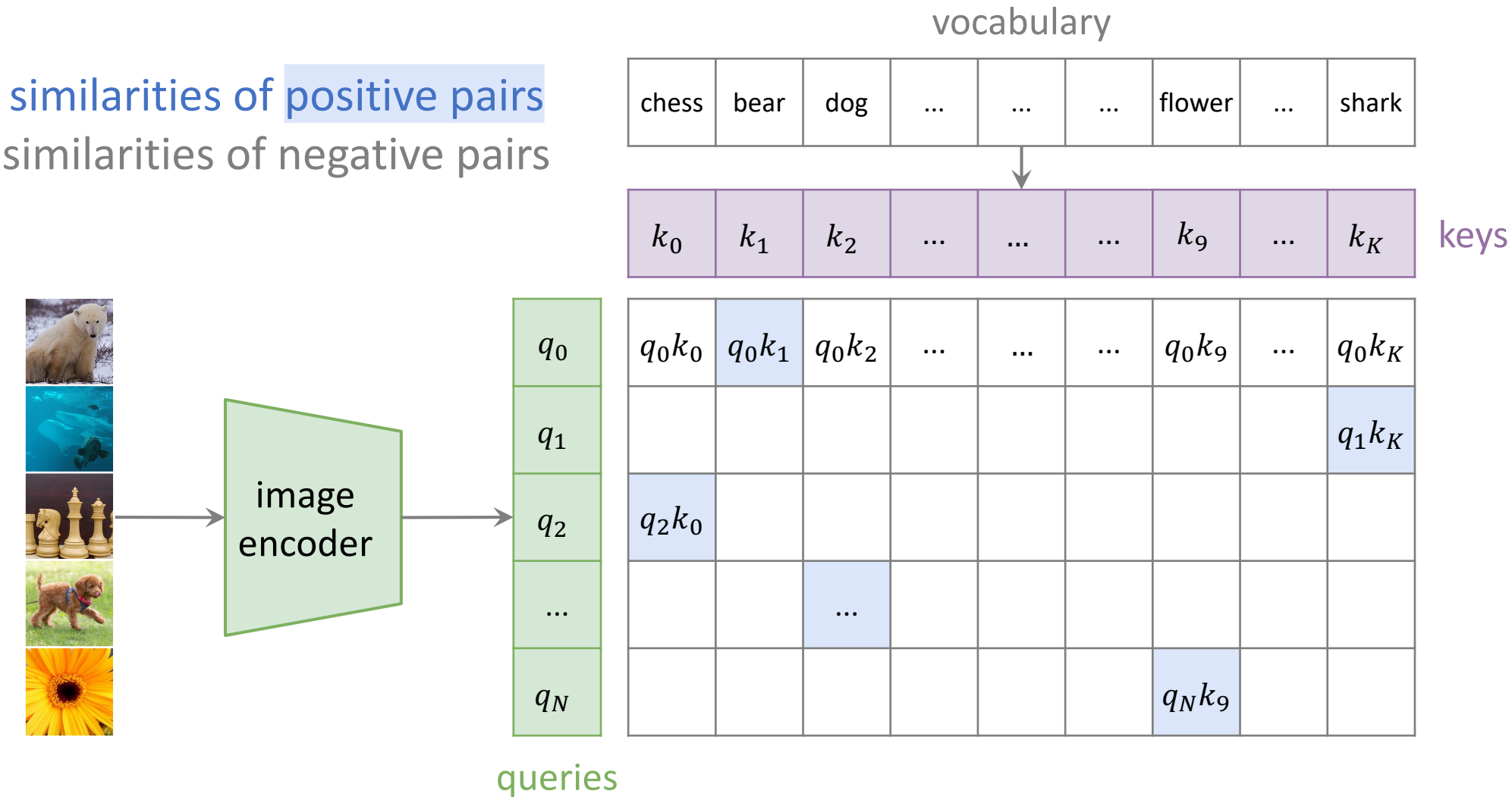
- **Supervised** classification is Contrastive Learning



Contrastive Learning

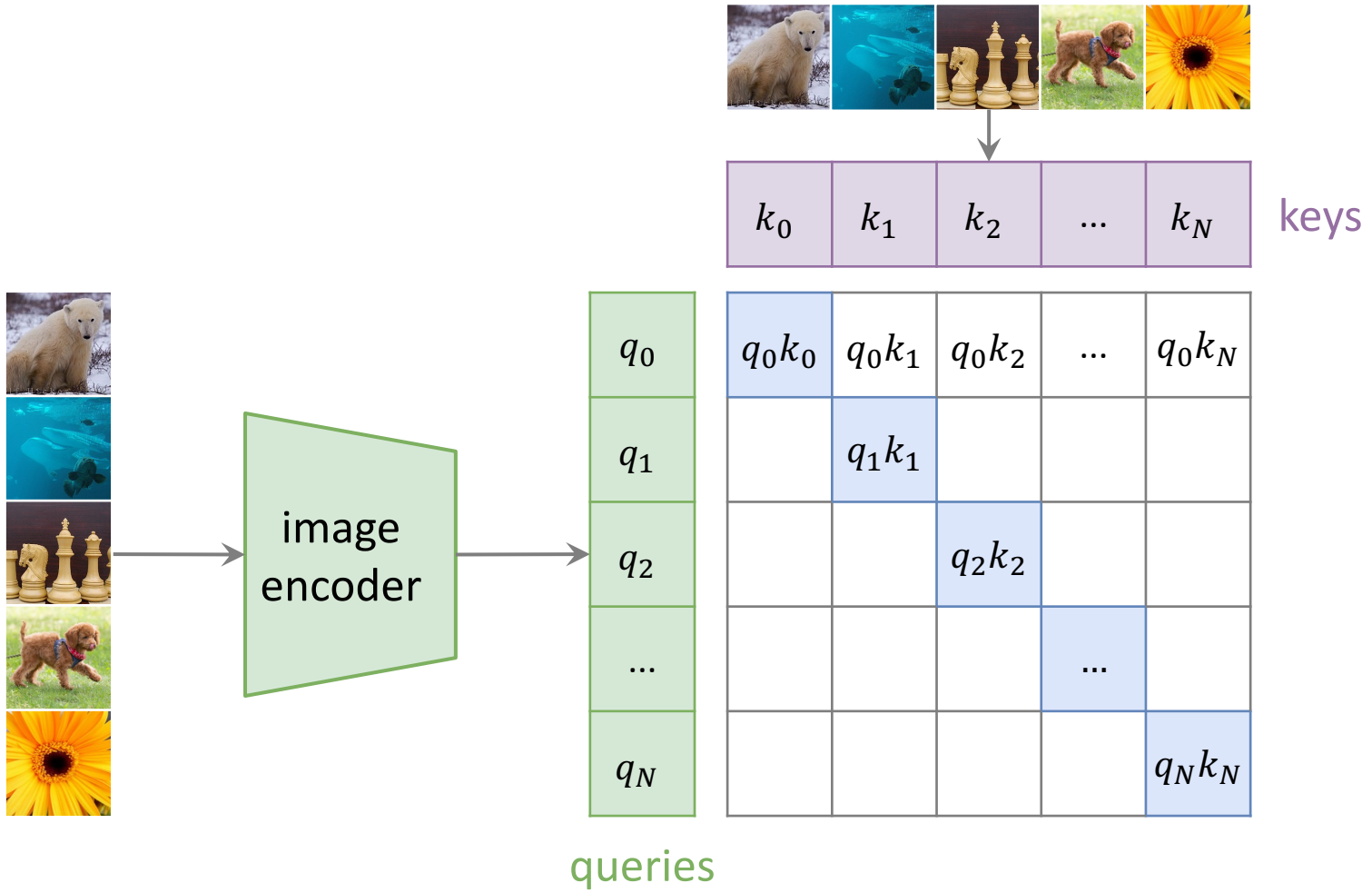
- **Supervised** classification is Contrastive Learning

  - maximize similarities of positive pairs
  - minimize similarities of negative pairs



