Lecture 8

Representation Learning:
Intro &
Reconstruction-based Methods

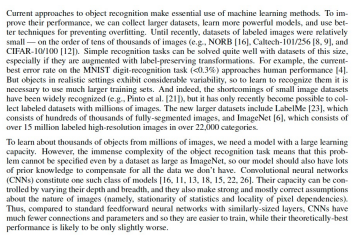
Speaker: Kaiming He

Overview

- Introduction to Representation Learning
- Unsupervised Learning and Self-supervised Learning
- Reconstruction-based Methods

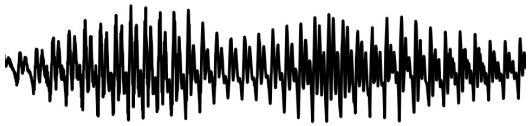
Deep Learning is Representation Learning

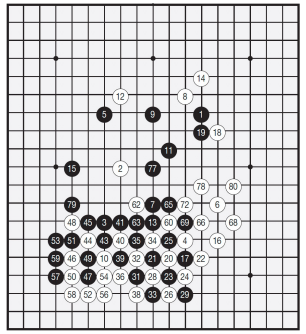
- Represent raw data to solve complex problems
  - compression, abstraction, conceptualization
- Raw data in different forms
  - pixels, words, waves, states, molecules, DNA, …

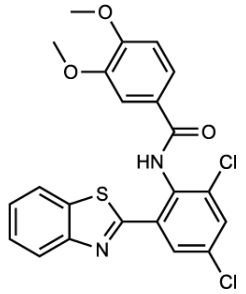


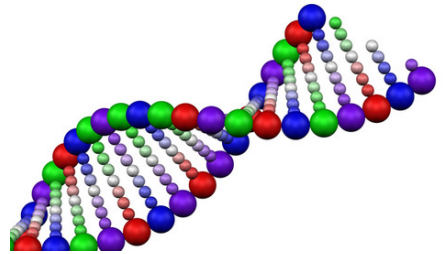
Current approaches to object recognition make essential use of machine learning methods. To improve their performance, we can collect larger datasets, learn more powerful models, and use better techniques for preventing overfitting. Until recently, datasets of labeled images were relatively small — on 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.





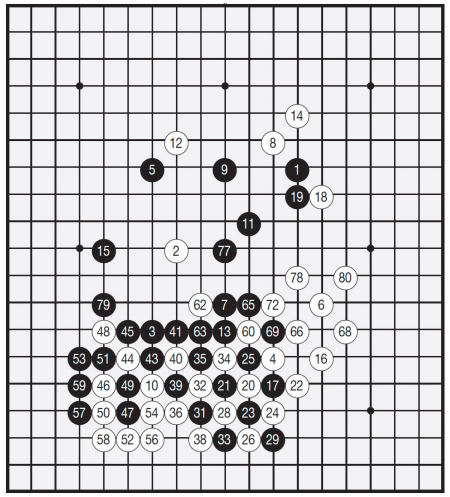




Defines a conference name: ICLR

(International Conference on Learning Representations)

Deep Learning is Representation Learning

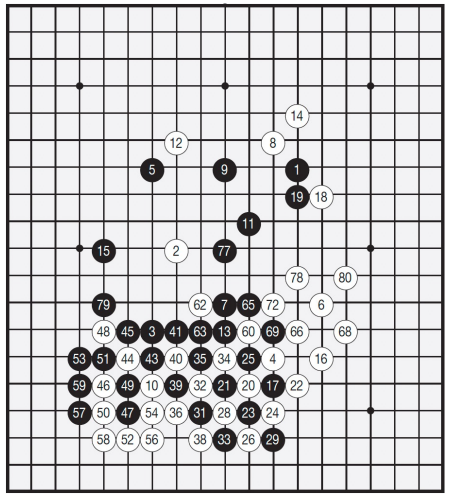


3^{361} states?

Game of Go

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

Deep Learning is Representation Learning



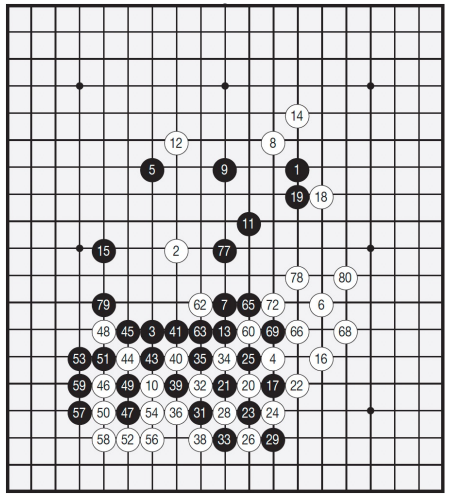
3^{361} states?

Bad representation

$256^{3x500x500}$?

- Image space has exponentially more states than Go.

Deep Learning is Representation Learning

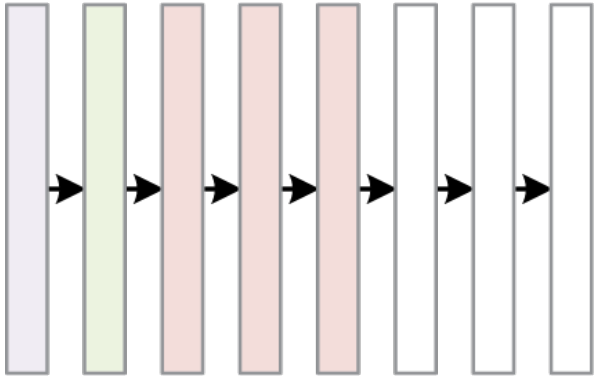


3^{361} states?

$256^{3x500x500}$?

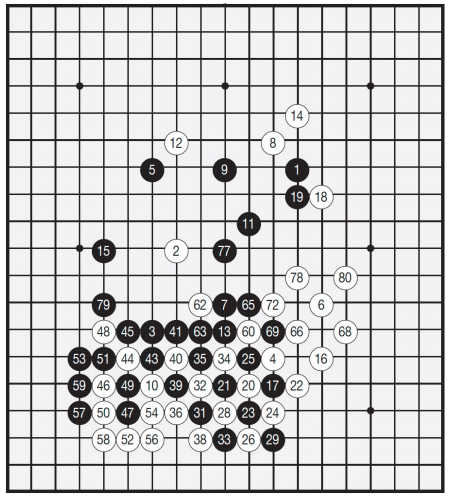
- Image recognition is solved in representation space.

Good representation

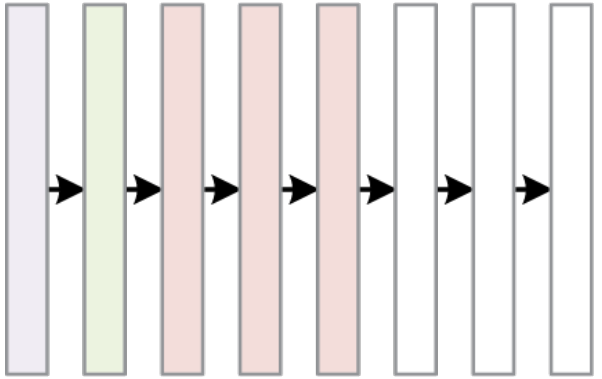


Deep Learning is Representation Learning

- Go playing can be solved in representation space.

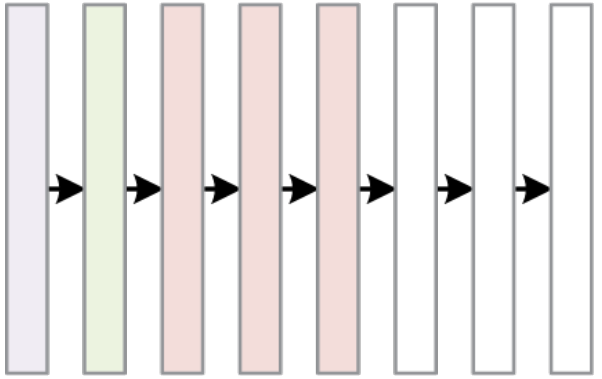


3^{361} states?



Good representation

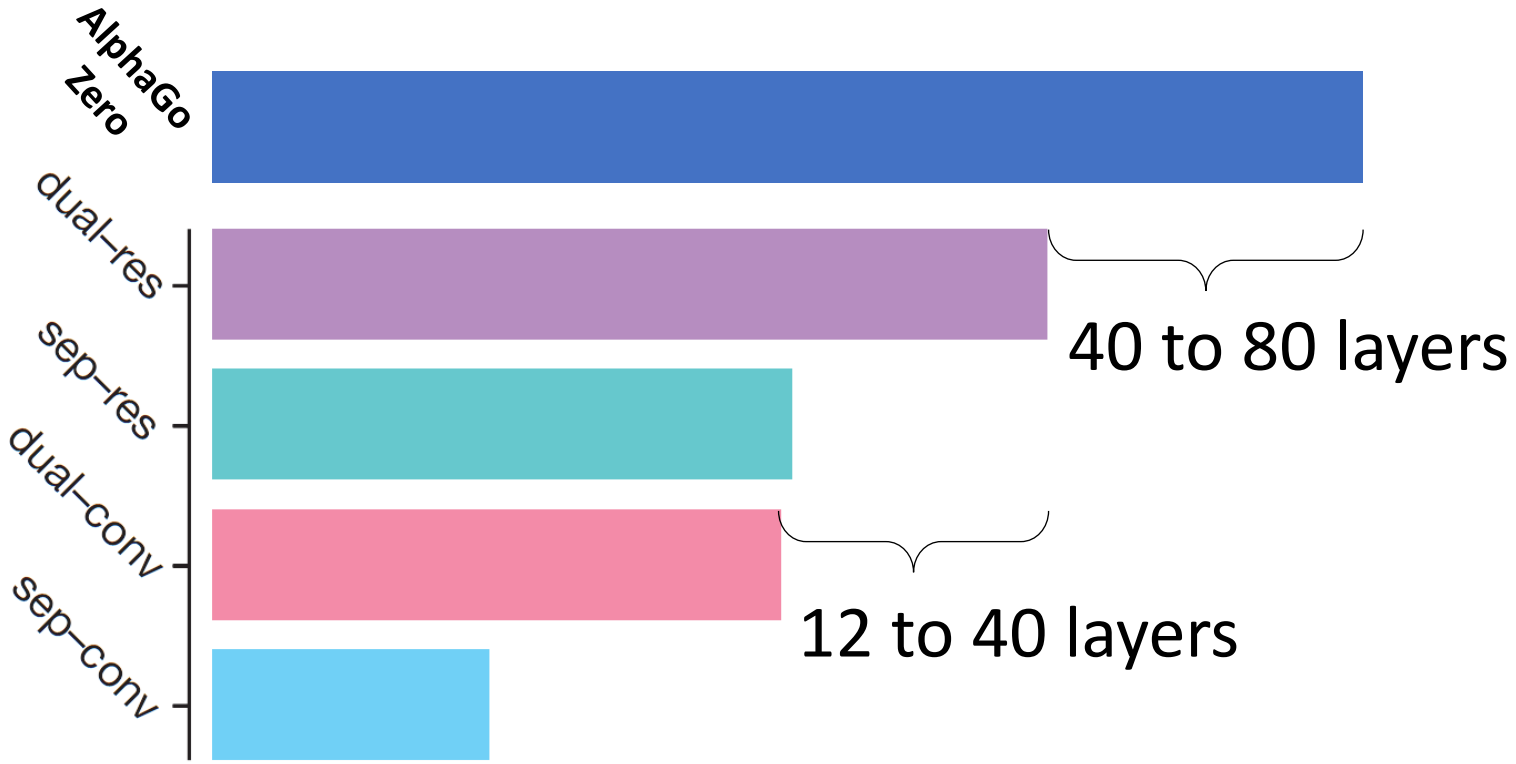
$256^{3x500x500}$?