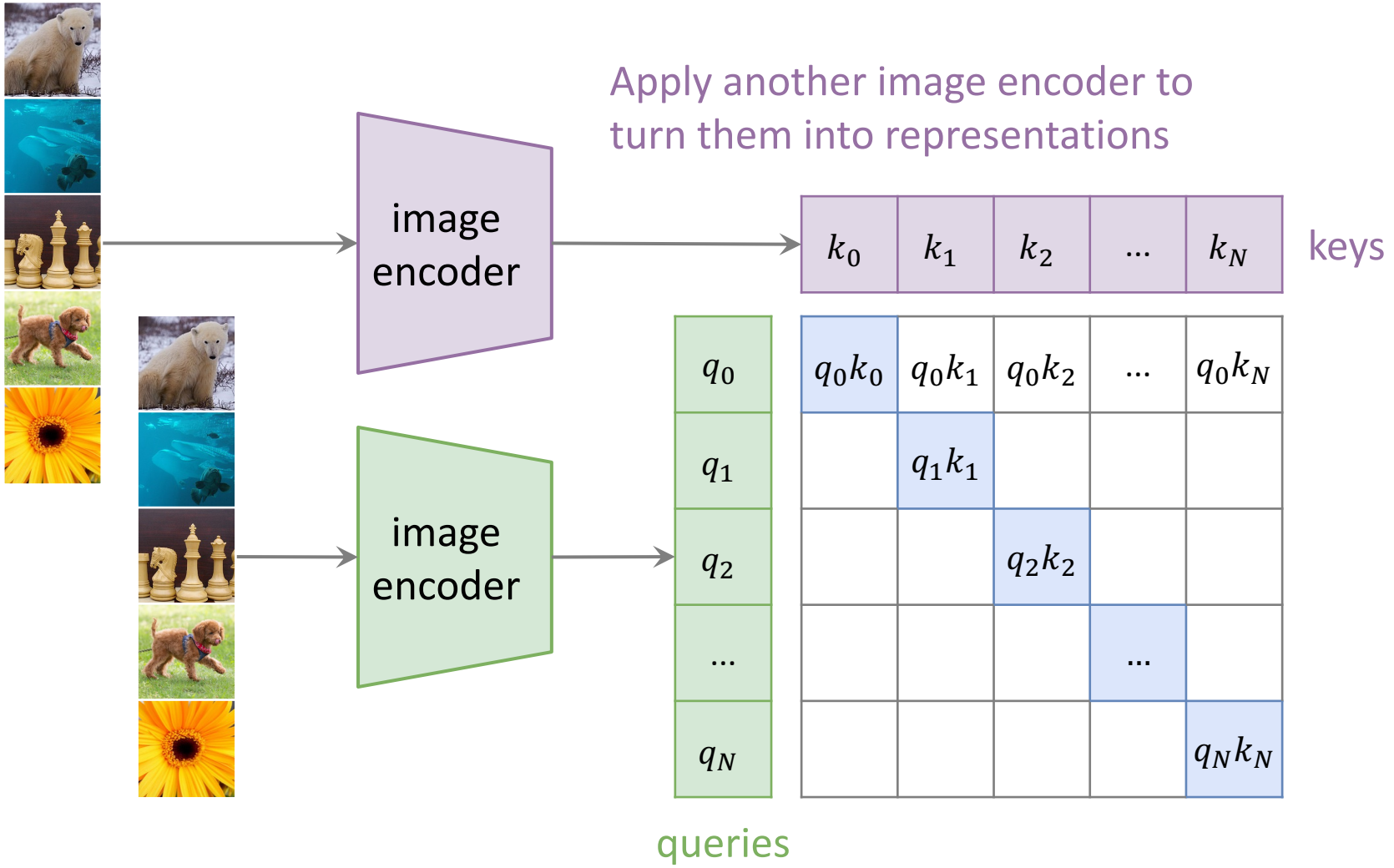
Contrastive Learning

- What if we don't have the vocabulary? Form a vocabulary w/ data **itself**



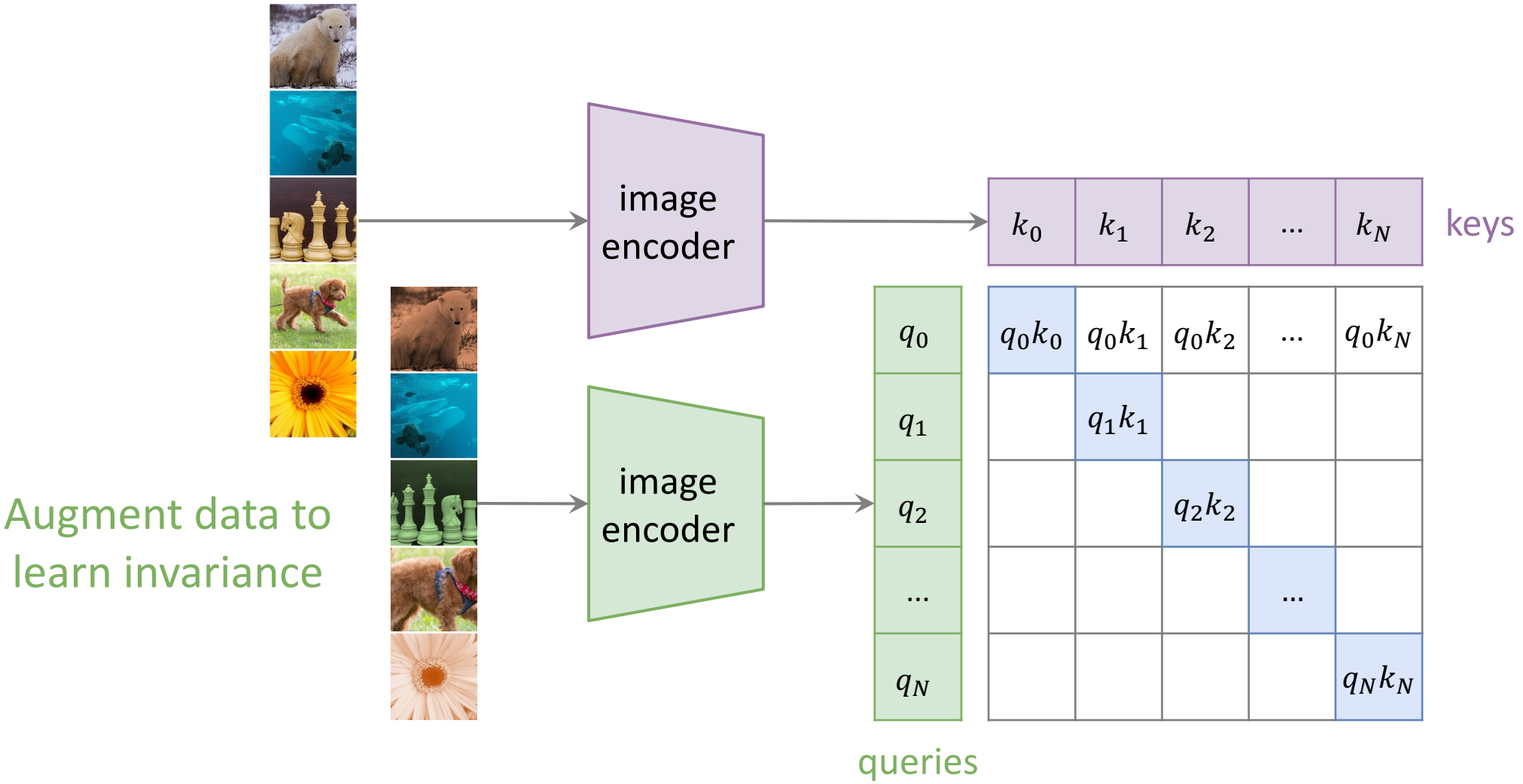
Contrastive Learning

- What if we don't have the vocabulary? Form a vocabulary w/ data **itself**



Contrastive Learning

- What if we don't have the vocabulary? Form a vocabulary w/ data **itself**

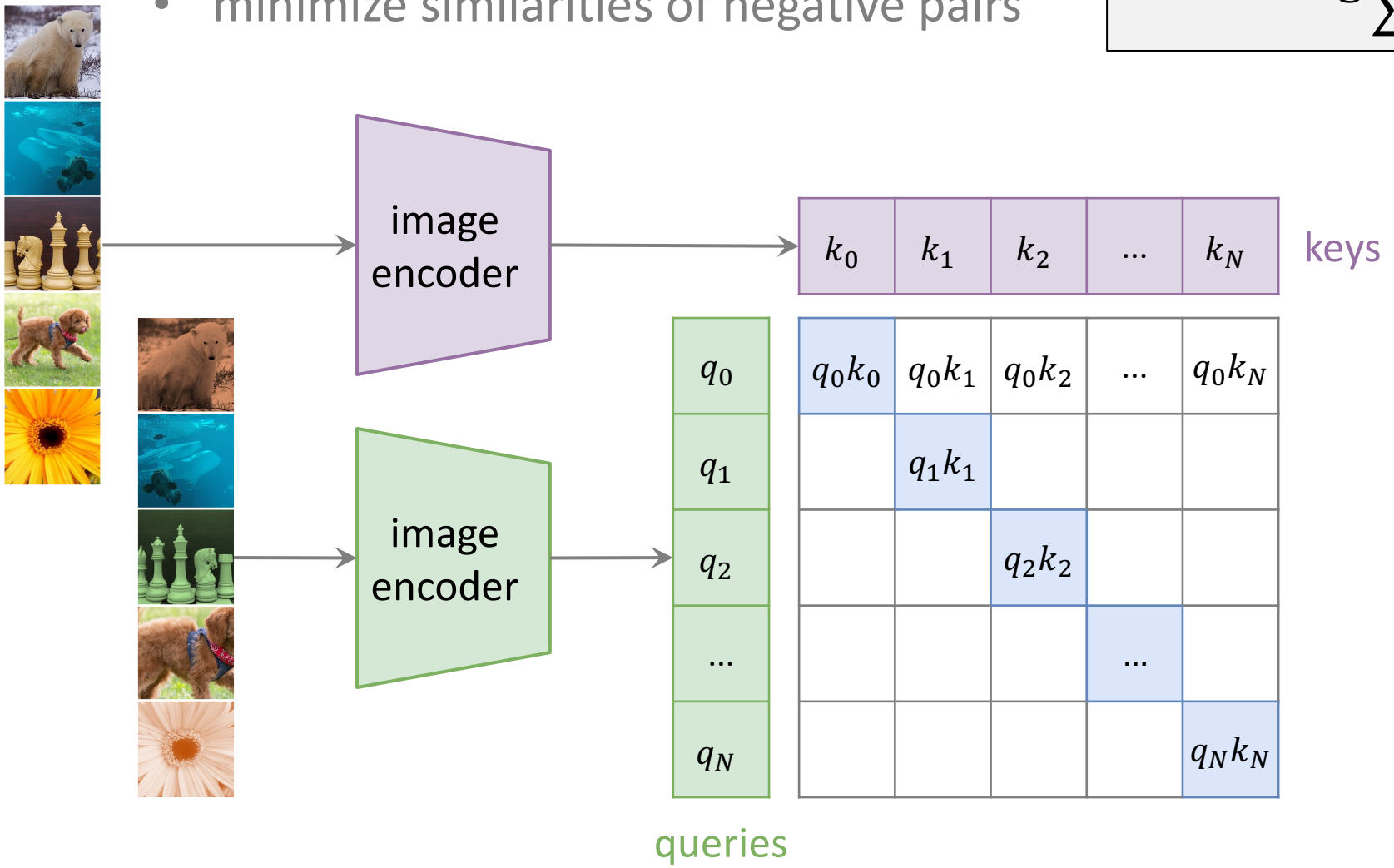


Contrastive Learning

- maximize similarities of positive pairs
- minimize similarities of negative pairs

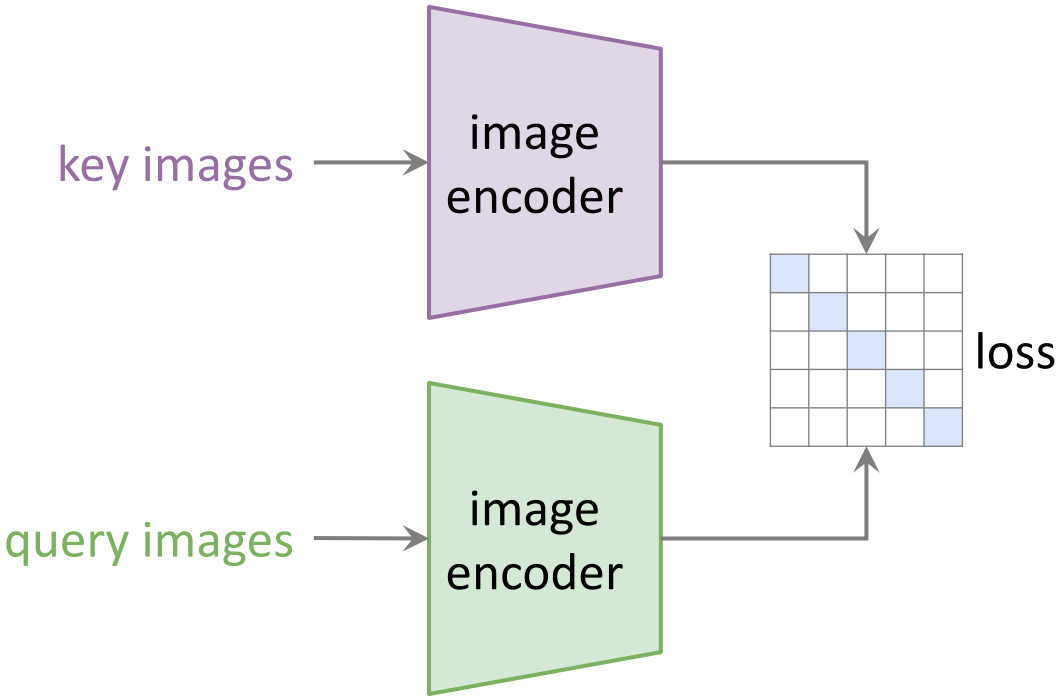
contrastive loss

$$L = -\log \frac{\exp(\boldsymbol{q}\boldsymbol{k}_+/\tau)}{\sum_i \exp(\boldsymbol{q}\boldsymbol{k}_i/\tau)}$$



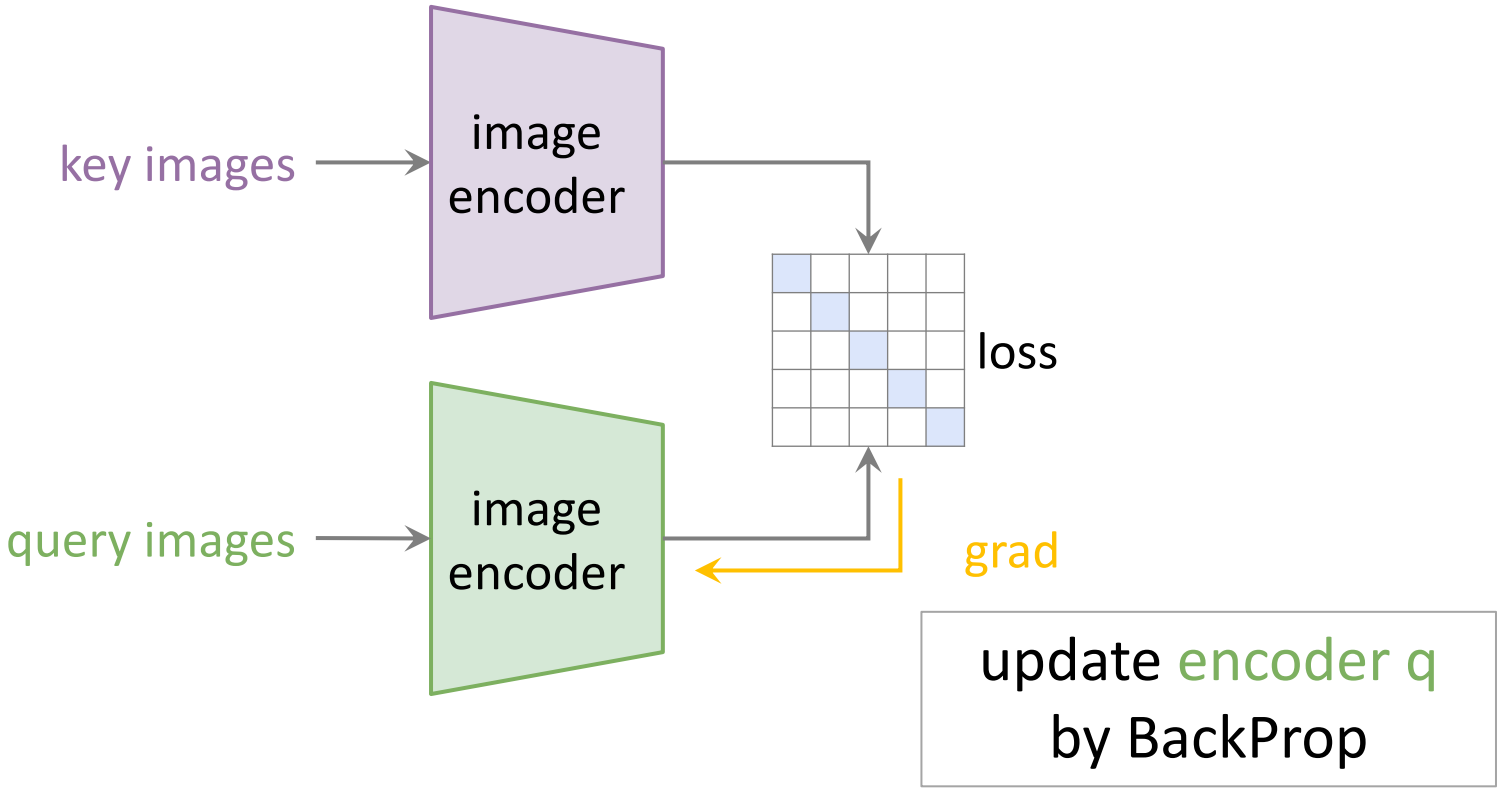
Contrastive Learning

- maximize similarities of positive pairs
- minimize similarities of negative pairs



Momentum Contrastive Learning (MoCo)

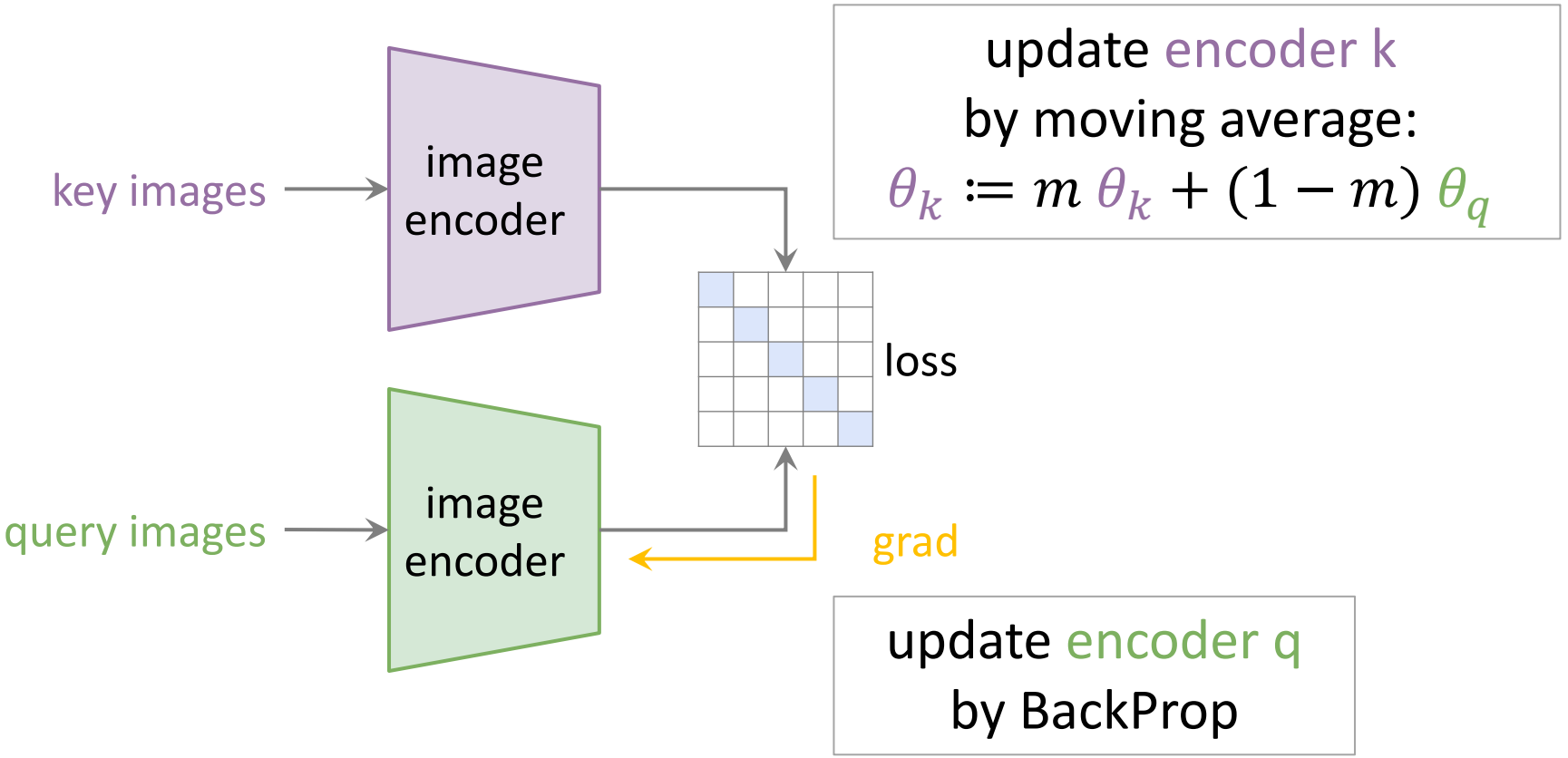
- maximize similarities of positive pairs
- minimize similarities of negative pairs



He et al. “Momentum Contrast for Unsupervised Visual Representation Learning”, CVPR 2020

Momentum Contrastive Learning (MoCo)

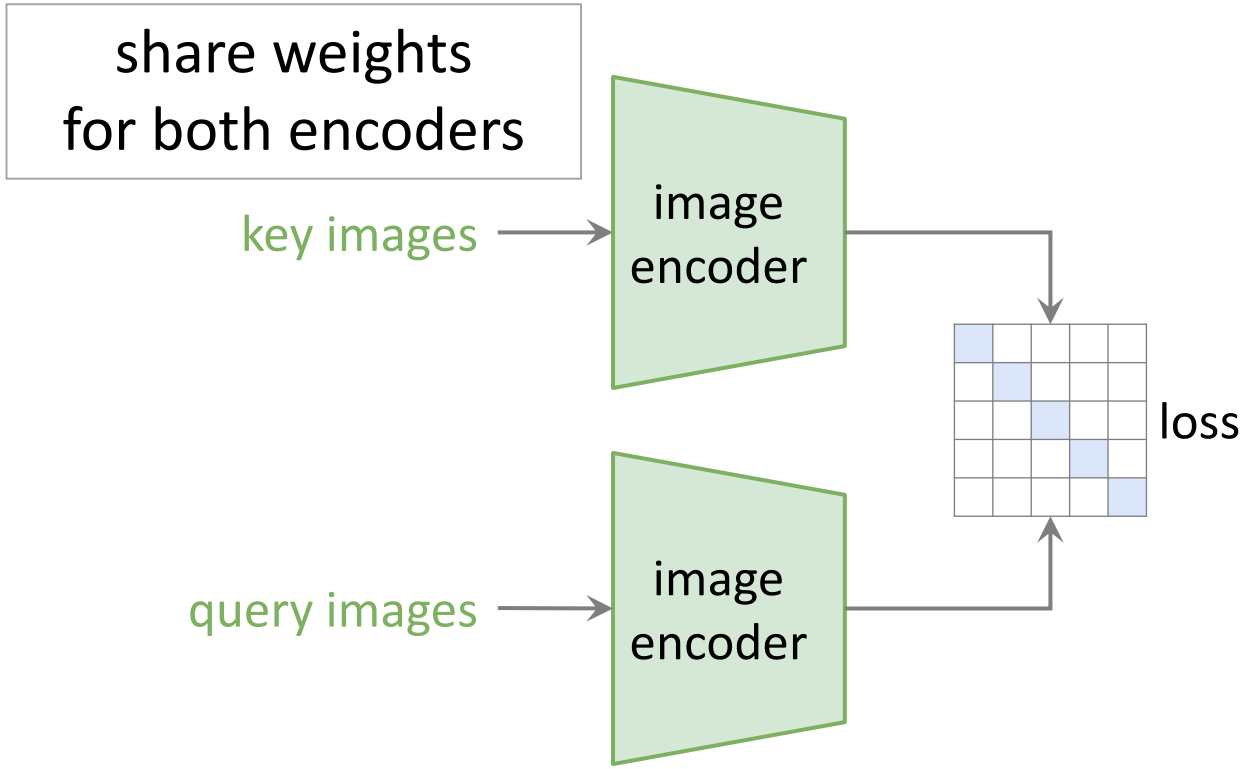
- maximize similarities of positive pairs
- minimize similarities of negative pairs



He et al. “Momentum Contrast for Unsupervised Visual Representation Learning”, CVPR 2020

Simple Contrastive Learning (SimCLR)

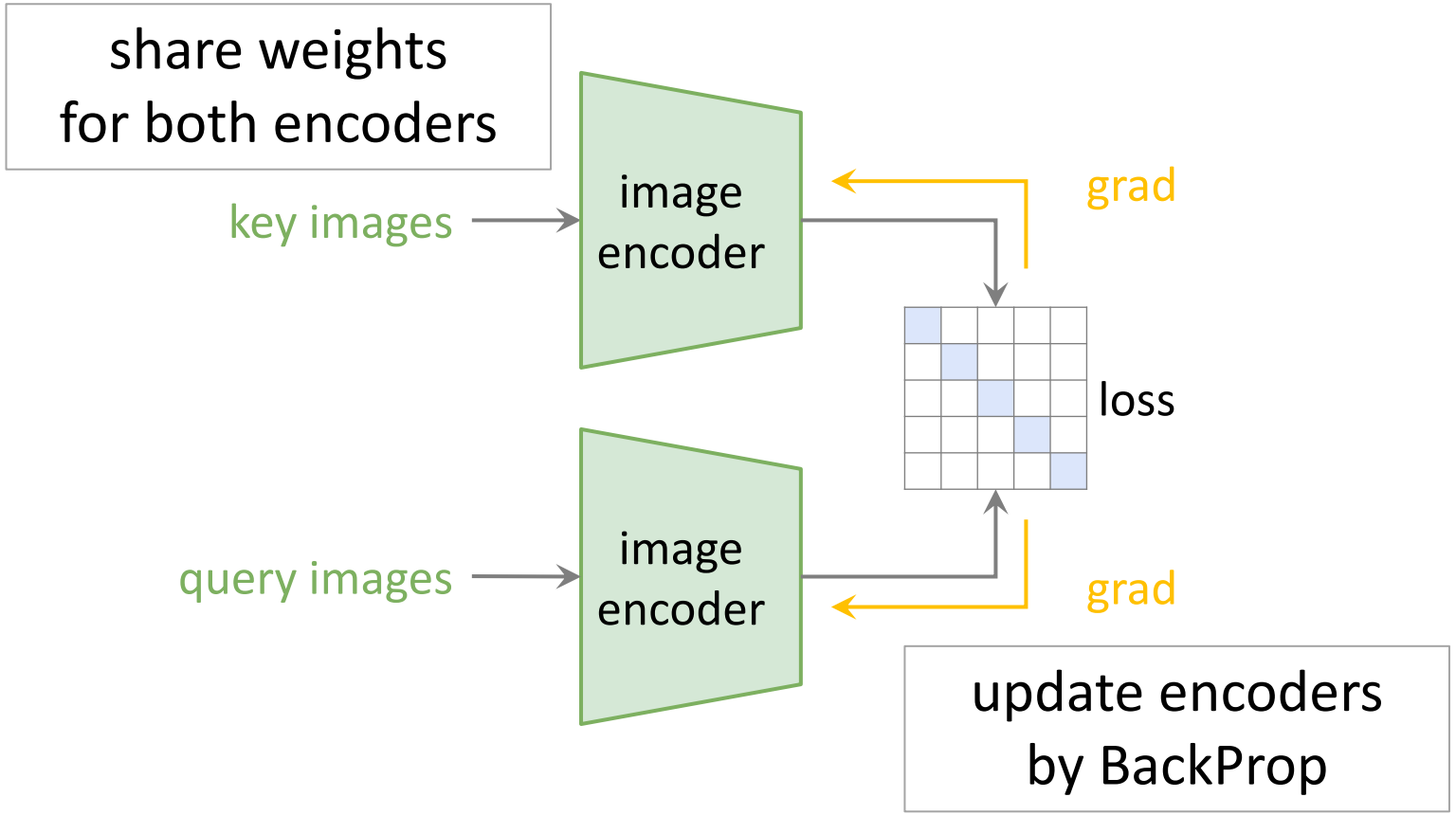
- maximize similarities of positive pairs
- minimize similarities of negative pairs



Chen et al. “A Simple Framework for Contrastive Learning of Visual Representations”, ICML 2020

Simple Contrastive Learning (SimCLR)

- maximize similarities of positive pairs
- minimize similarities of negative pairs



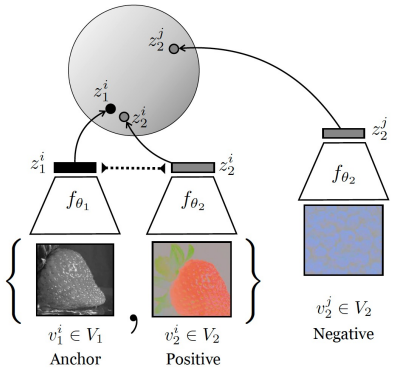
Chen et al. “A Simple Framework for Contrastive Learning of Visual Representations”, ICML 2020

Many More Variants

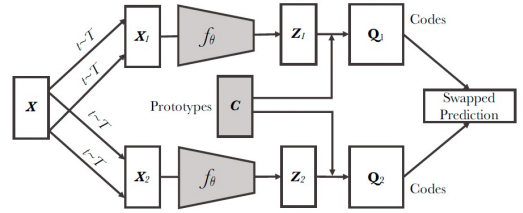
Compare samples in the latent space

- Different data augmentation
- Different loss functions
- w/ or w/o negative samples
- w/ or w/o momentum encoder

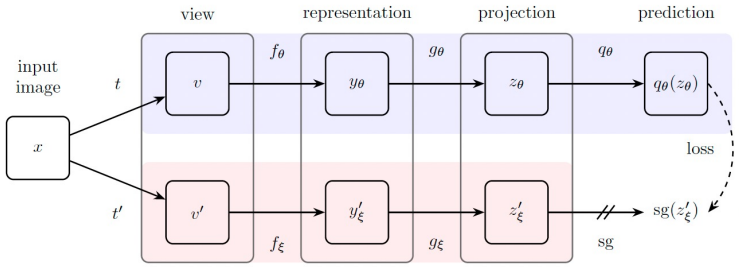
A frontier research direction



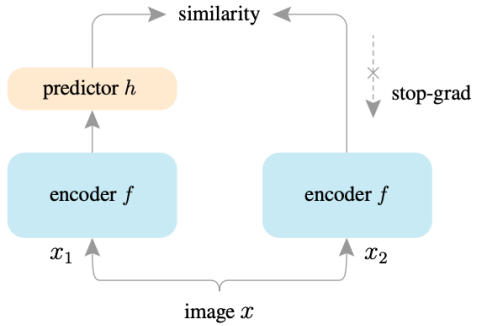
CMC



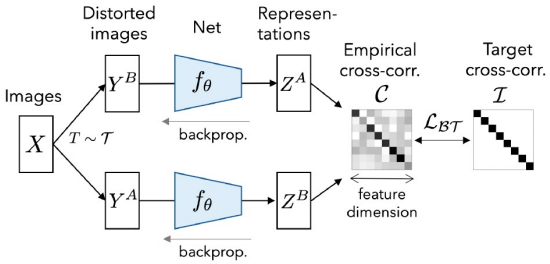
SwAV



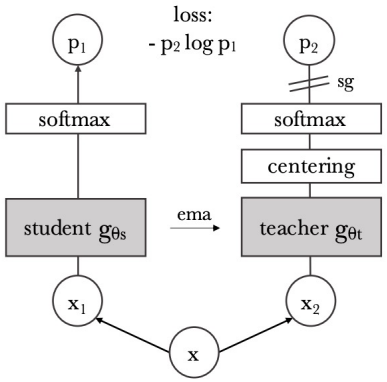
BYOL



SimSiam



Barlow Twins



DINO

Tian, ’19; Caron+, ’20; Grill+, ’20; X. Chen+, ’20; Zbontar+, ’21; Caron+, ’21;

Self-supervised Learning

Modern self-supervised learning methods:

- Contrastive Learning
- **Predictive Learning**

Predictive Learning: Language Models

Next word prediction (GPT)

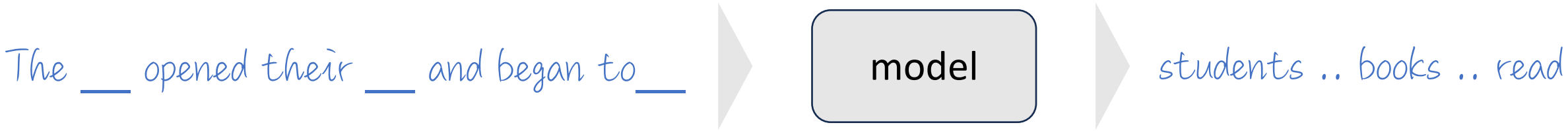
- Predict the next word (token) given a prefix

Radford, et al., “Improving Language Understanding by Generative Pre-Training”, 2018

Predictive Learning: Language Models

Masked language modeling (BERT)

- Predict the masked words (tokens) in a text



Devlin, et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, 2018

Predictive Learning: Computer Vision

Masked image modeling (Context Encoders)

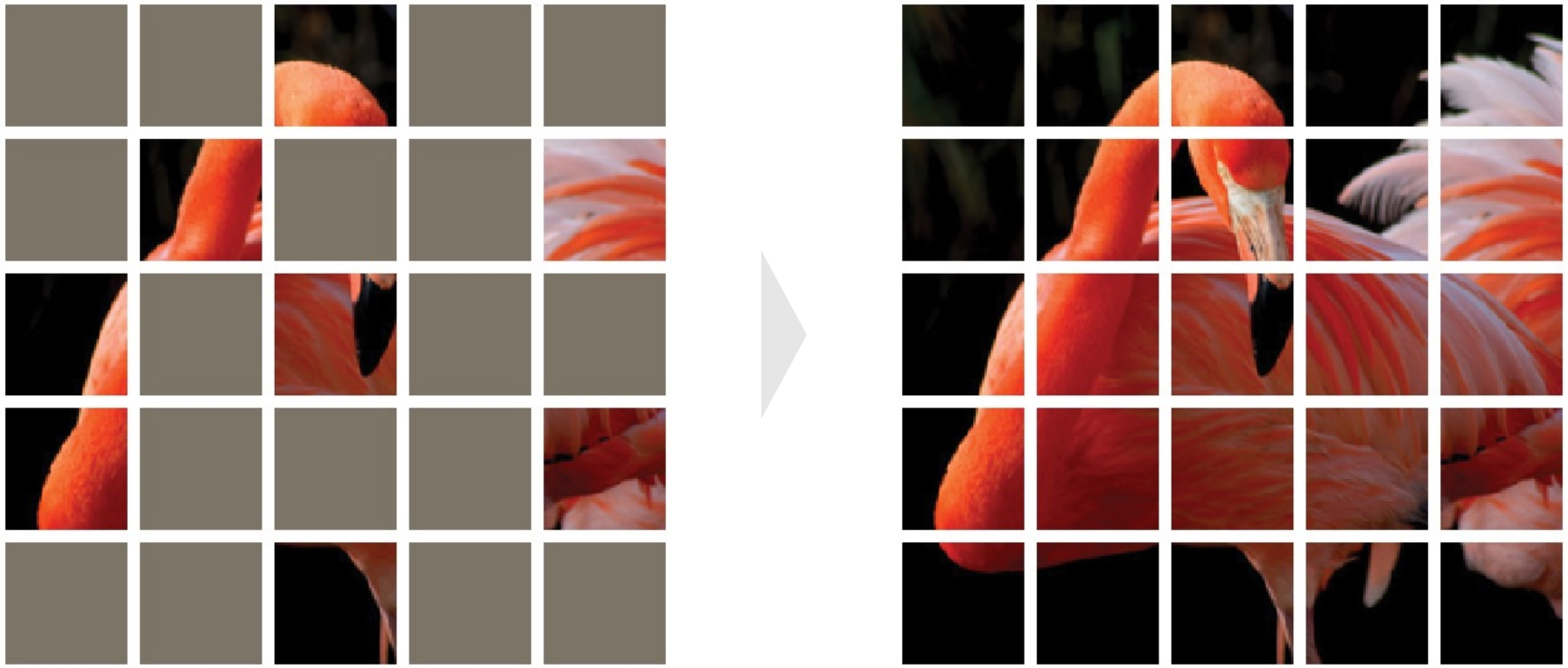
- Predict the masked regions using ConvNets