Deep Learning is Representation Learning

AlphaGo: better representation leads to higher rating

- outperform best human players
- without human knowledge

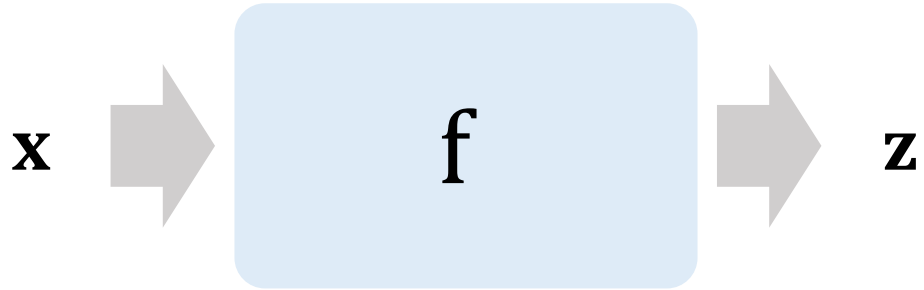


Silver, et al., Mastering the game of Go without human knowledge, Nature 2017

What is a representation?

- A representation of a data domain $\mathcal{X}$ is a function f: $\mathcal{X} \rightarrow R^d$ that assigns a feature vector to each input in that domain.

- A representation of a datapoint $\mathbf{x}$ is a vector $\mathbf{z} \in R^d$ with $\mathbf{z} = \mathrm{f}(\mathbf{x})$.



can be extended to:

- high-dim arrays: $d = d_1 \times d_2 \times \ldots$
- other data structures: trees, list of arrays (pyramids), ...

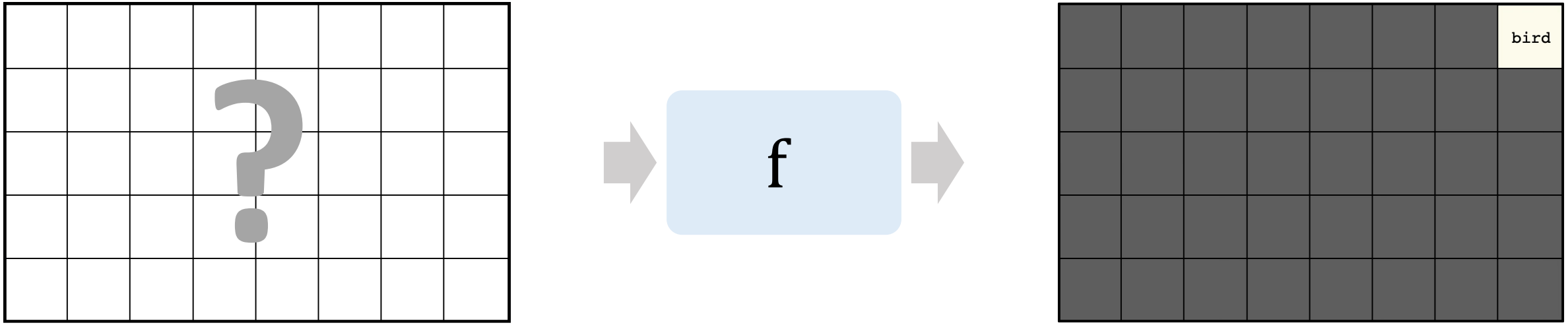
What does a representation represent?

f

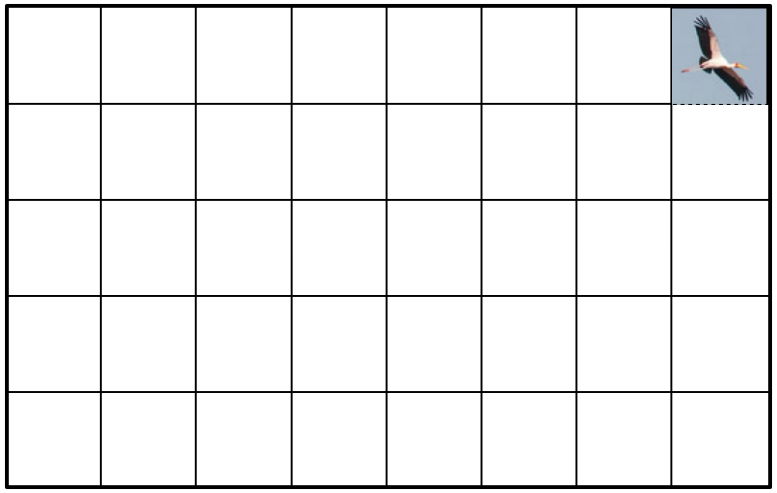
sky	sky	sky	sky	sky	sky	sky	bird
sky	sky	sky	sky	sky	sky	sky	sky
sky	sky	sky	sky	sky	sky	sky	sky
bird	bird	bird	bird	sky	bird	sky	sky
sky	sky	sky	bird	bird	sky	sky	sky

bird = [0,0,1,0,0,...,0]
sky = [0,0,0,0,1,...,0]

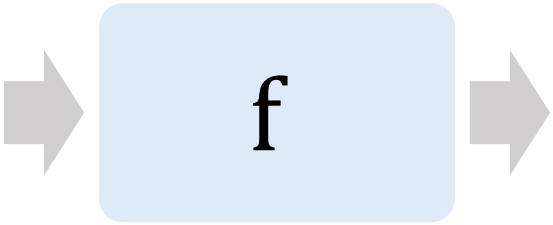
What does a representation represent?

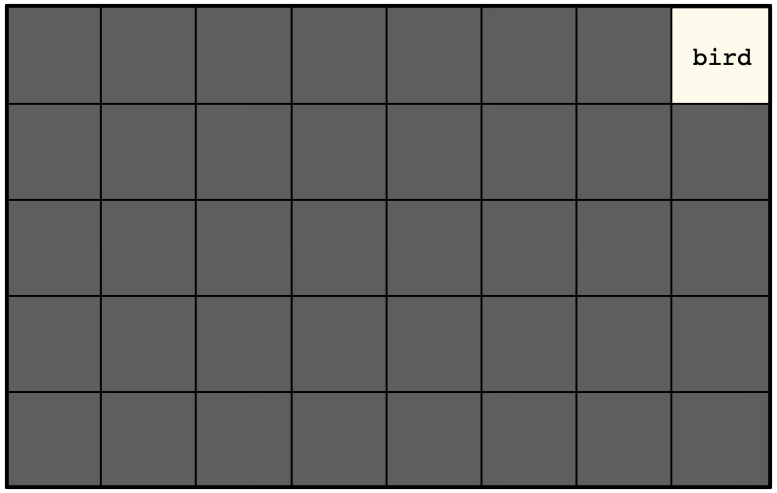


What does a representation represent?