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

Predictive Learning: Computer Vision

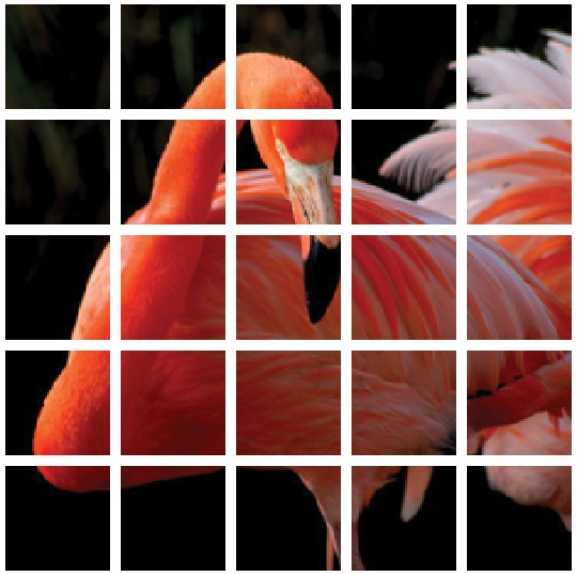
Masked image modeling (Masked Autoencoder)

- Predict the masked patches using Transformers



He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

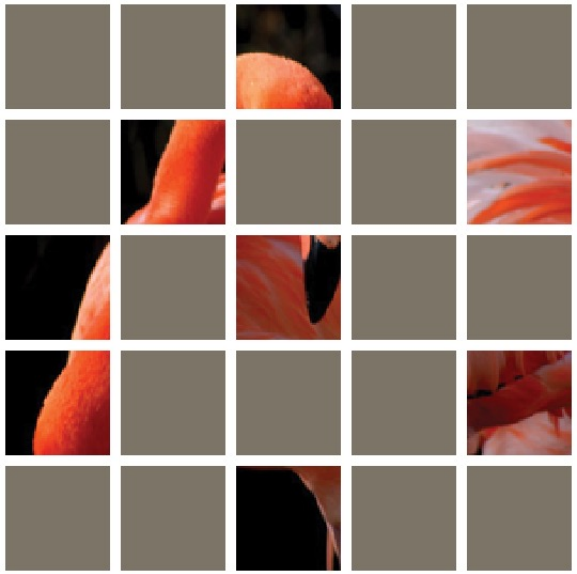
Masked Autoencoder (MAE)



patches as visual tokens
(Vision Transformer)

He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

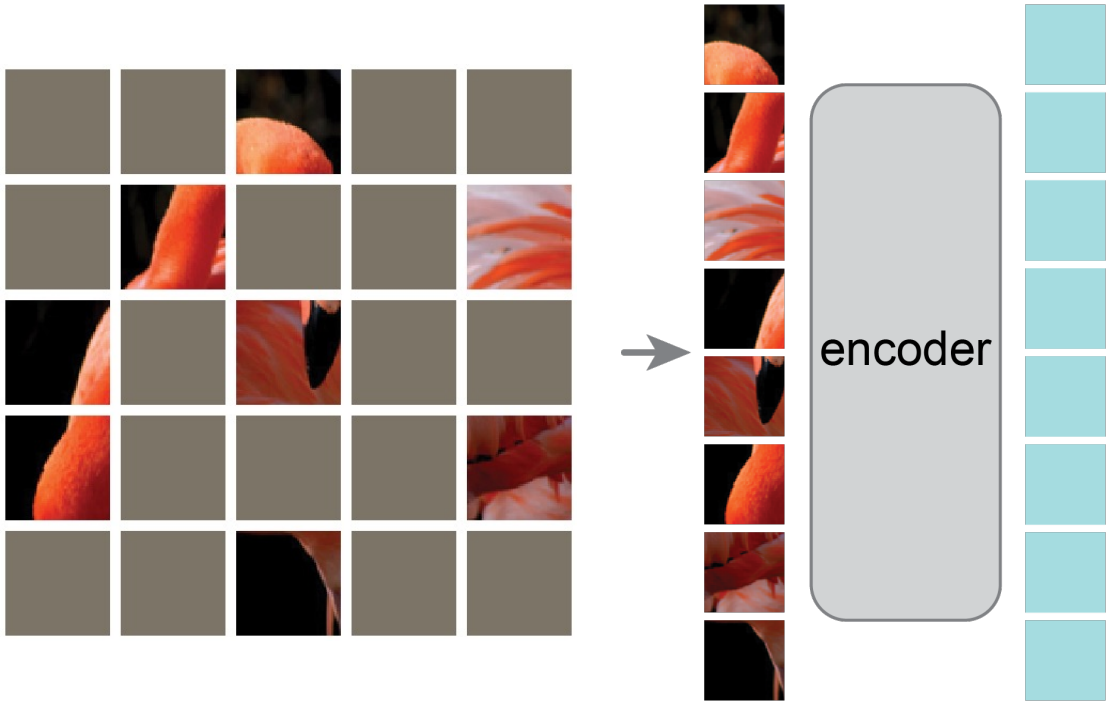
Masked Autoencoder (MAE)



random masking

He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

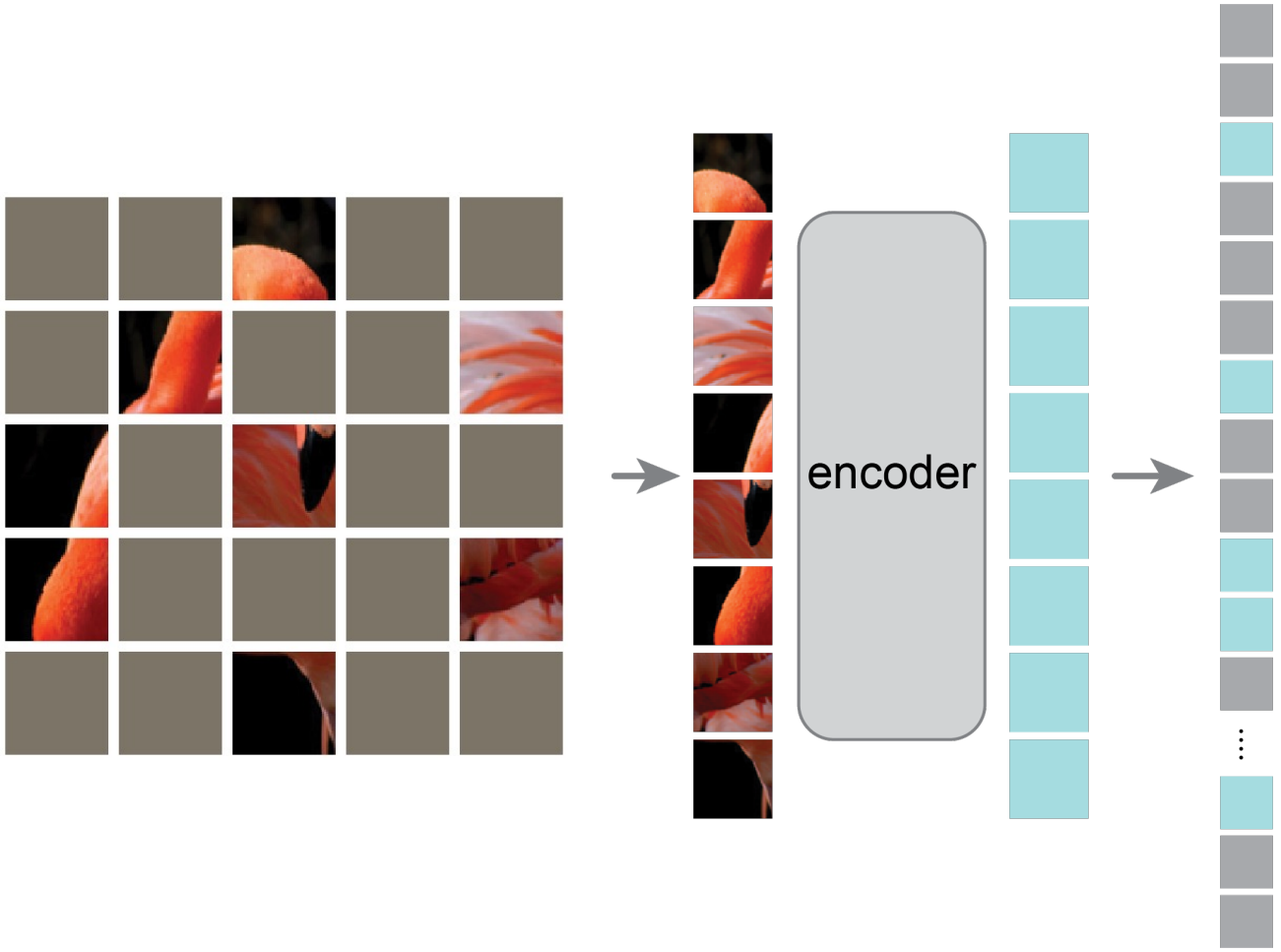
Masked Autoencoder (MAE)



encode visible patches
w/ Transformer

He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

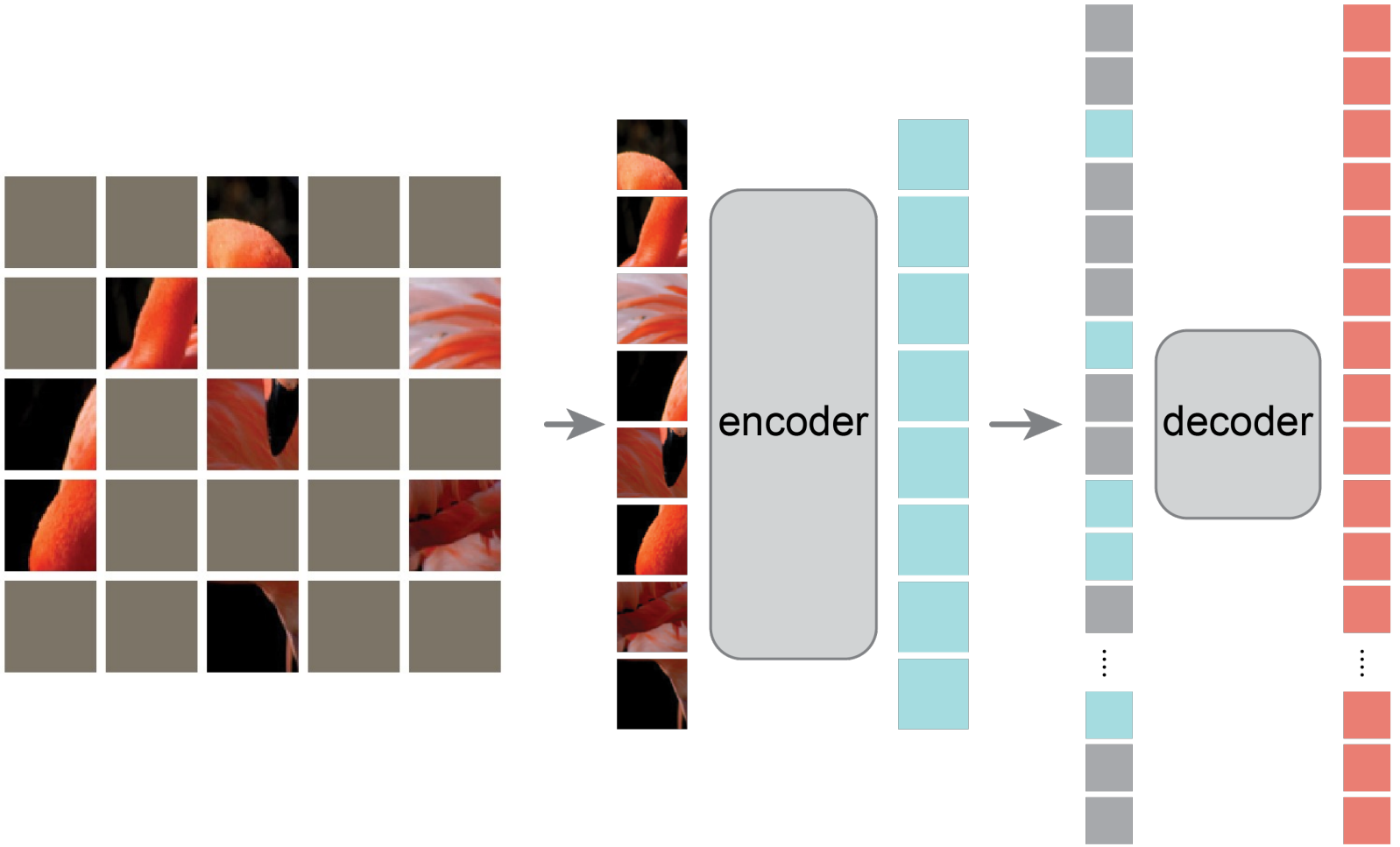
Masked Autoencoder (MAE)



expand with “mask tokens”
(specify where to predict)