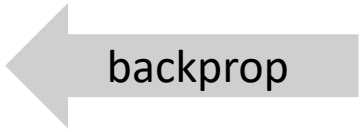


$\mathbf{x}$





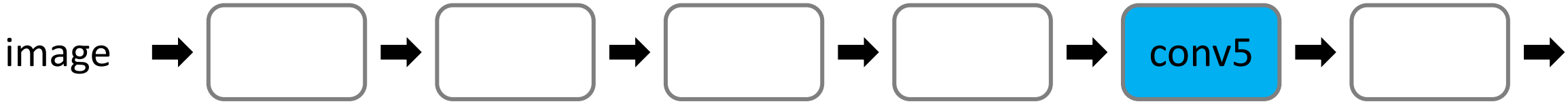
$\mathbf{z}$

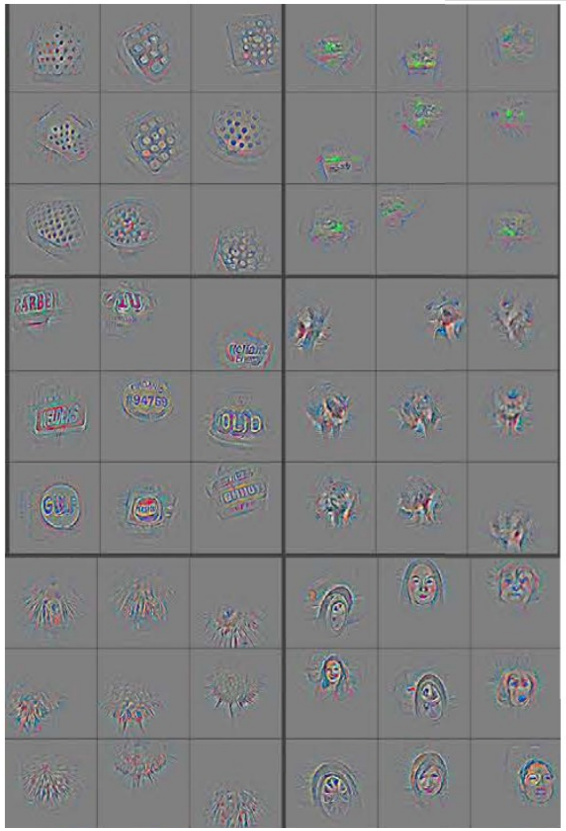


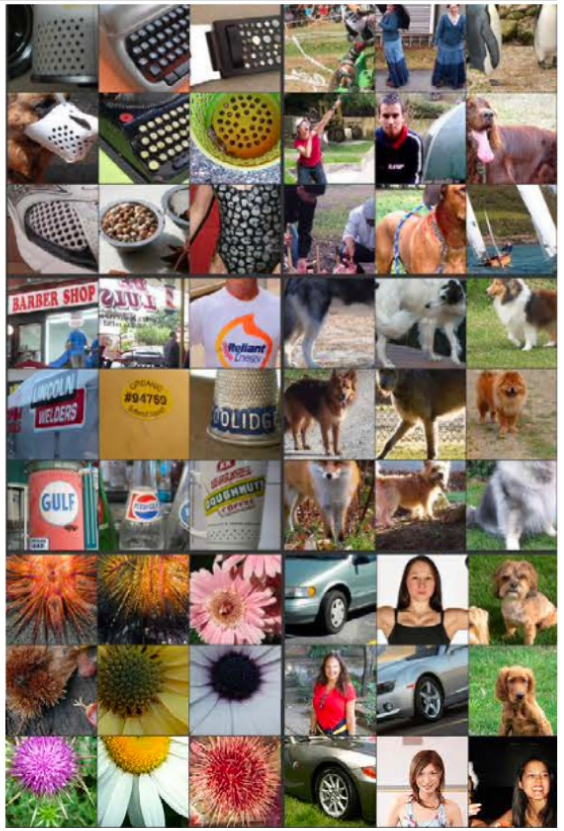
find $\mathbf{x}$

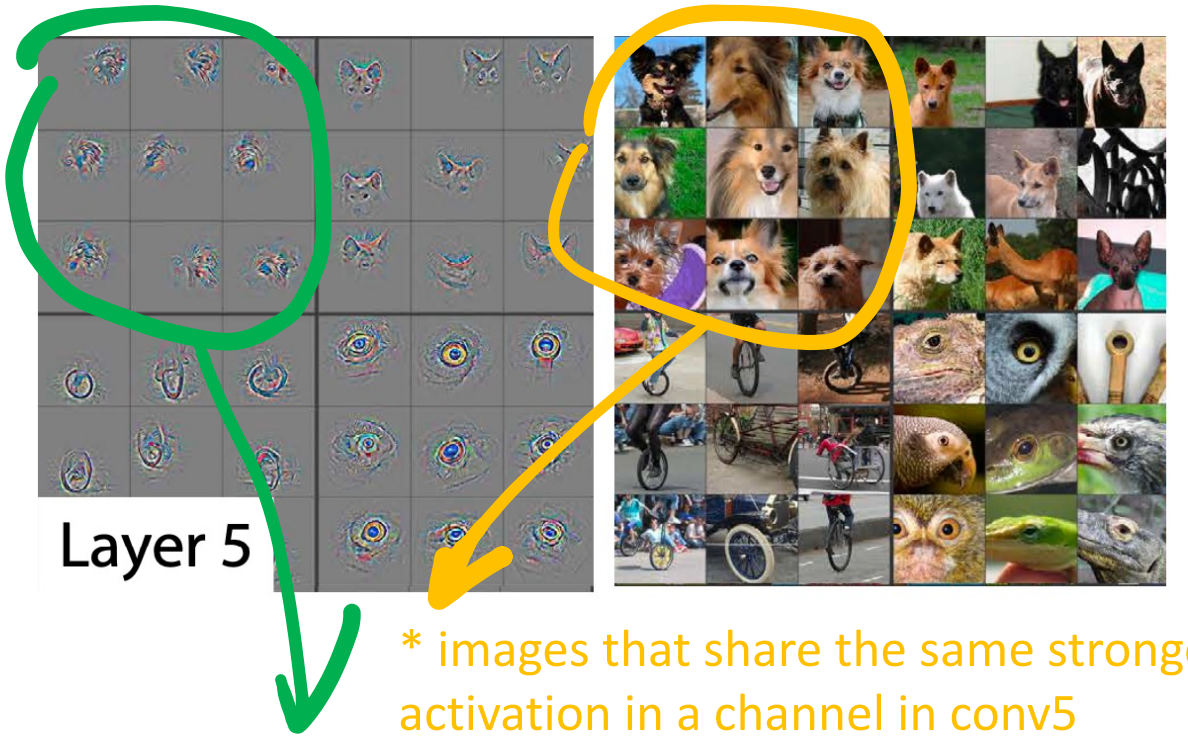
s.t. $\mathbf{z} = \mathrm{f}(\mathbf{x})$

What does a representation represent?







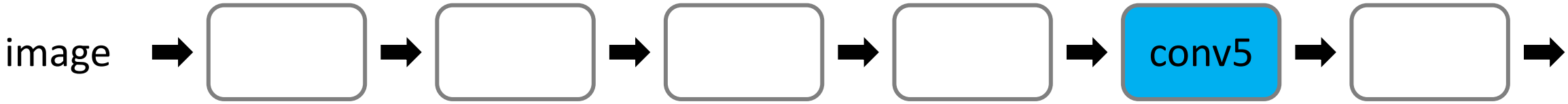


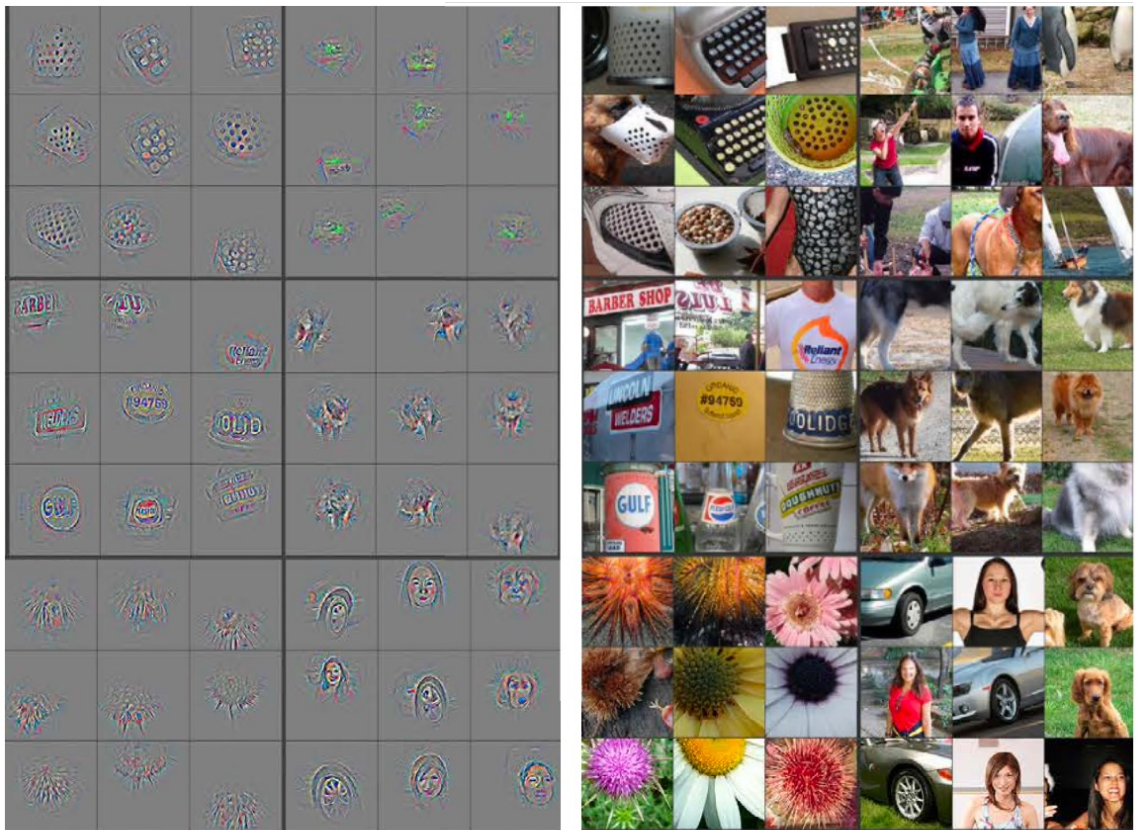
* images that share the same strongest activation in a channel in conv5

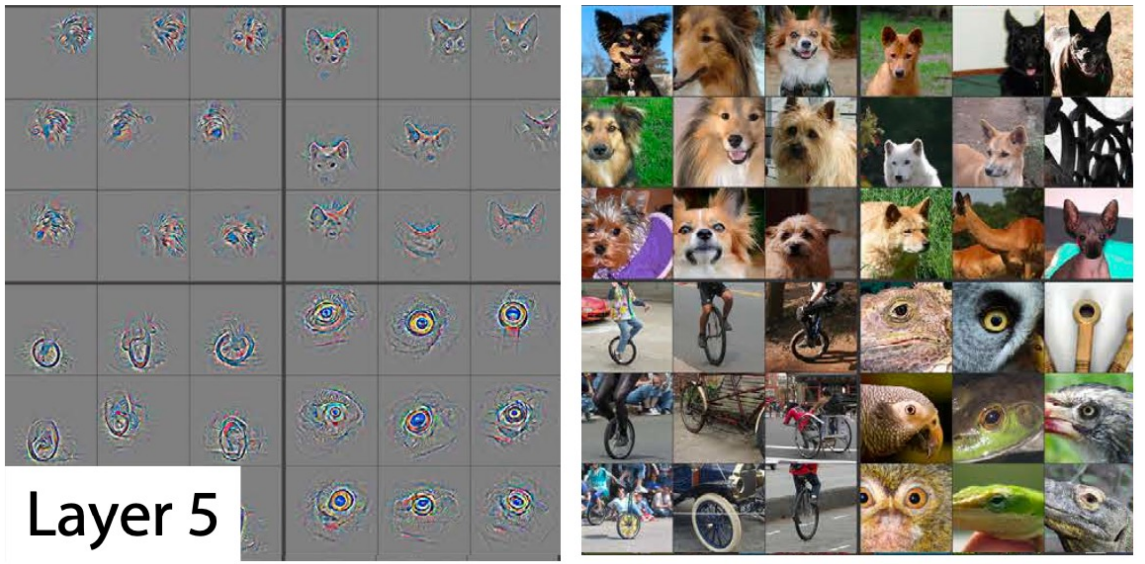
* backprop the 1-hot activation to pixels

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

What does a representation represent?