He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

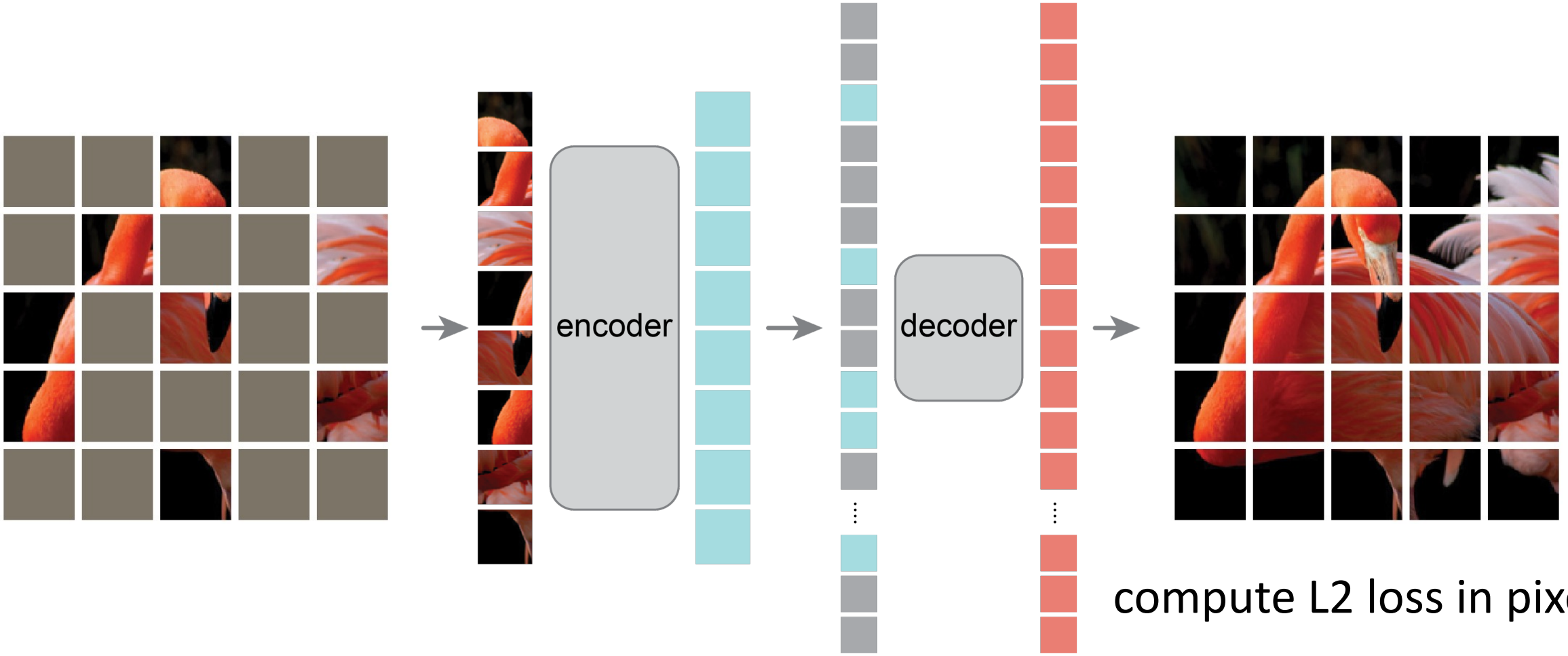
Masked Autoencoder (MAE)



predict the unknown

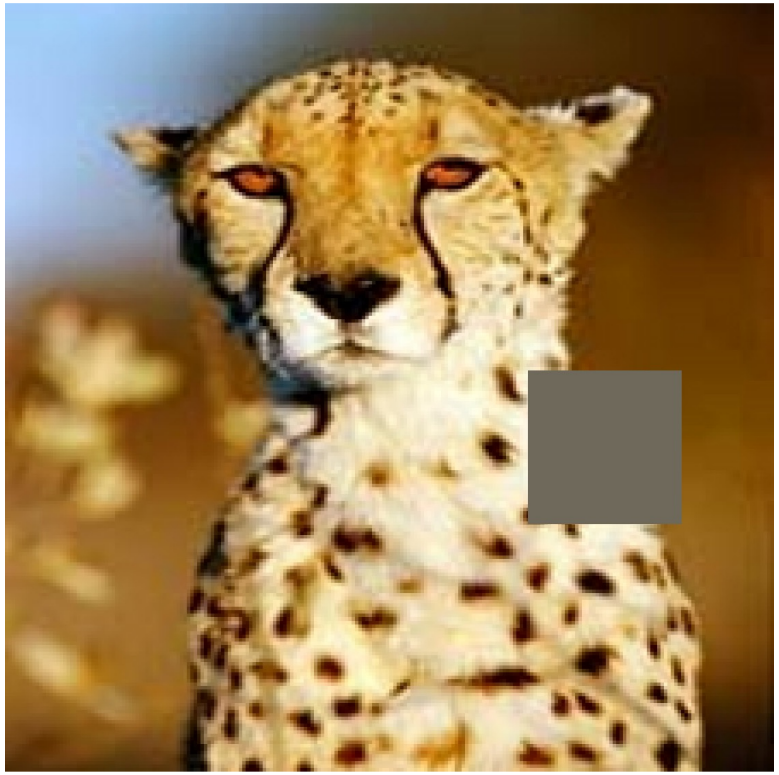
He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

Masked Autoencoder (MAE)

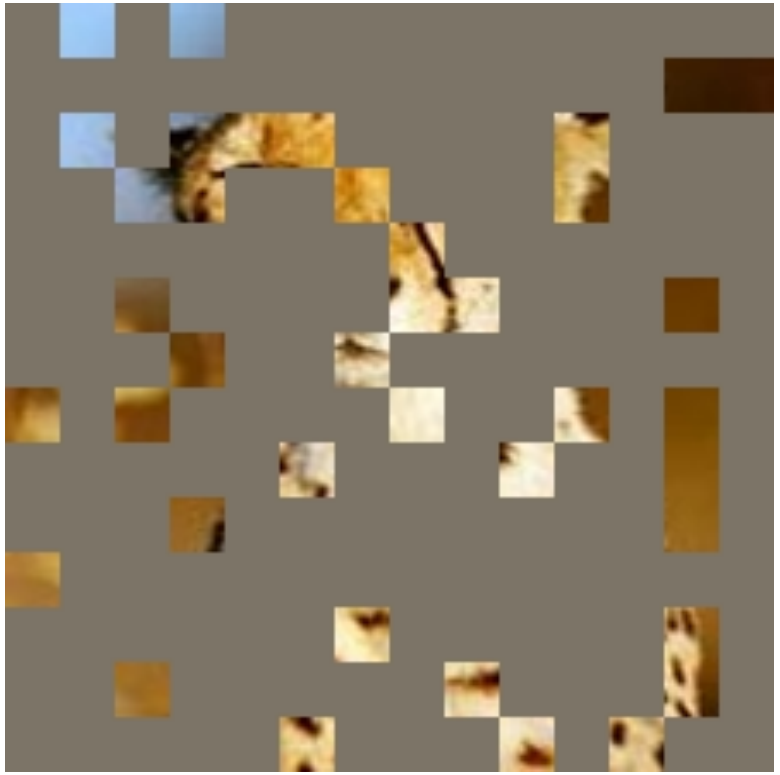


He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

How to learn good representations by predicting?



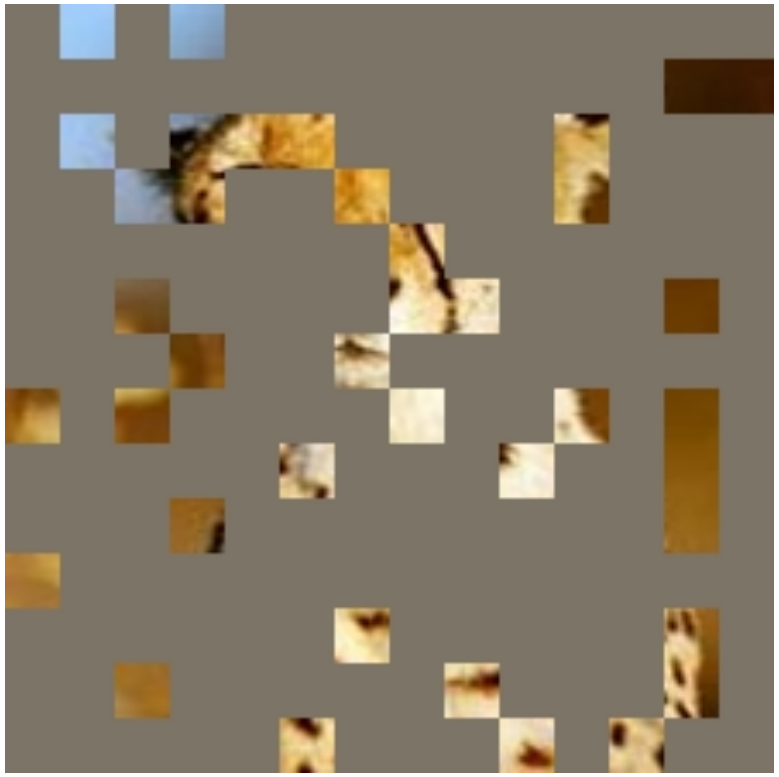
- predicting a small portion may not require high-level understanding



- predicting a large portion of unknown patches encourages to learn semantic features

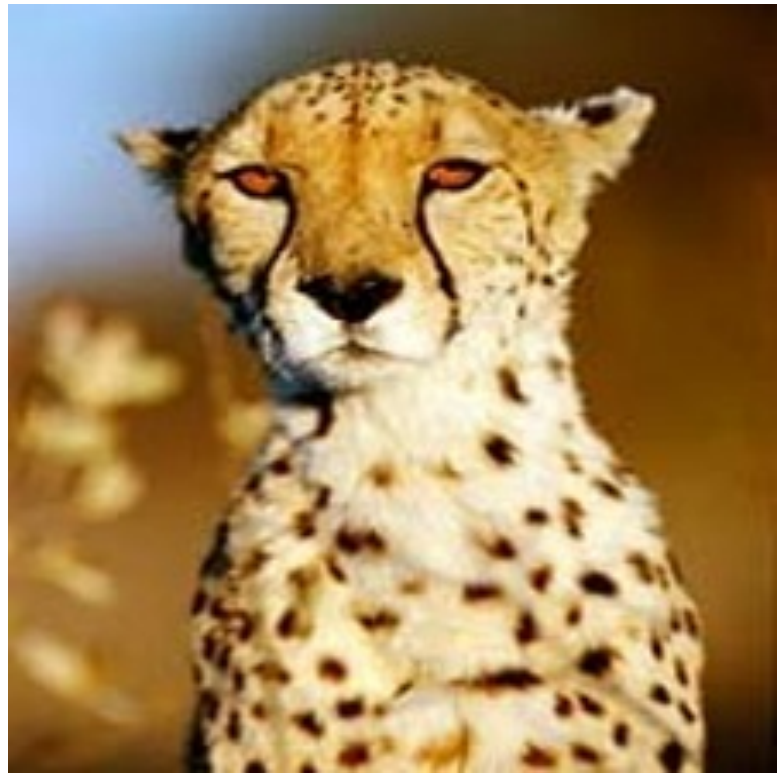
He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

How to learn good representations by predicting?



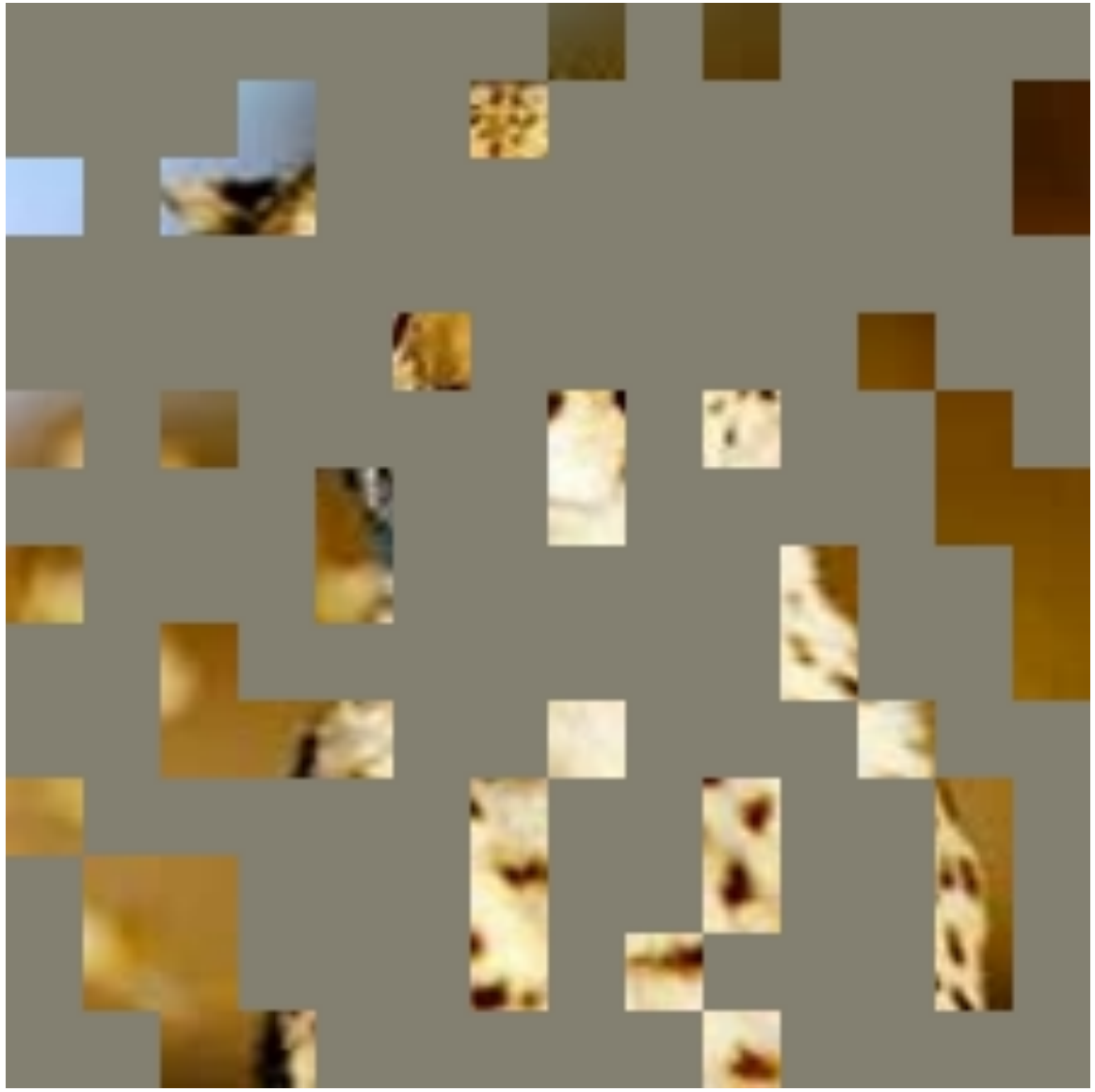
input

MAE prediction



original

He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022



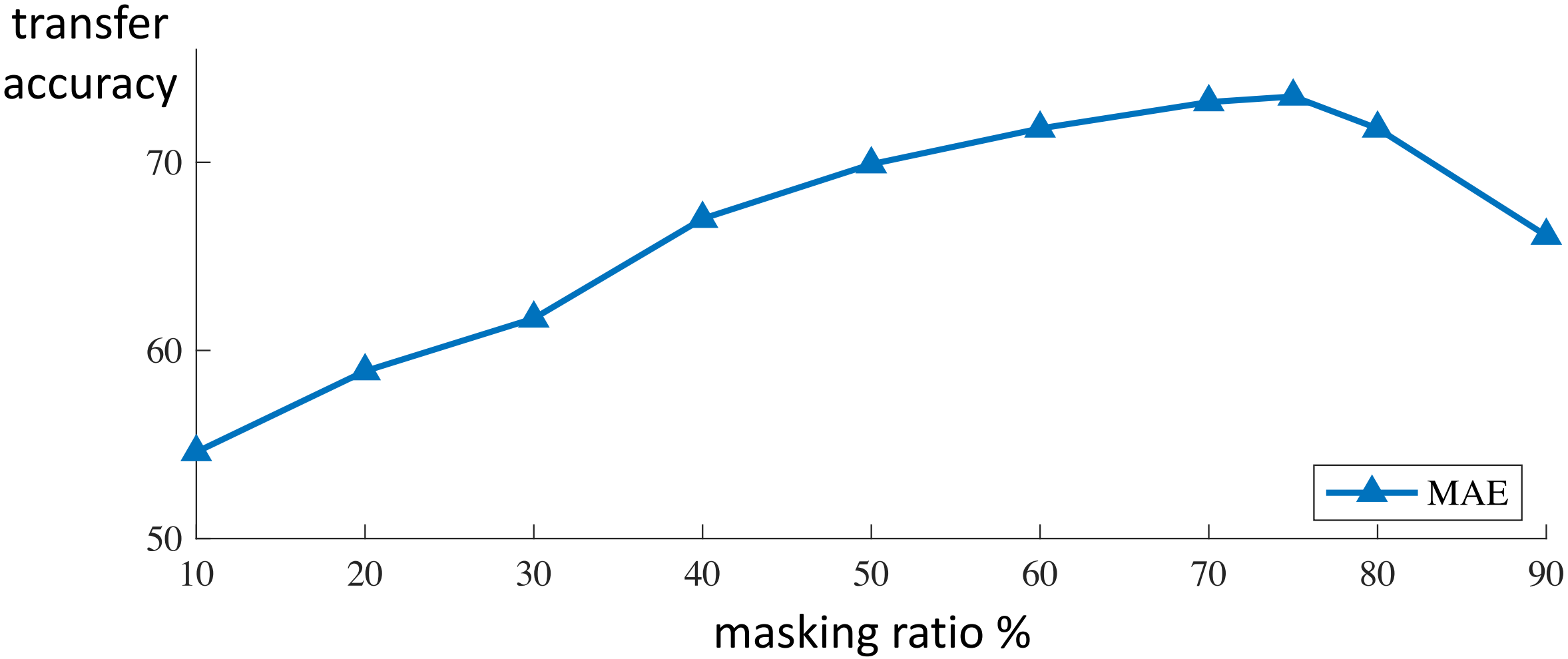
- **The learning process**: the network gradually makes sense of the semantic patterns