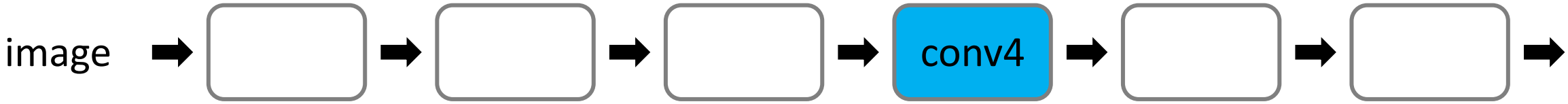


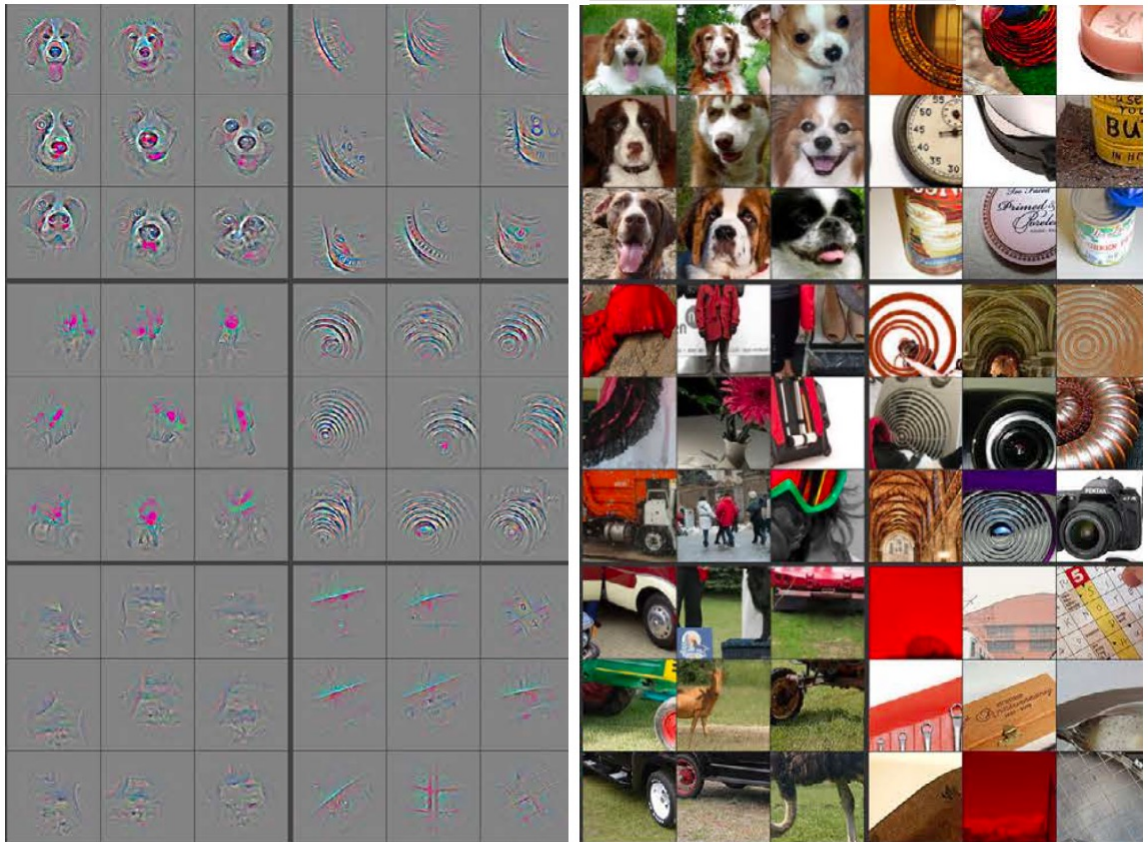


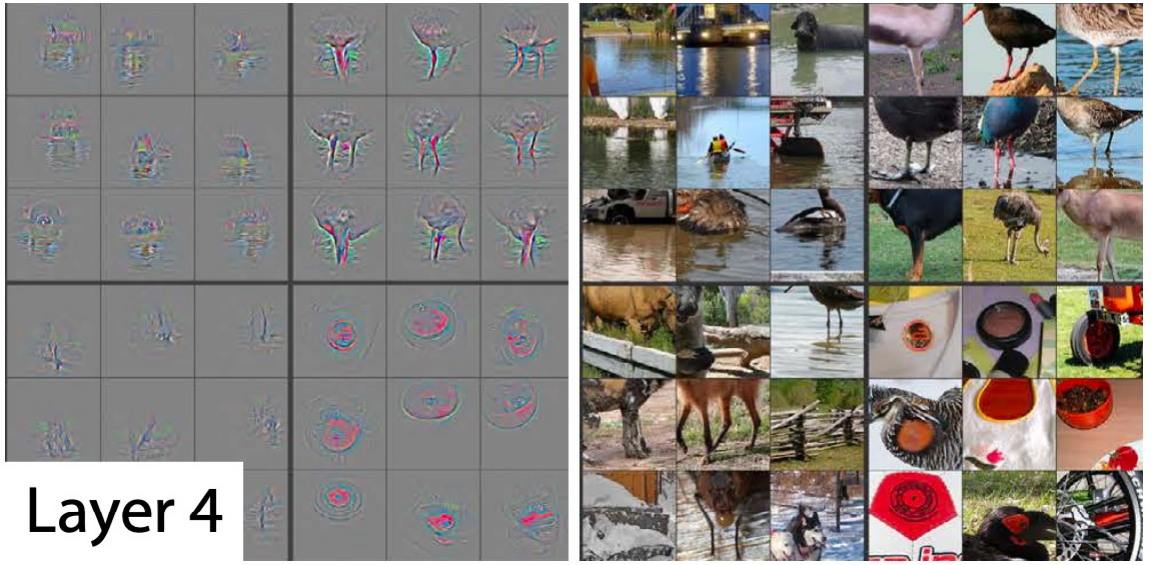


Deeper layers: higher-level abstraction

What does a representation represent?



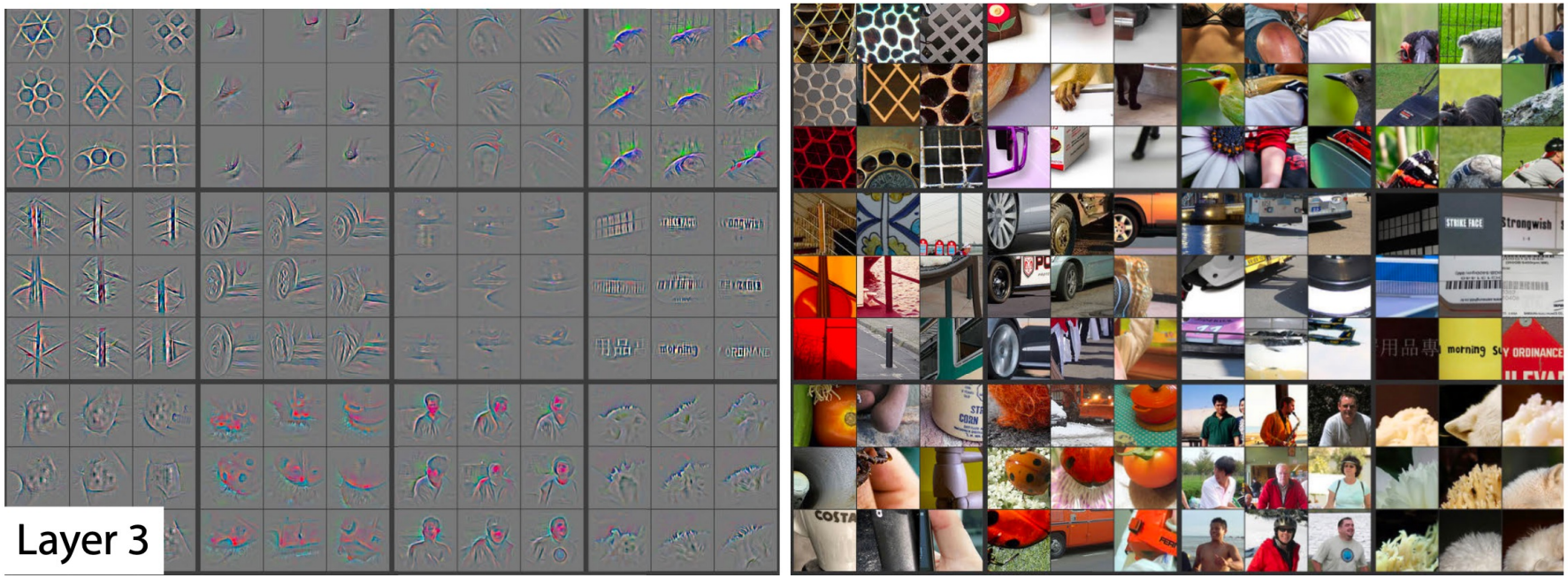




Deeper layers: higher-level abstraction

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

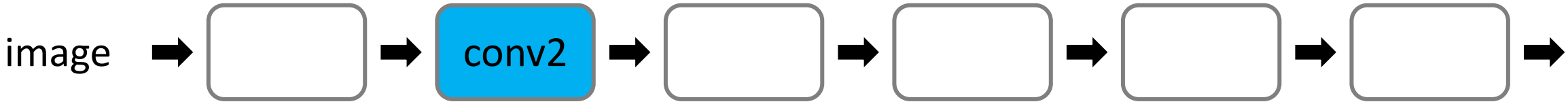
What does a representation represent?

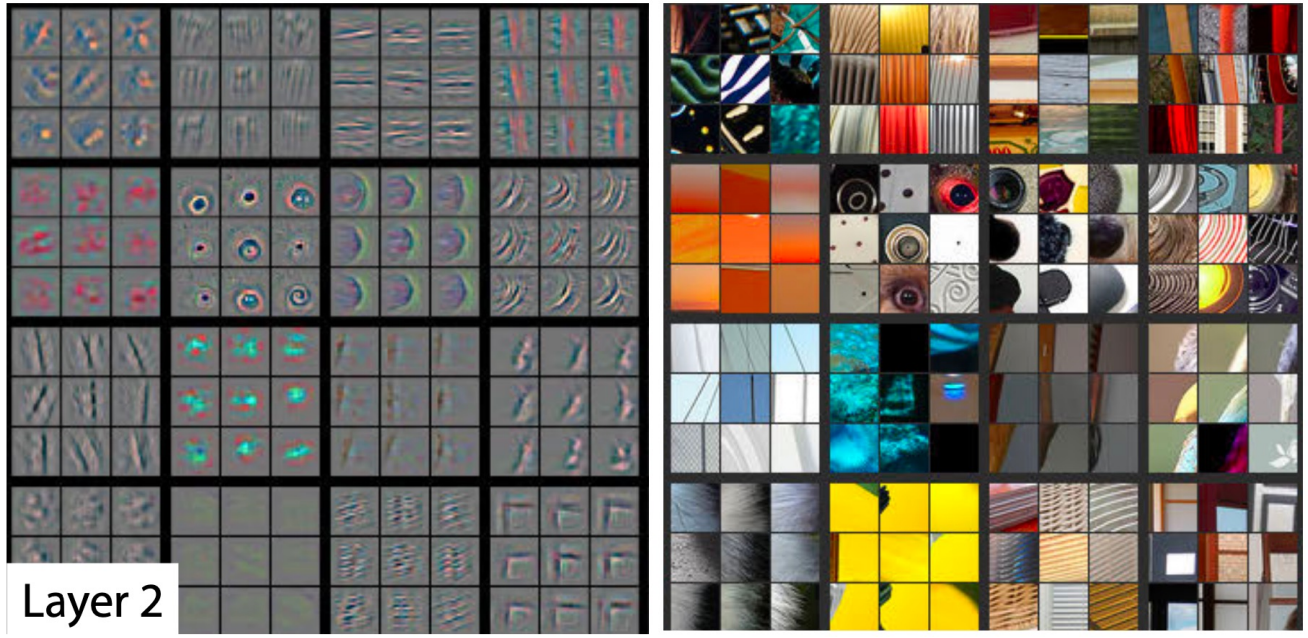


mid-level abstraction

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

What does a representation represent?

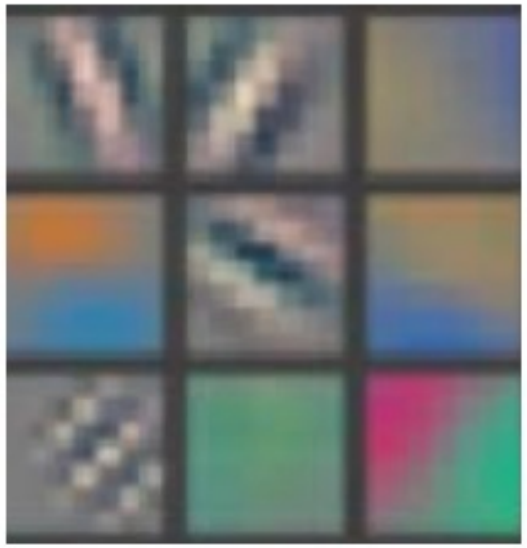




low-level abstraction

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

What does a representation represent?



low-level abstraction

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

Deep Representations are Transferrable

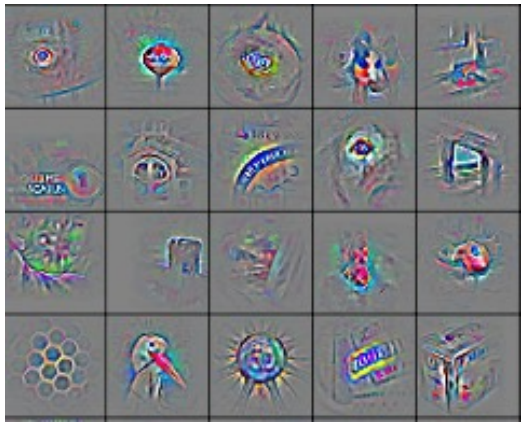
Transfer learning:

- pre-train: large data
- fine-tune: small data

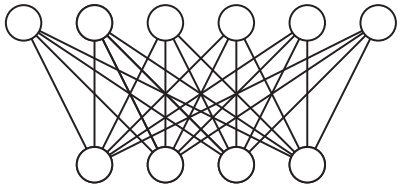
- enable DL for small data
- revolutionize CV and many areas (LLM)

- data: engine for learning representations

The single most important discovery in DL revolution!