He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

- **The learning process**: the network gradually makes sense of the semantic patterns

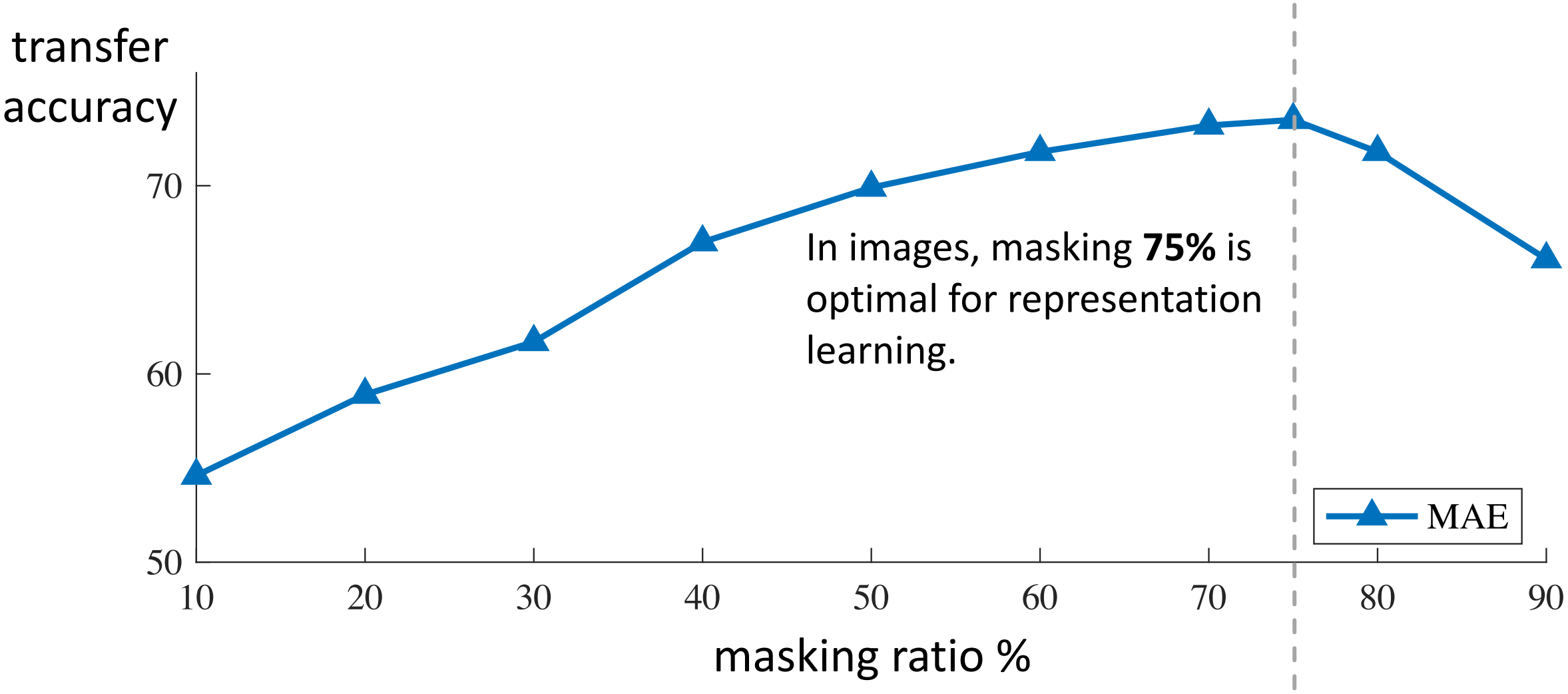
He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

How to learn good representations by predicting?



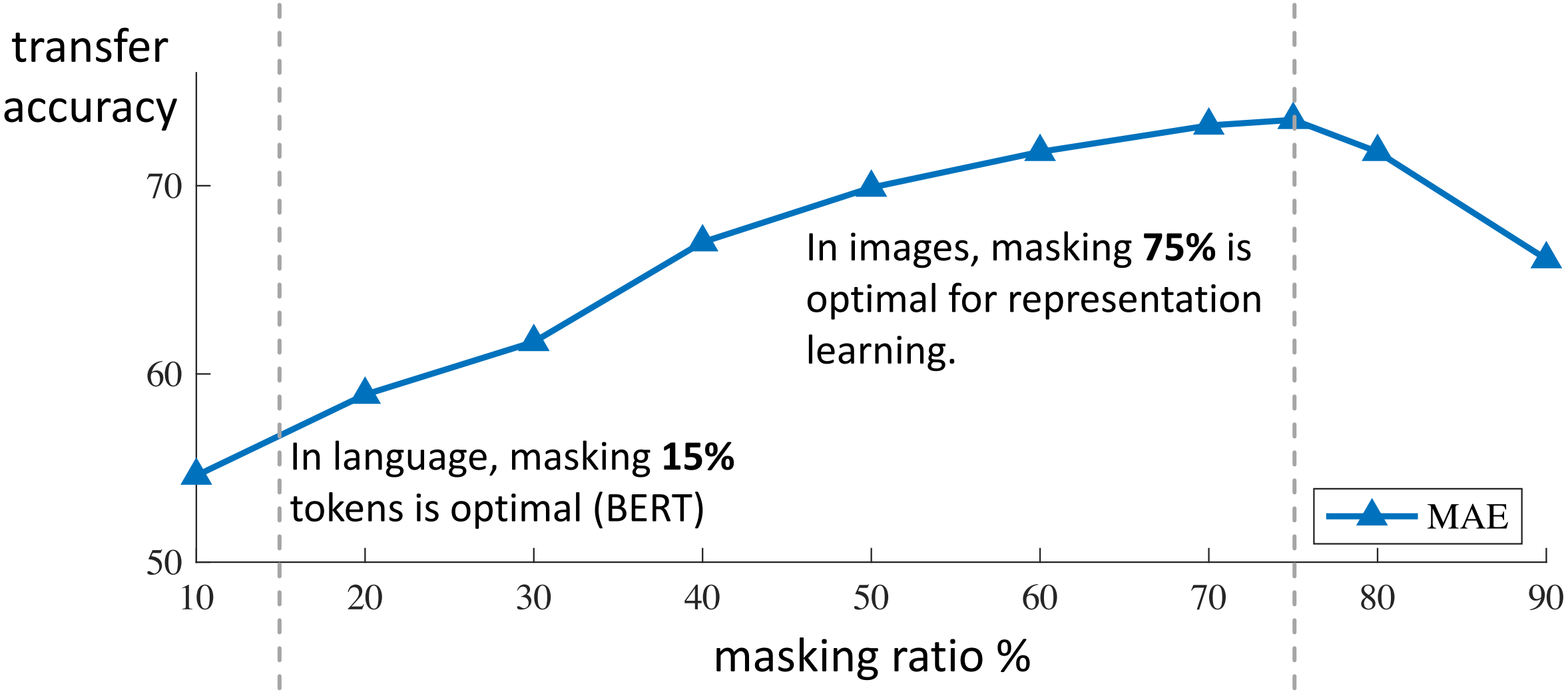
He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

How to learn good representations by predicting?



He, et al., "Masked Autoencoders Are Scalable Vision Learners", CVPR 2022

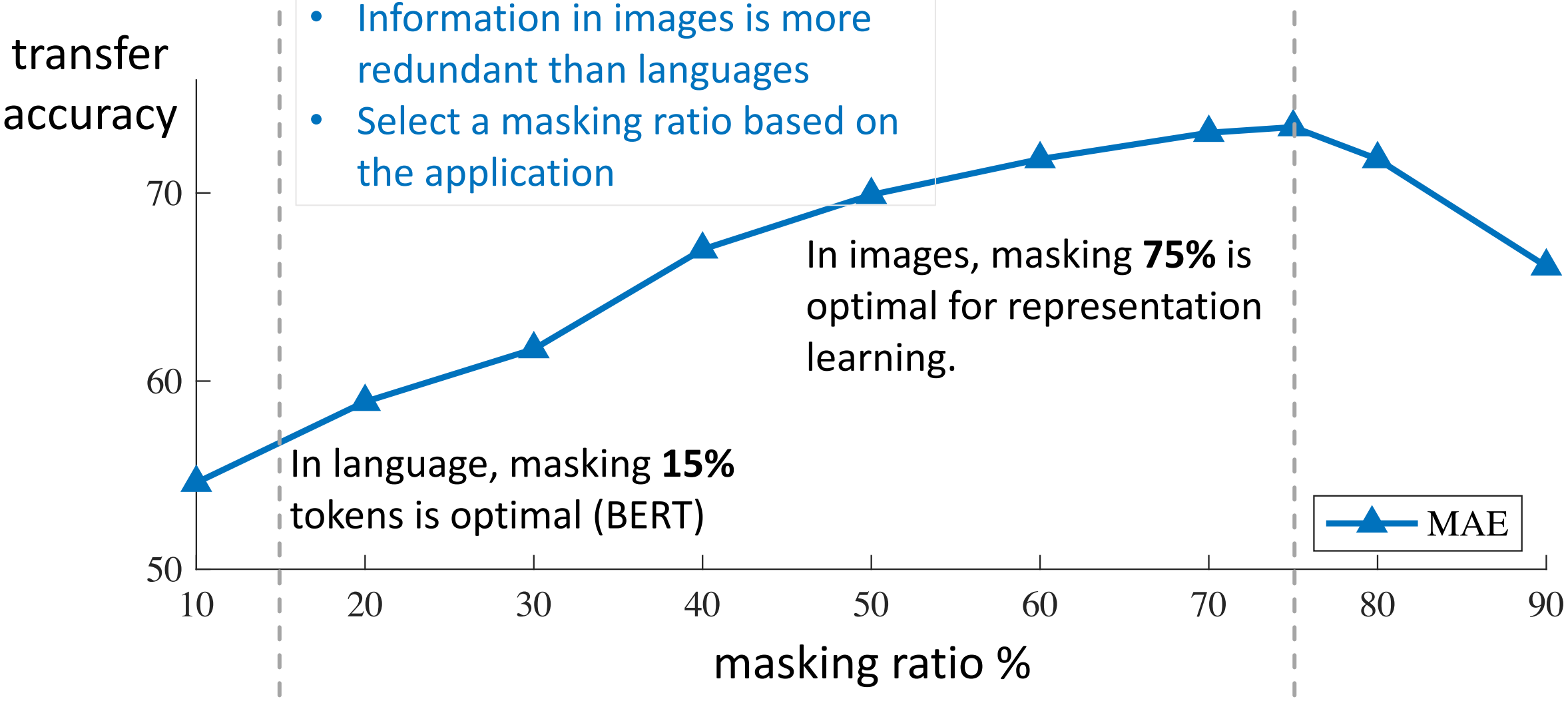
How to learn good representations by predicting?



He, et al., "Masked Autoencoders Are Scalable Vision Learners", CVPR 2022

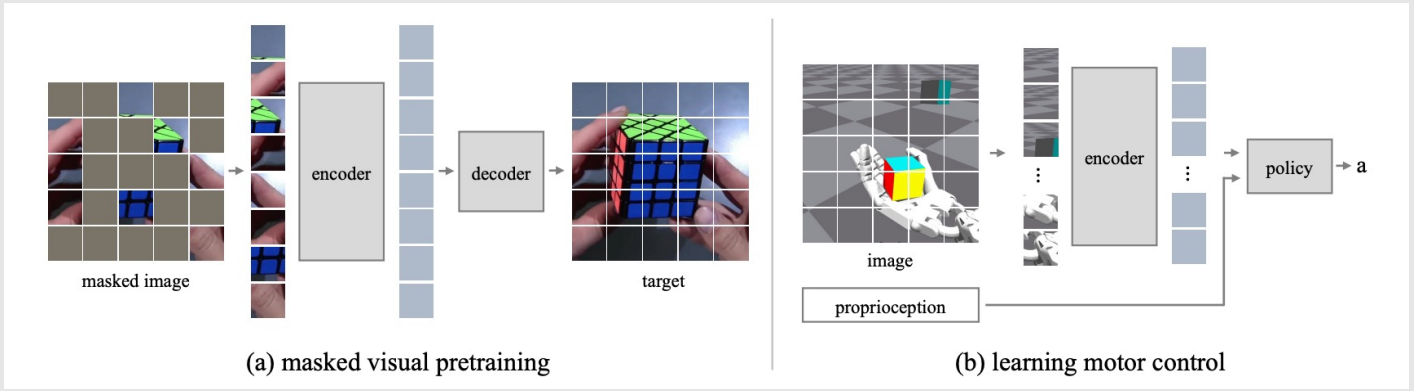
How to learn good representations by predicting?

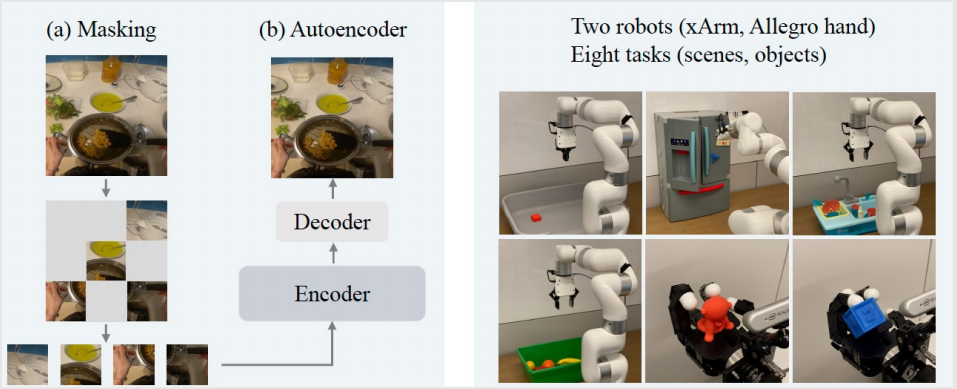
- Information in images is more redundant than languages
- Select a masking ratio based on the application

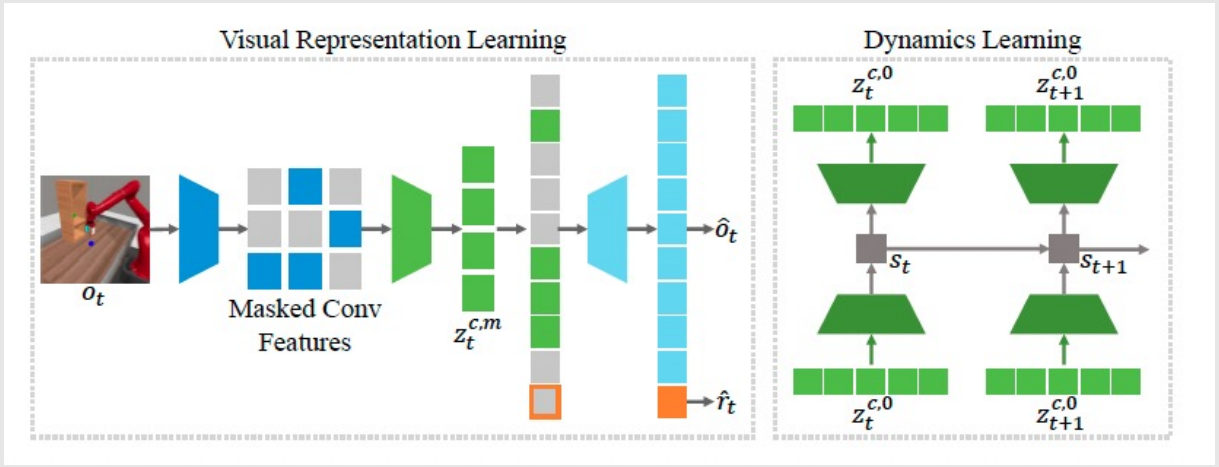


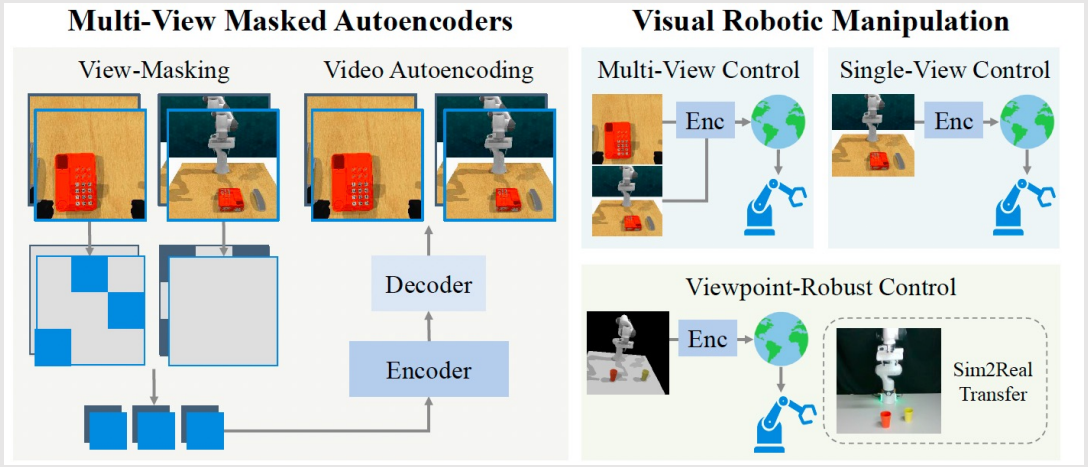
He, et al., "Masked Autoencoders Are Scalable Vision Learners", CVPR 2022

Masked Autoencoder: Extensions





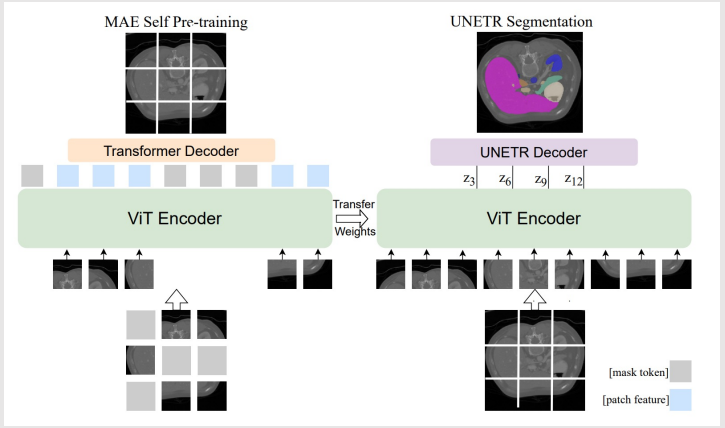


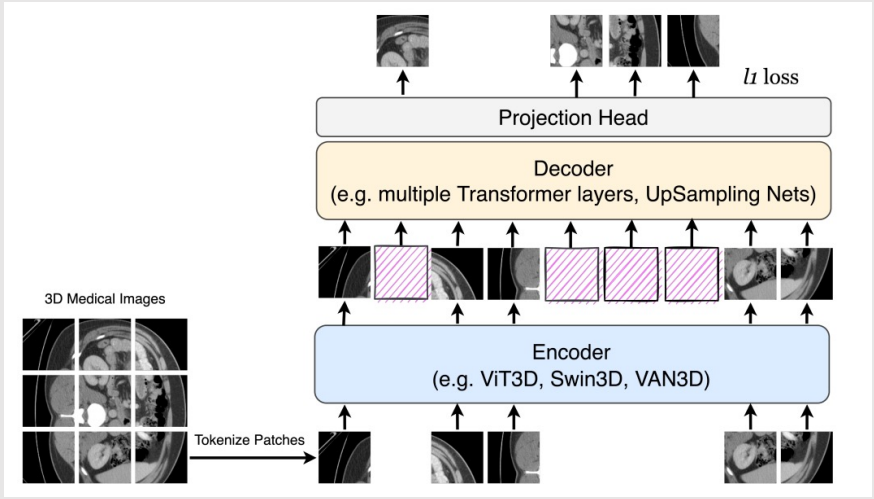


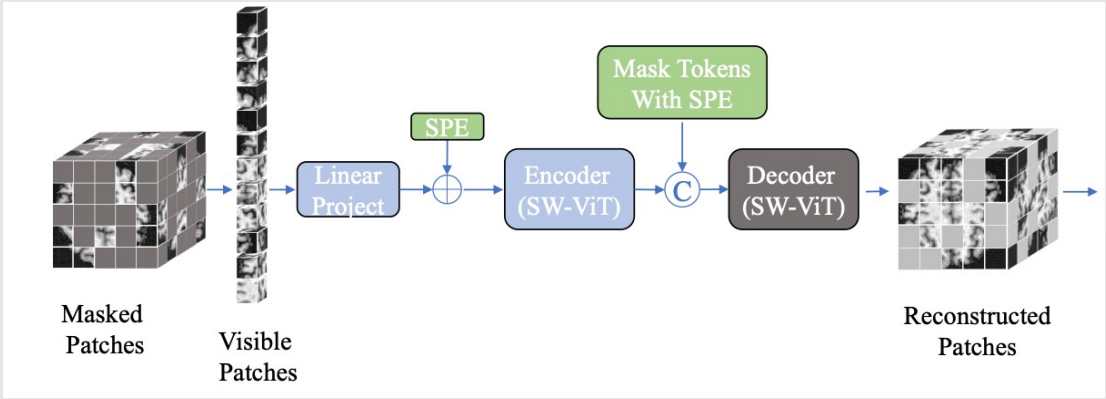
- **Robotics**

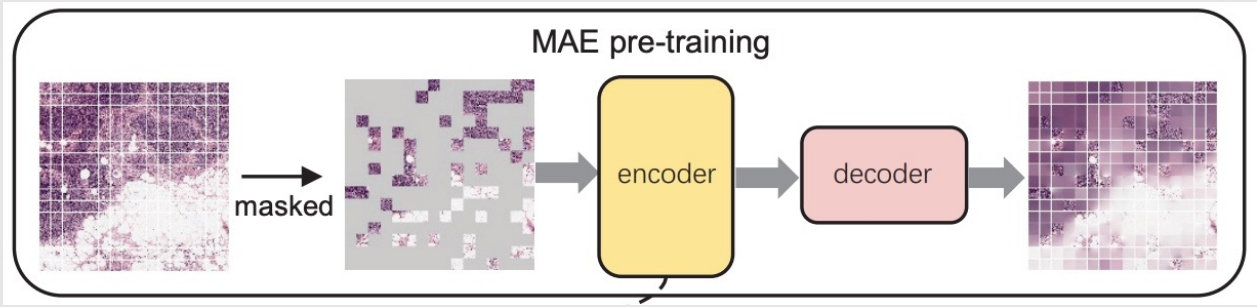
Xiao+, ’22; Radosavovic+, ’22; Seo+, ’22; Seo+, ’23

Masked Autoencoder: Extensions





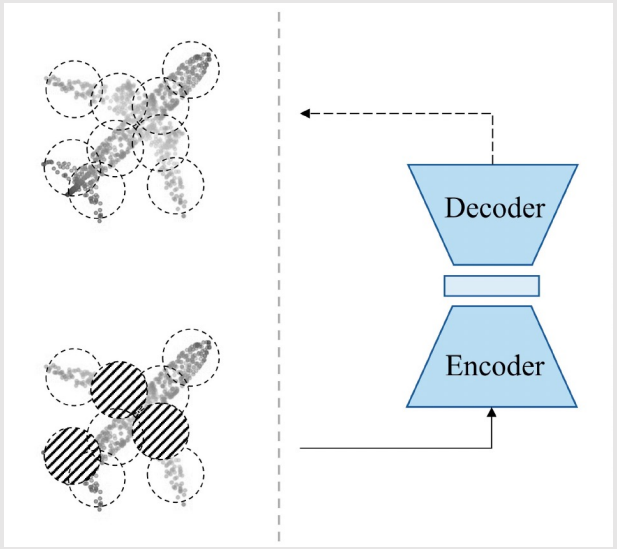


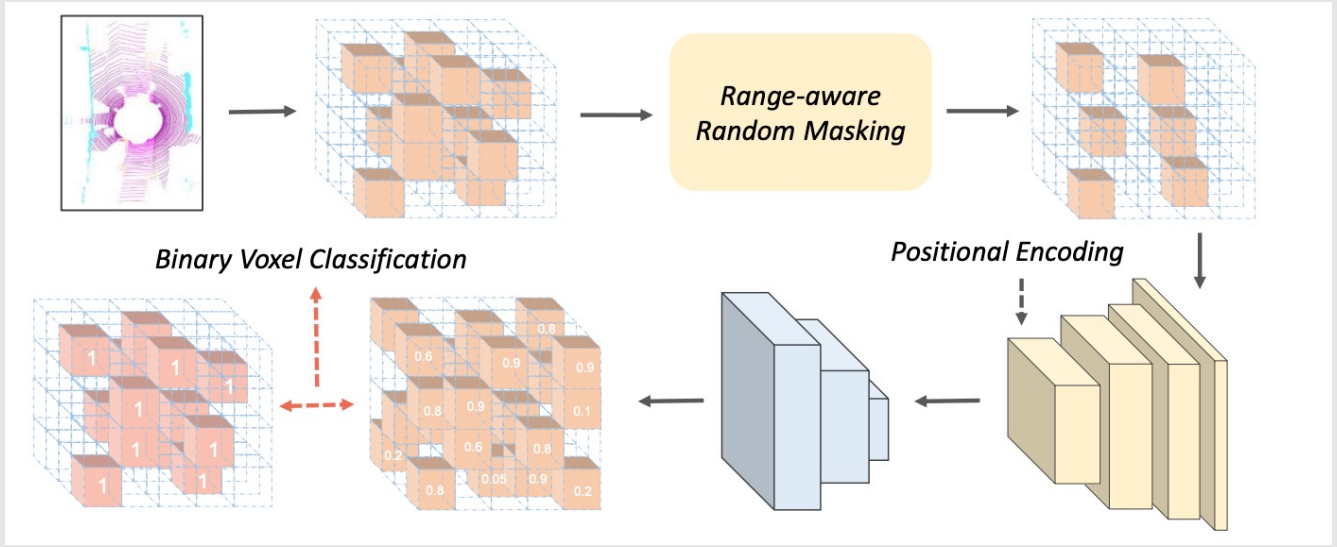


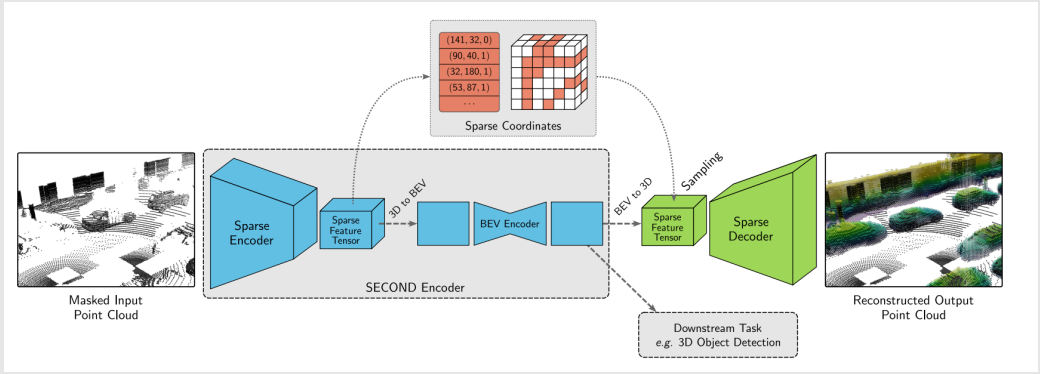
- **Medical Imaging**

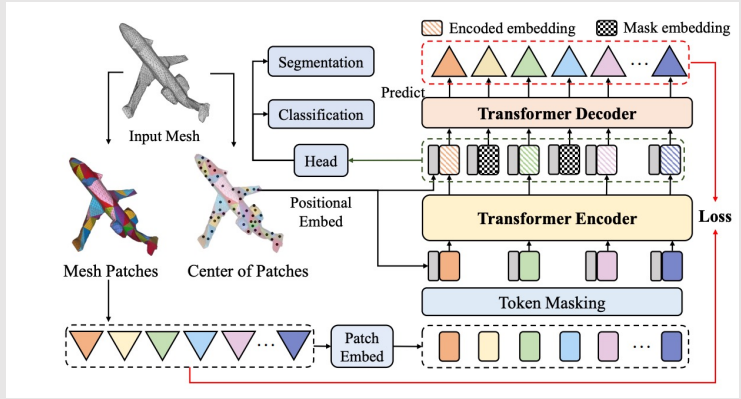
Zhou+,'22; Chen+,'22; Huang+,'22; An+,'22