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

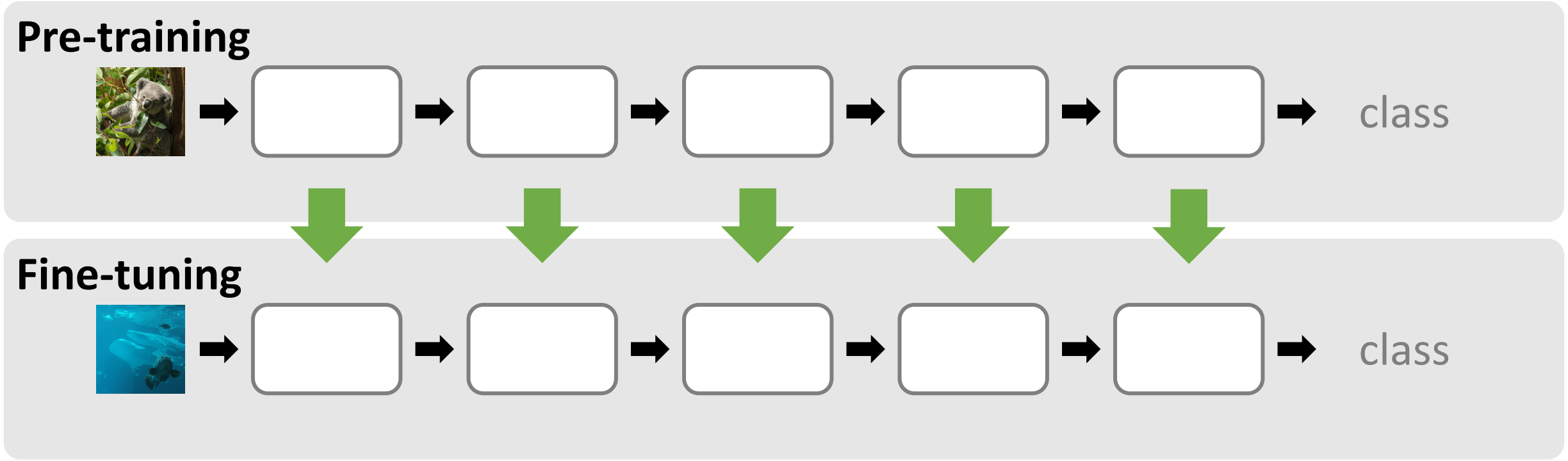
Transfer learning

Pre-training

Pre-training:

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

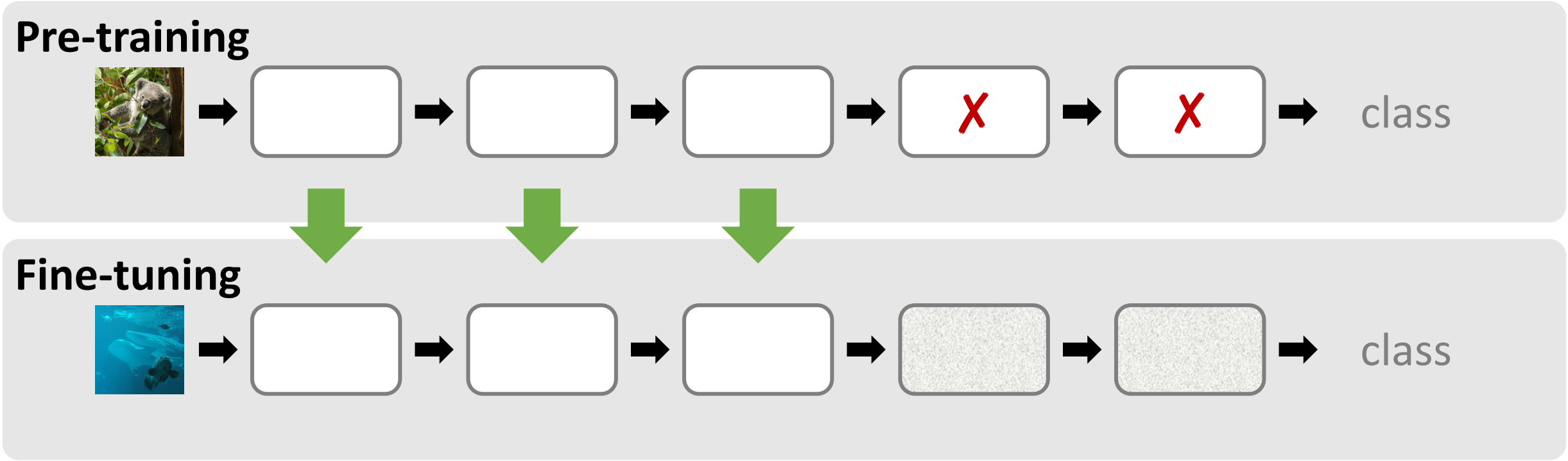
Transfer learning



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

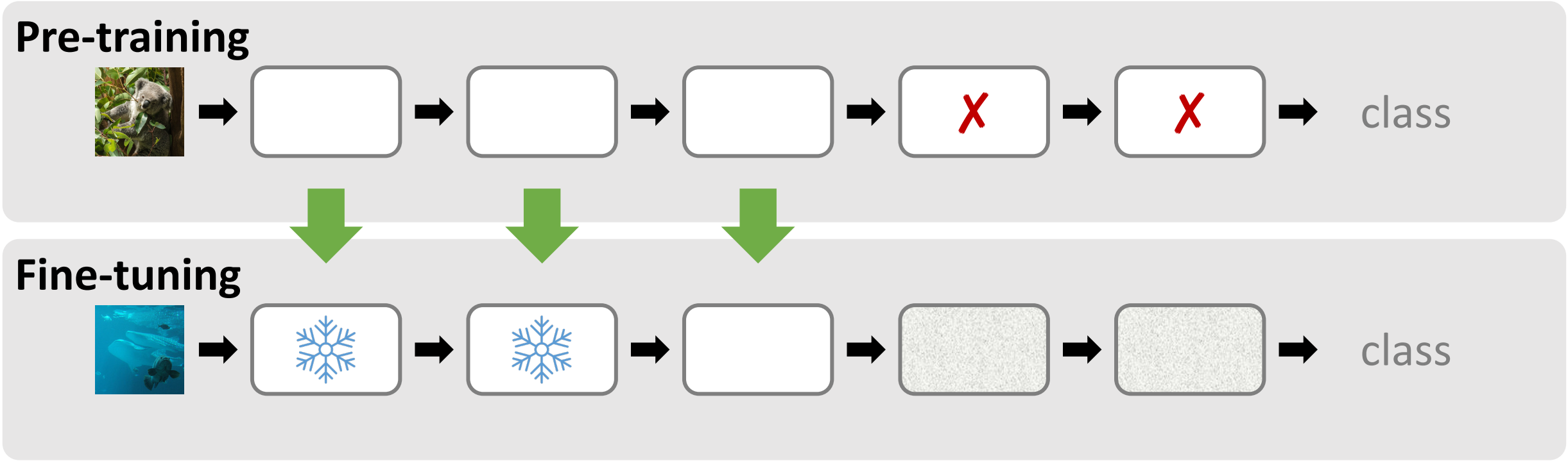
Transfer learning



Partial transfer

- pre-train and target domains may differ
- high-level representations too specialized
- randomly initialize new layers

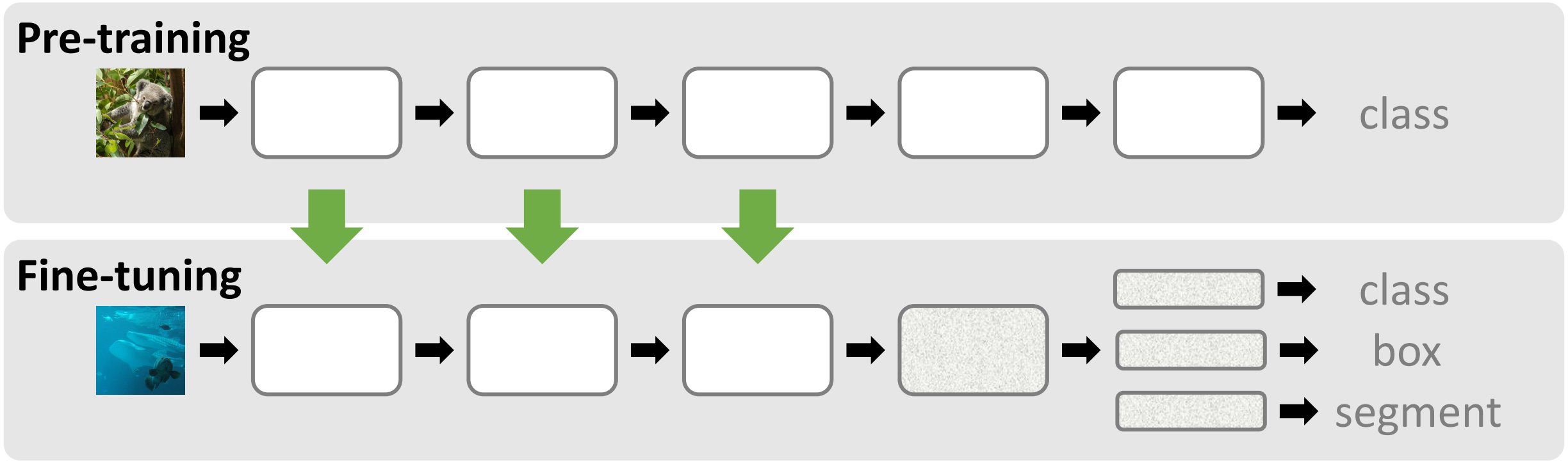
Transfer learning



Frozen weights

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

Transfer learning



Network surgery

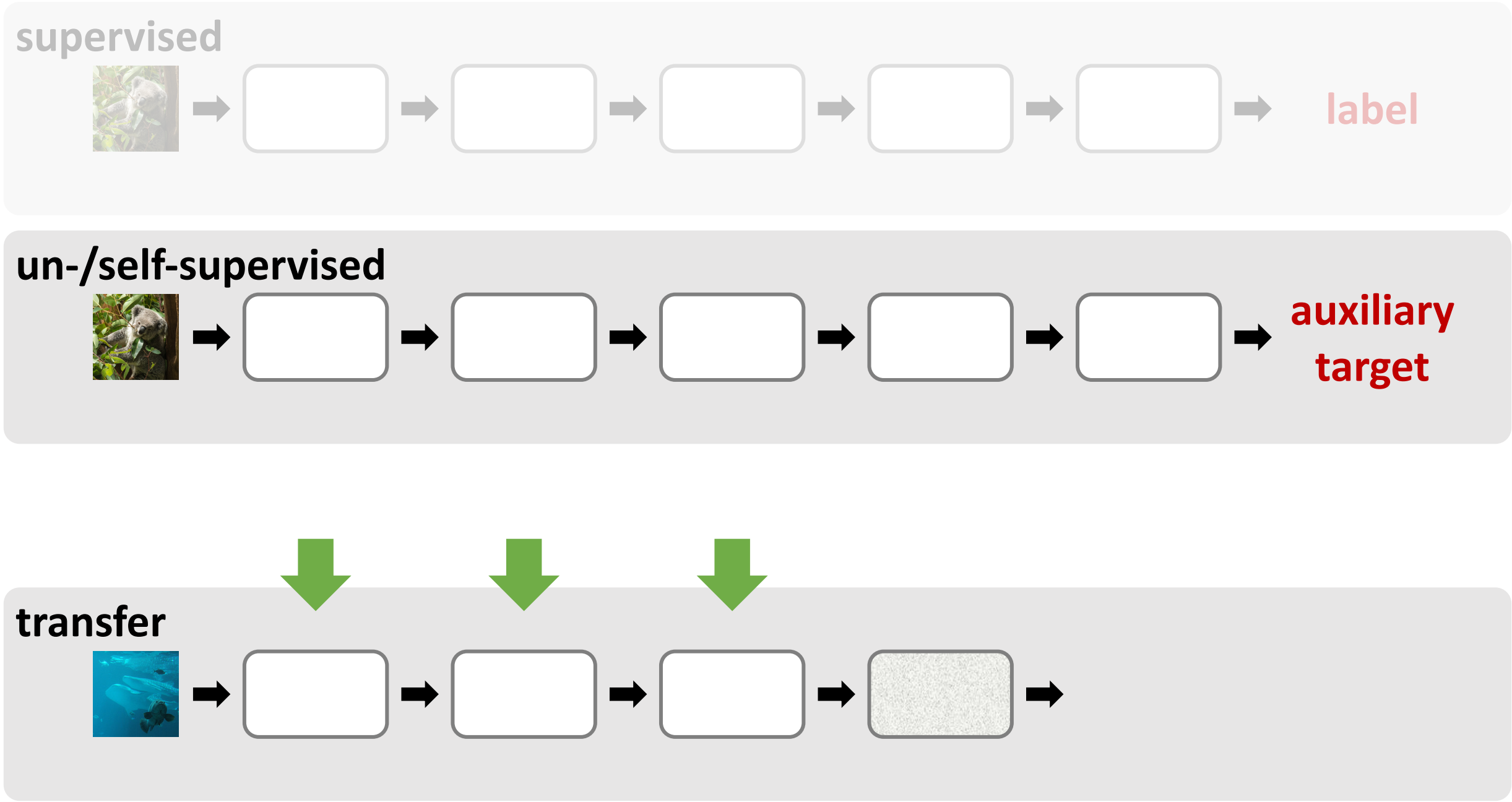
- re-purpose for other tasks (detect, segment)
- general representations + task-specific predictions

Pre-training Objectives

- **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


supervised
label
un-/self-supervised
auxiliary target
transfer

Reconstruction-based Methods

for Unsupervised Representation Learning

Reconstruction: classical yet highly powerful

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

Reconstruction: classical yet highly powerful

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

K-means clustering

Represent data by k centers

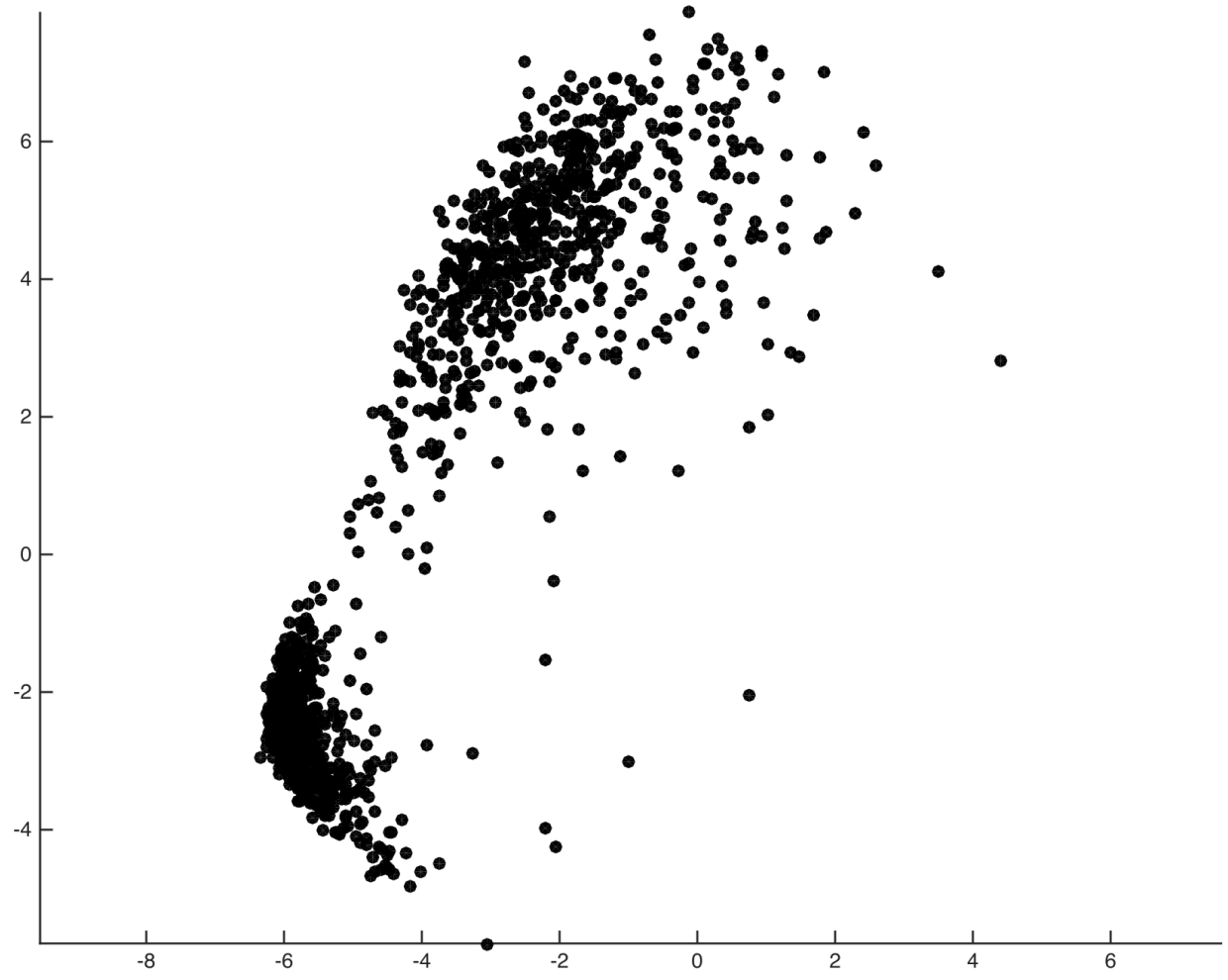


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

K-means clustering

Represent data by k centers

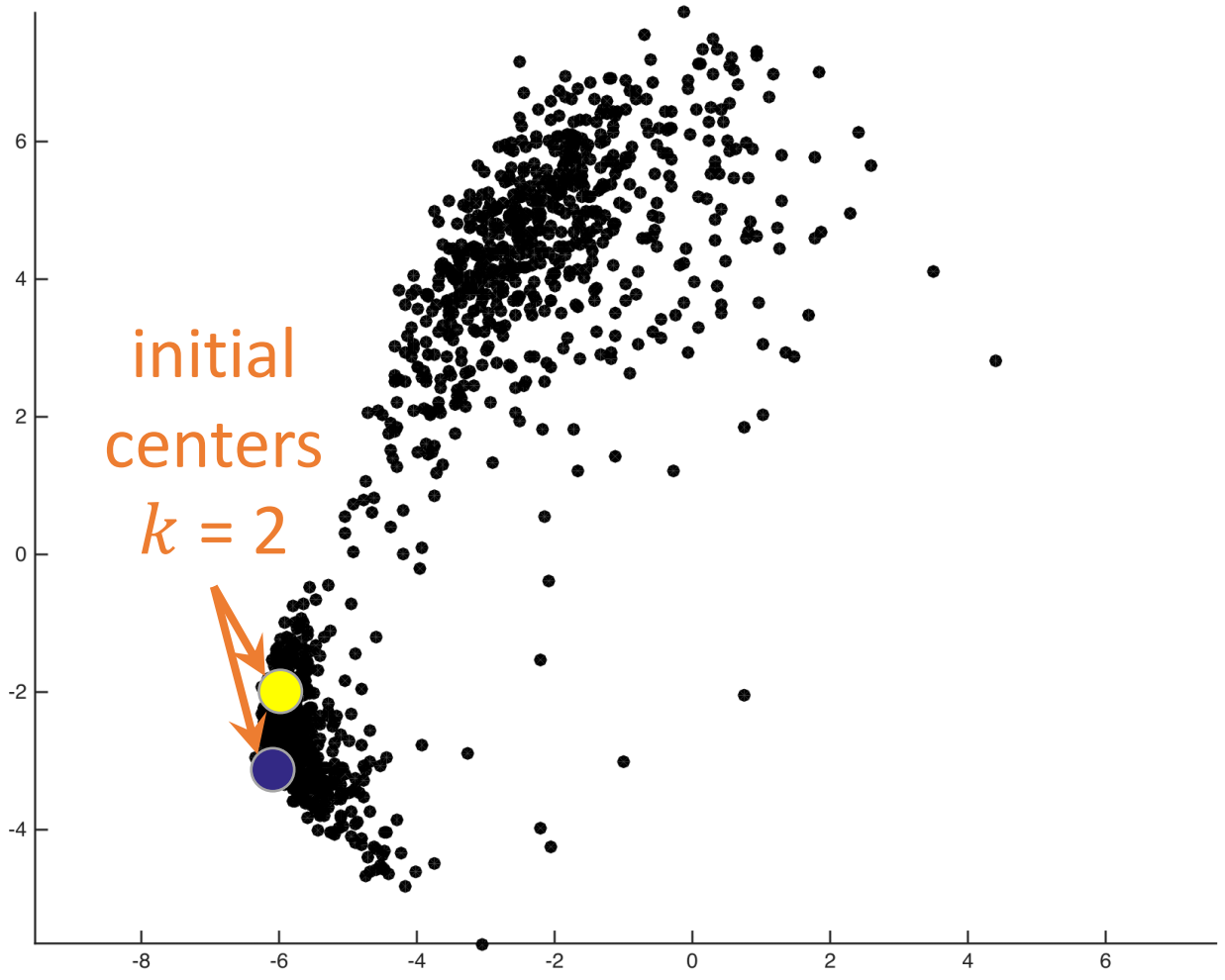


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