Masked Autoencoder: Extensions





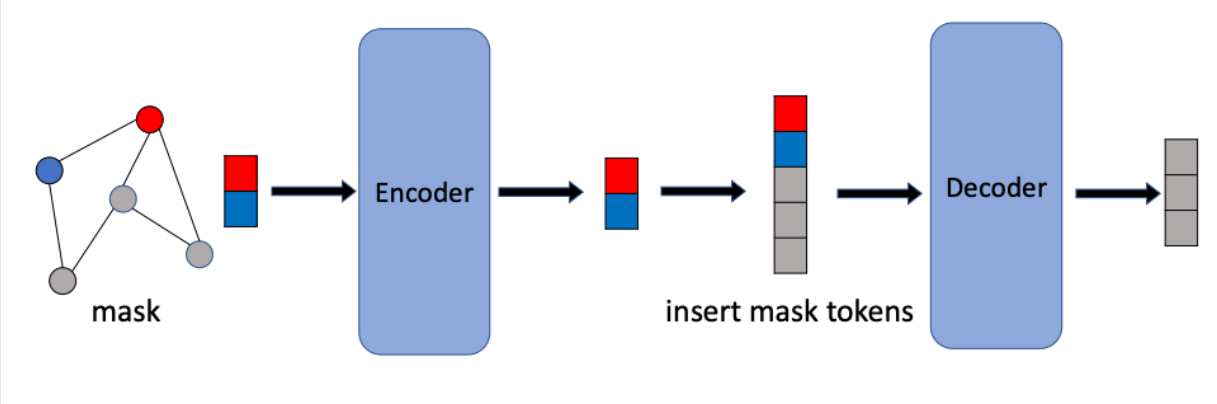


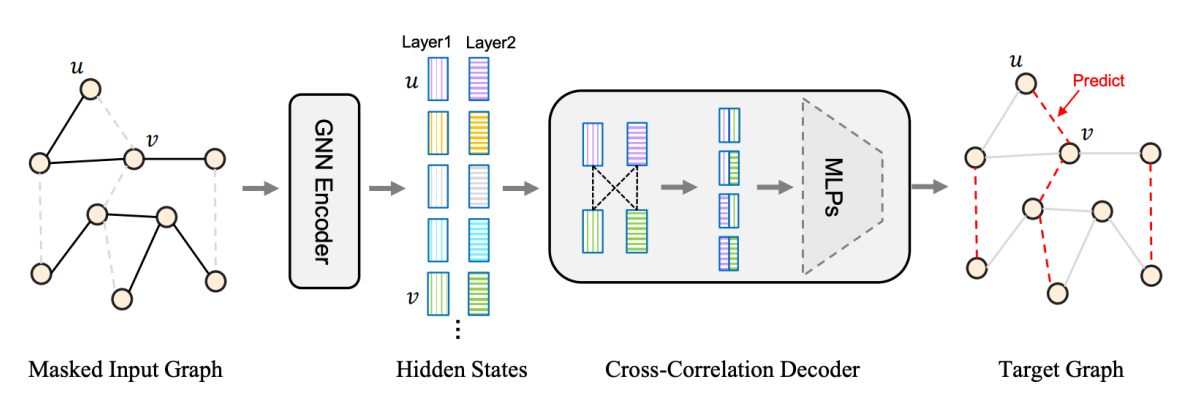


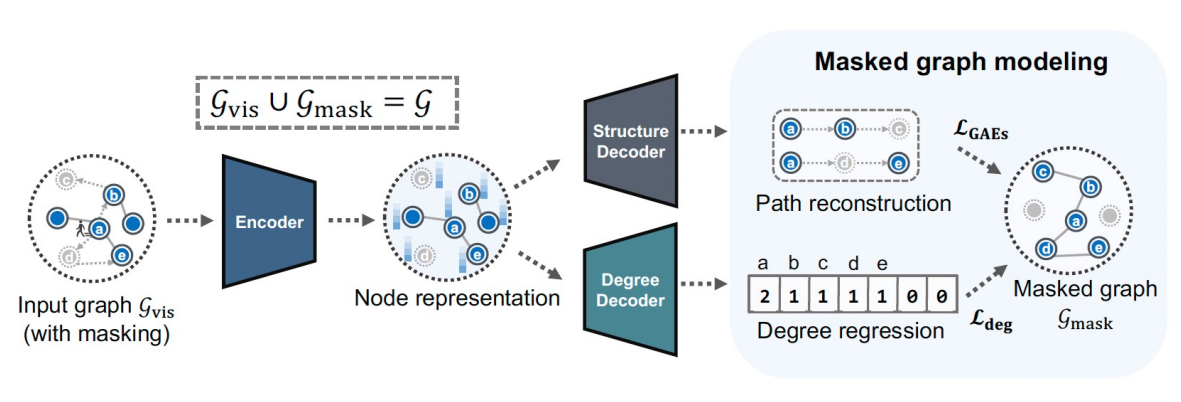
- **3D Geometry**

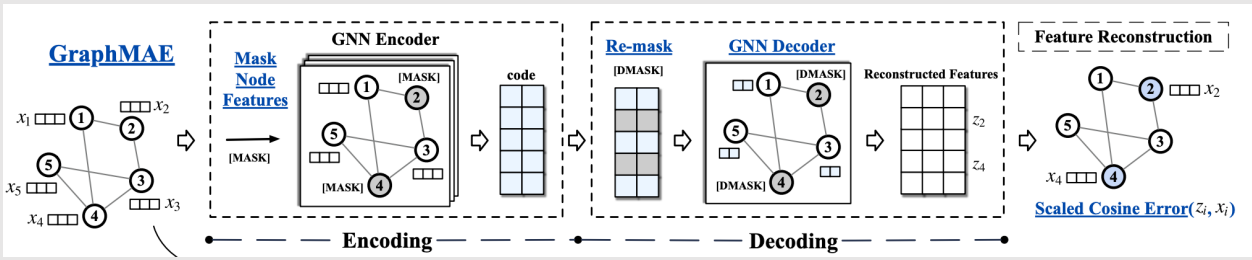
Pang+, ’22; Liang+, ’22; Min+, ’22; Krispel+, ’22

Masked Autoencoder: Extensions





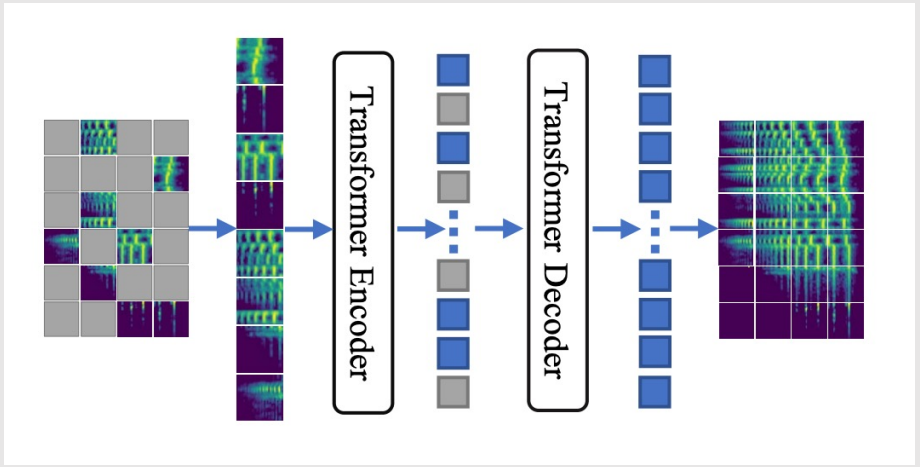


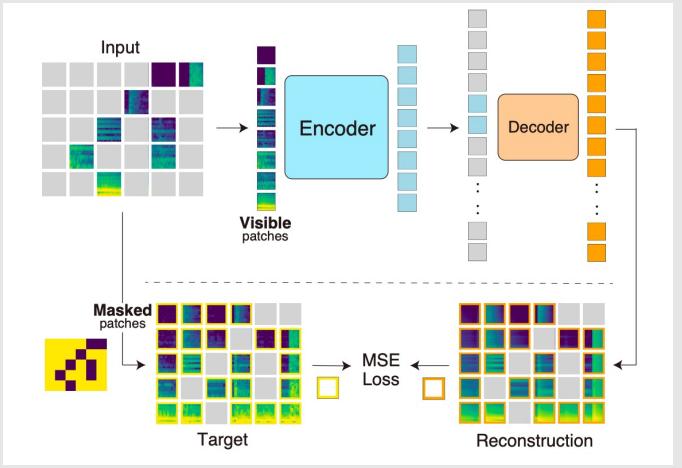


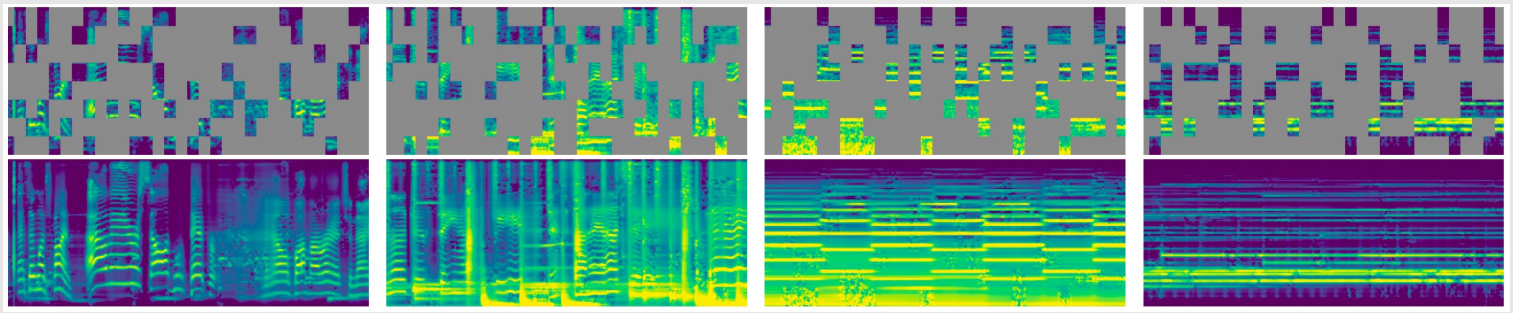
- **Graphs**

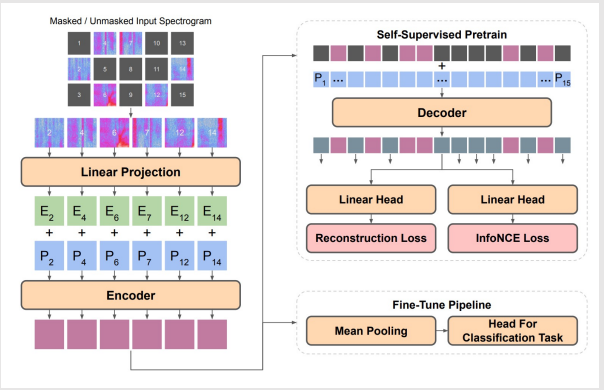
Tan+, '22; Zhang+, '22; Hou+, '22; Li+, '22

Masked Autoencoder: Extensions

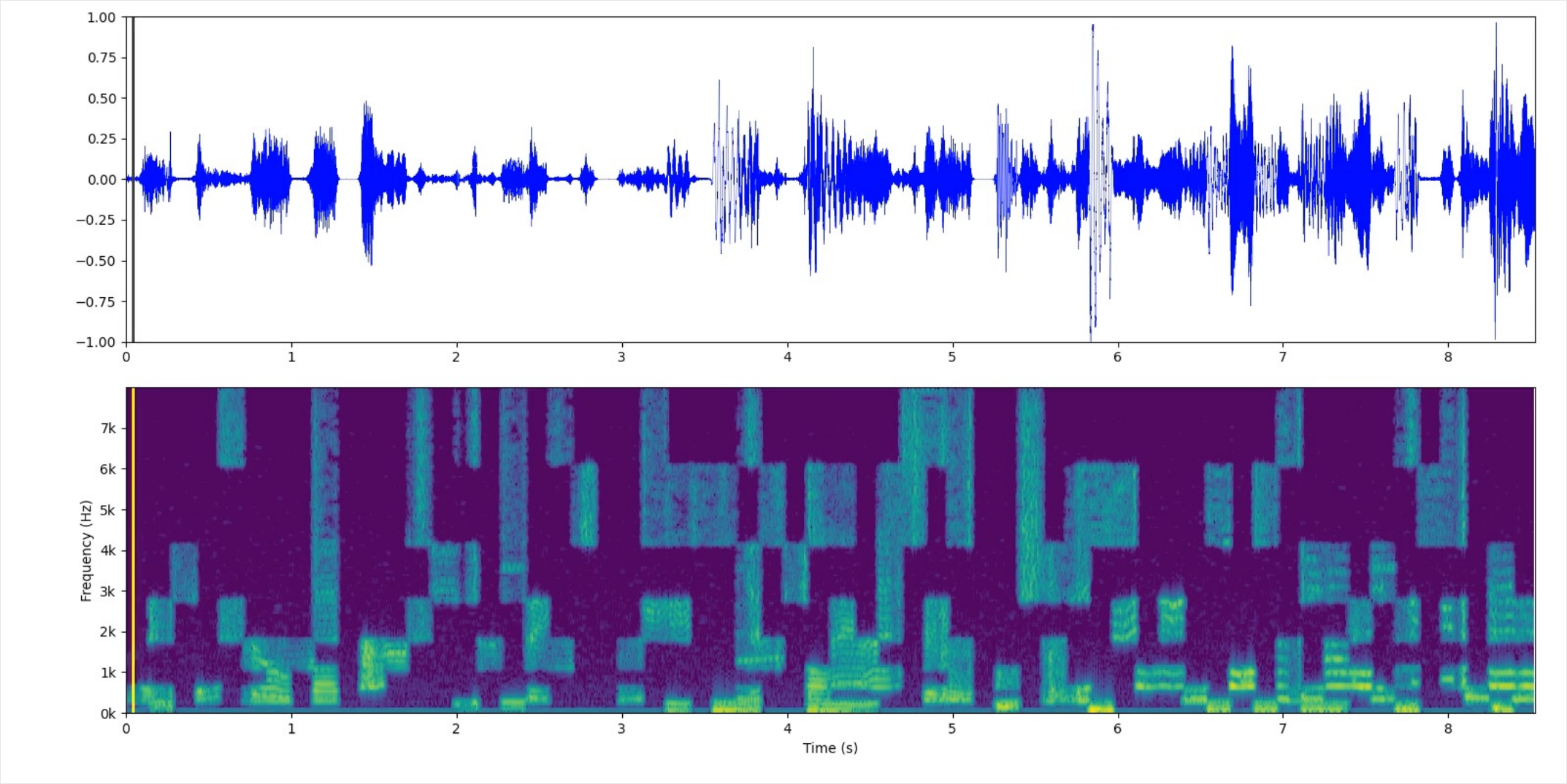








- **Audio**



Audio MAE

Huang, et al., *Masked Autoencoders that Listen*. NeurIPS ’22

Summary

Unsupervised learning - classical (but very useful) methods:

- K-means clustering
- Principal Component Analysis (PCA)
- Autoencoder

Self-supervised learning - modern advances:

- Contrastive Learning
- Predictive Learning

Lecture 12: Representation Learning

- Why should we care about Representation Learning?
- Transfer learning
- Unsupervised learning
- Self-supervised learning