K-means clustering

Represent data by k centers

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

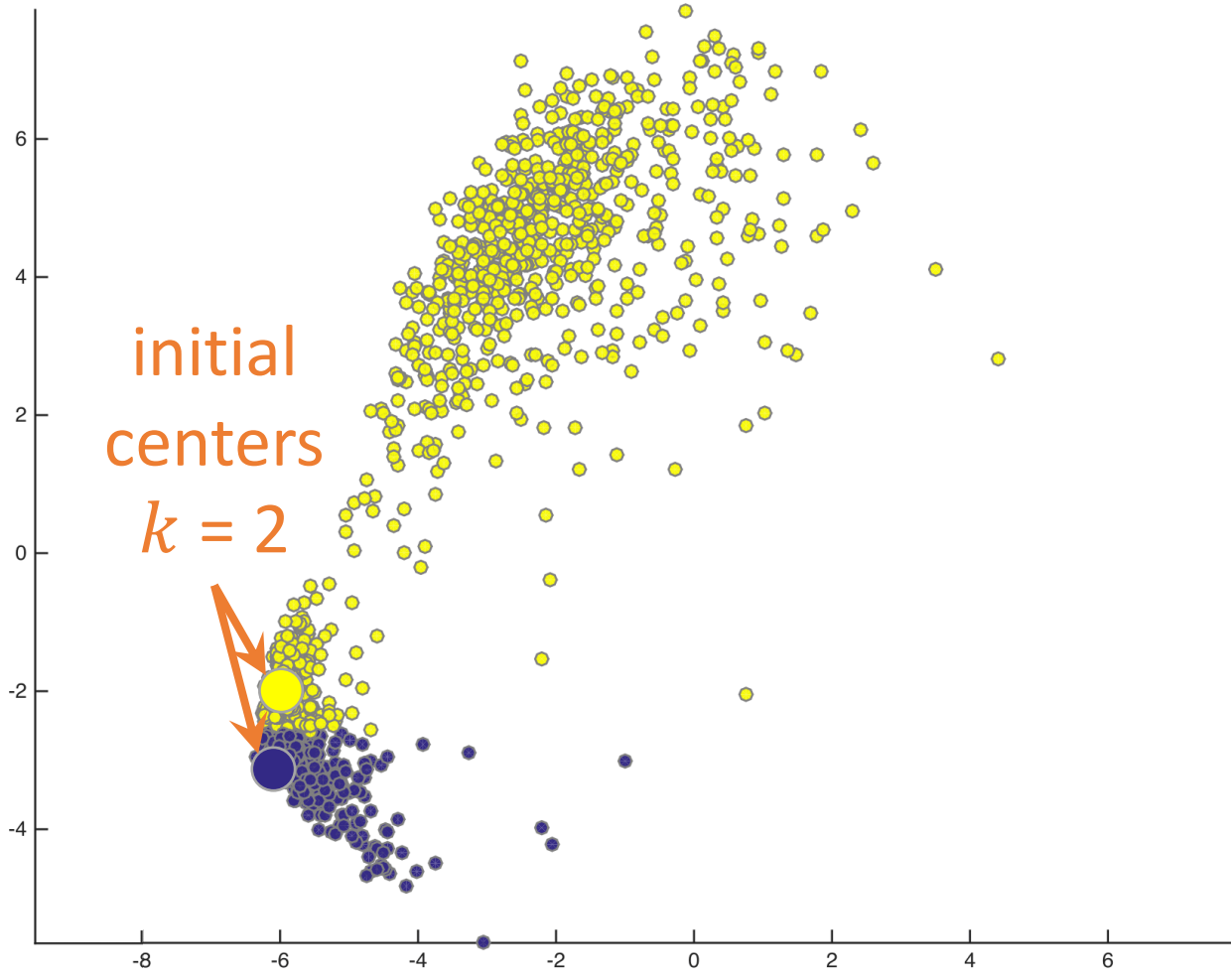


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

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

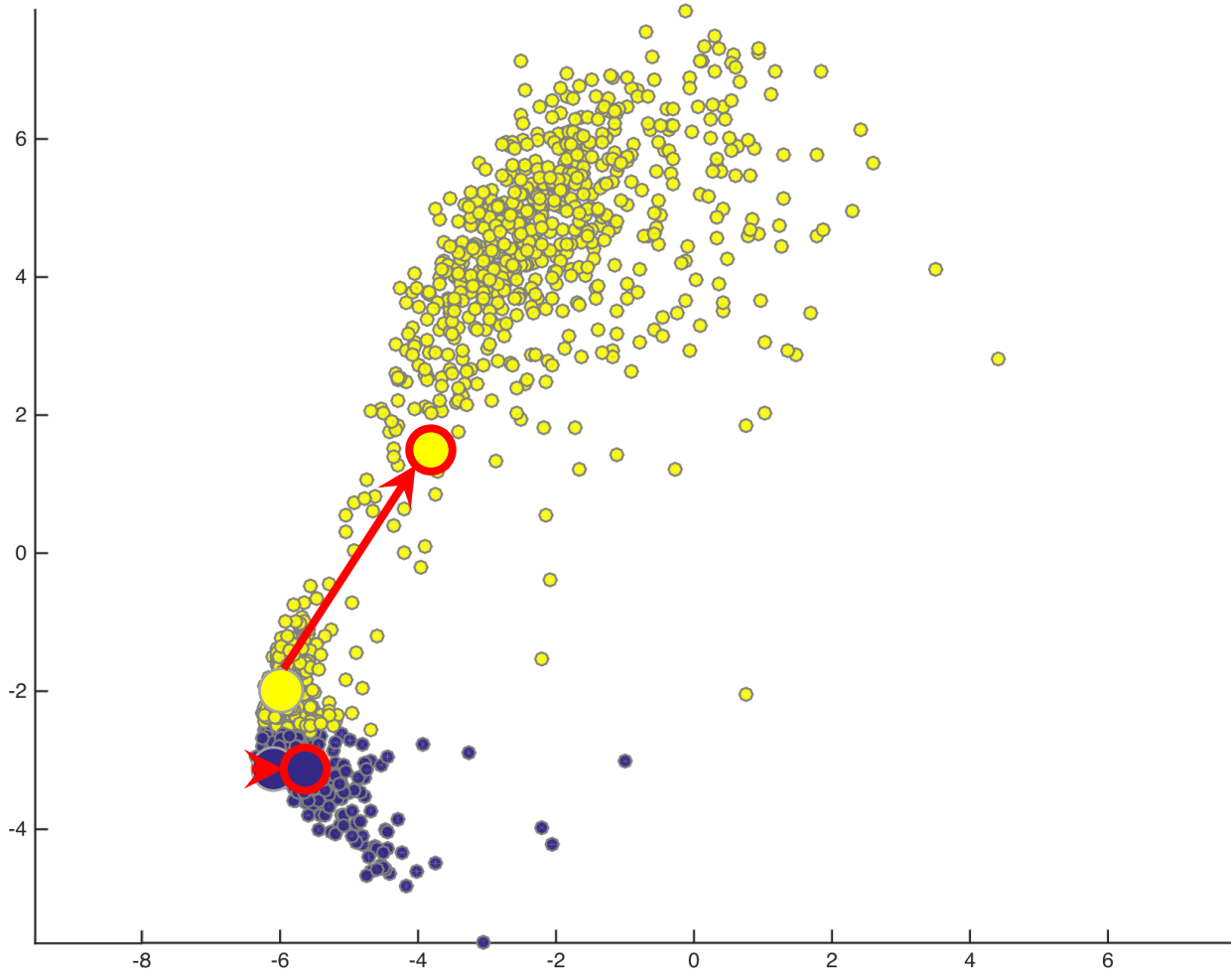


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

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

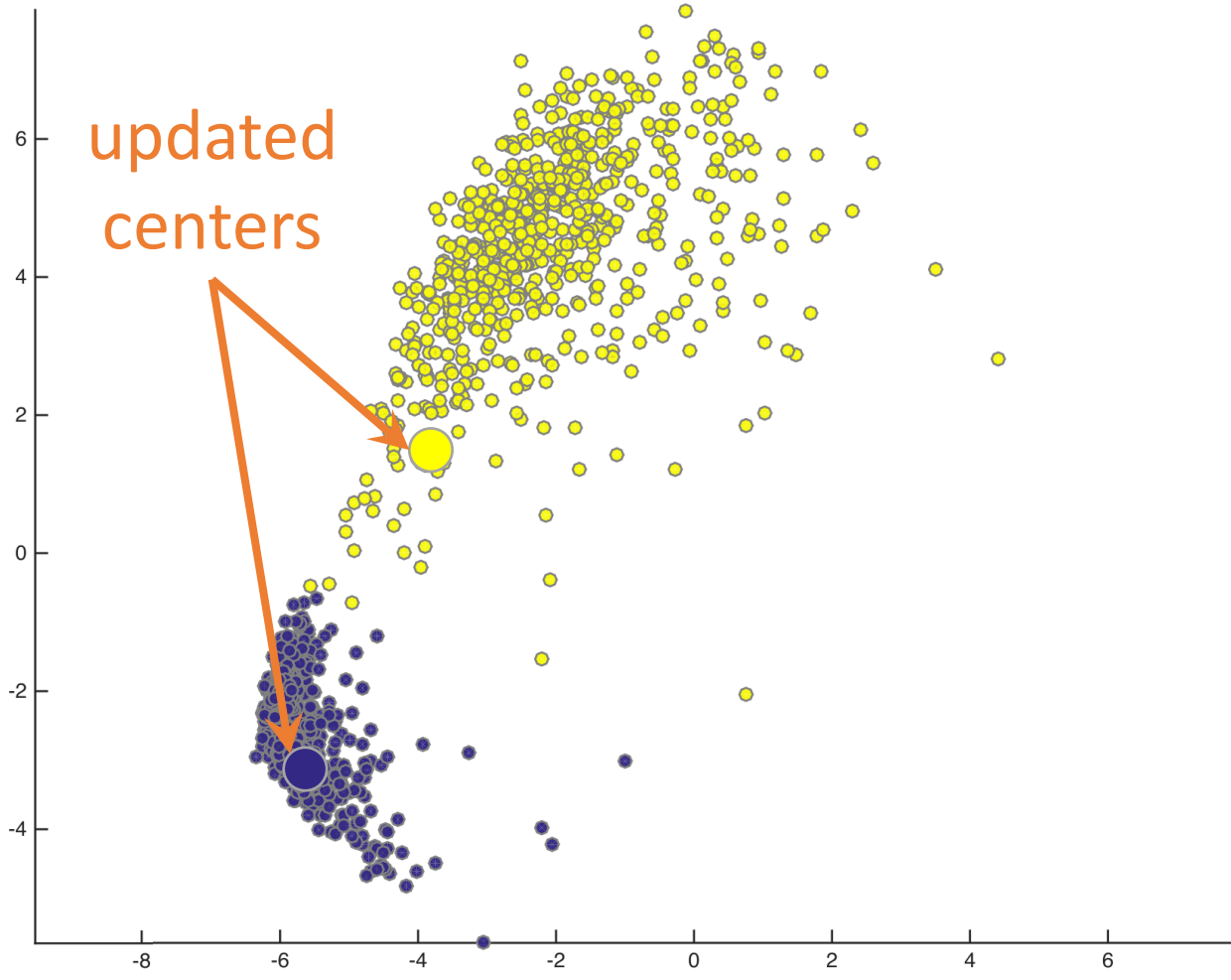


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

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

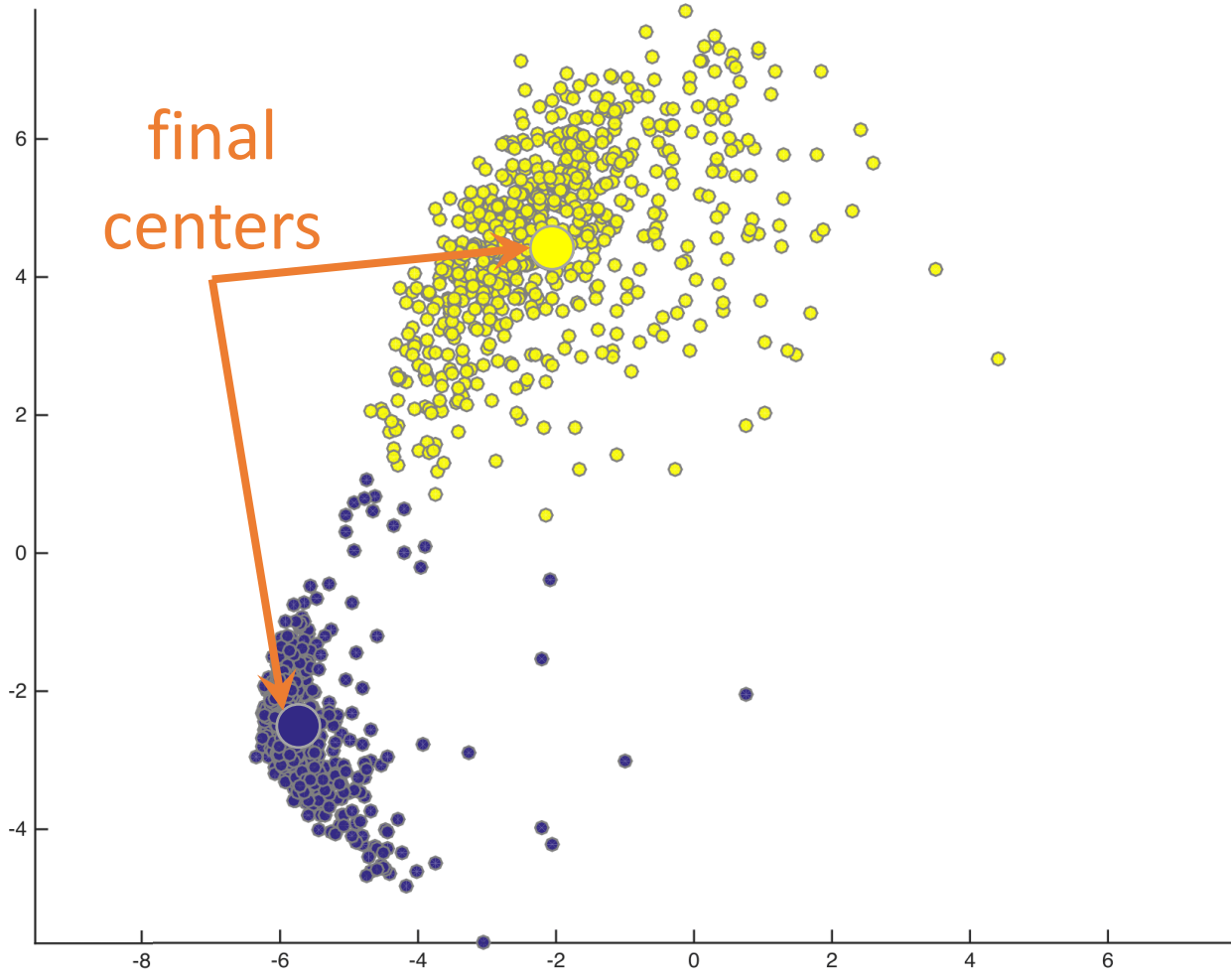


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

K-means clustering

Loss function:

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

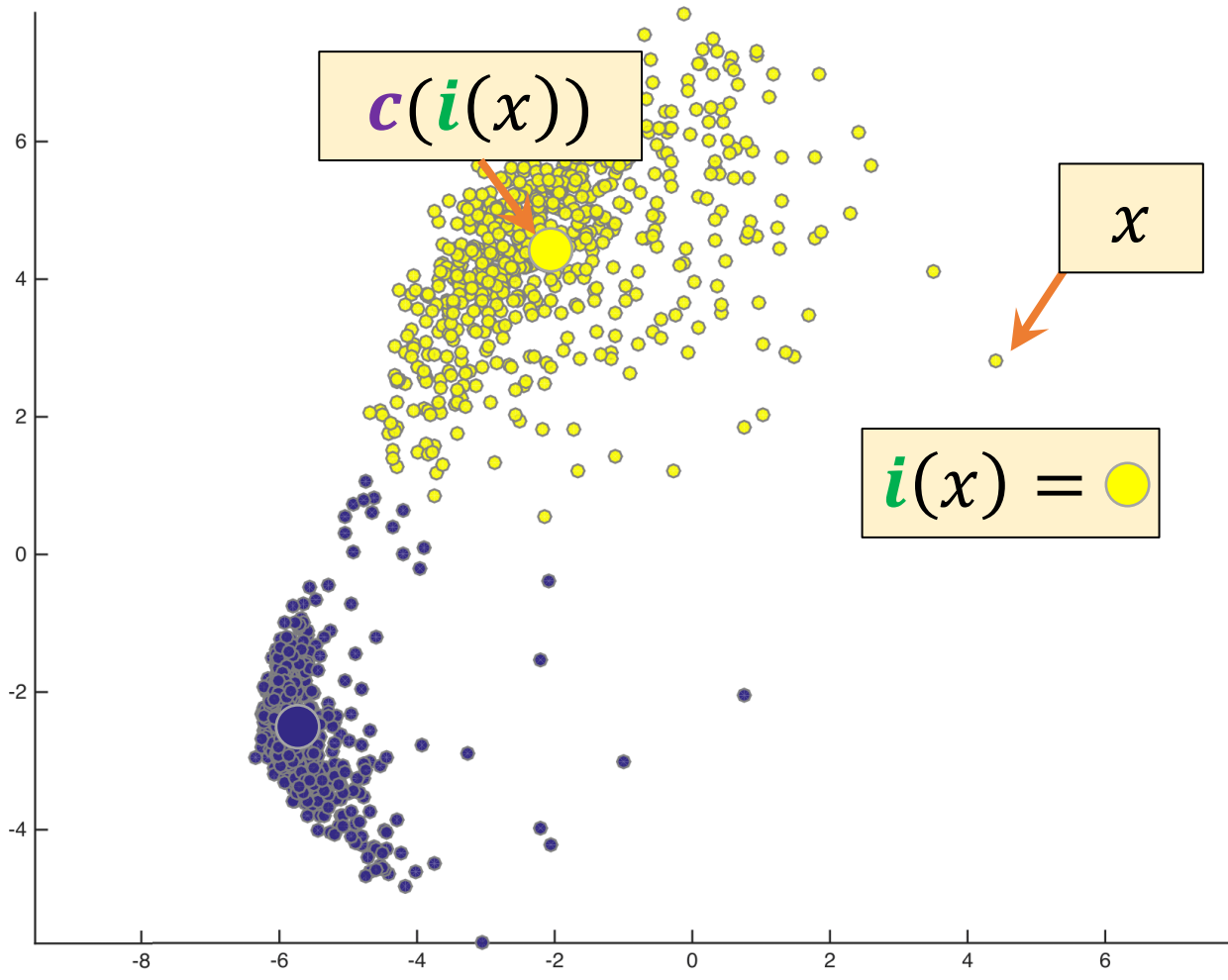


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

K-means clustering

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

Loss function:

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

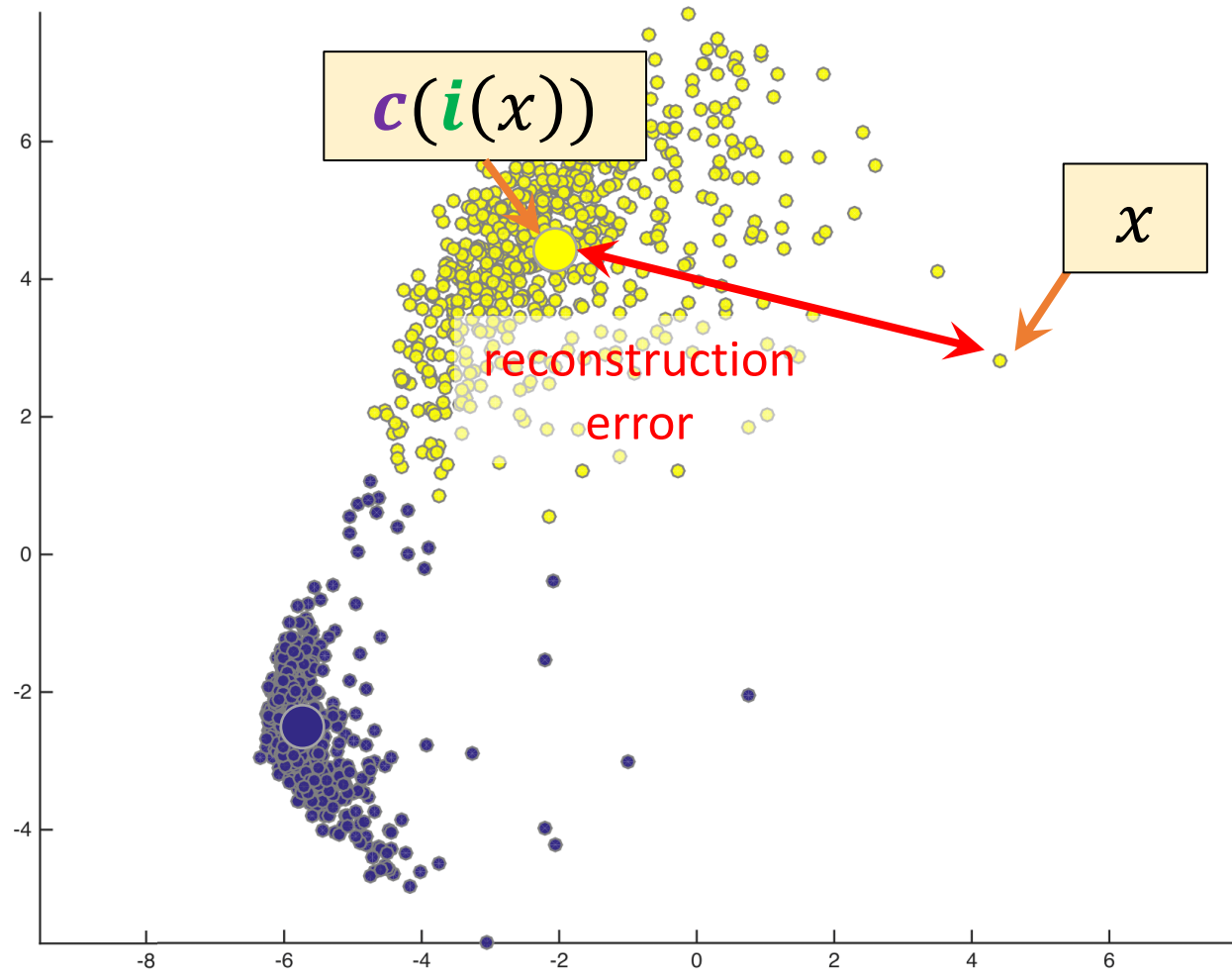


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

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 $\boldsymbol{i}$

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

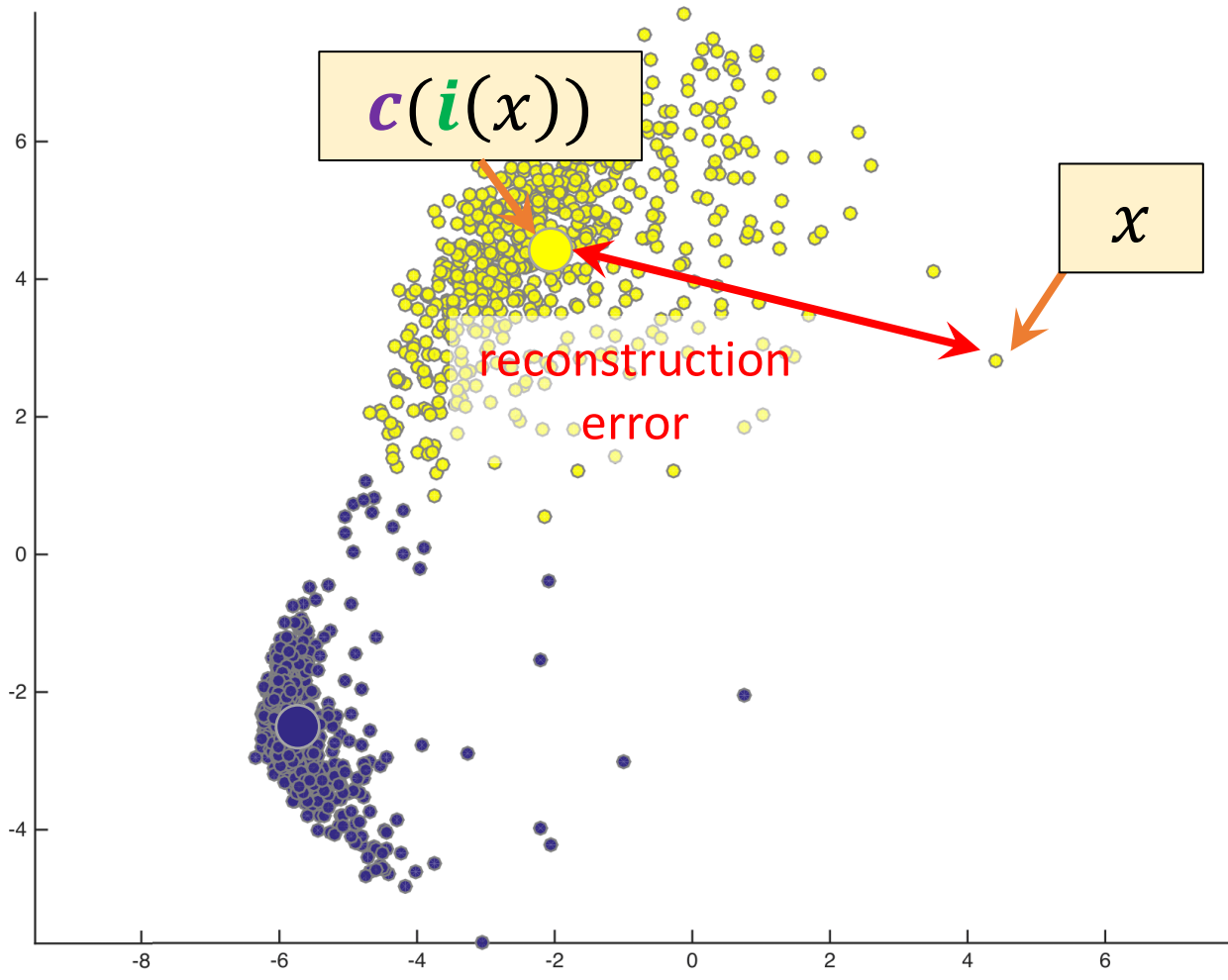


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

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 $\boldsymbol{c}$

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

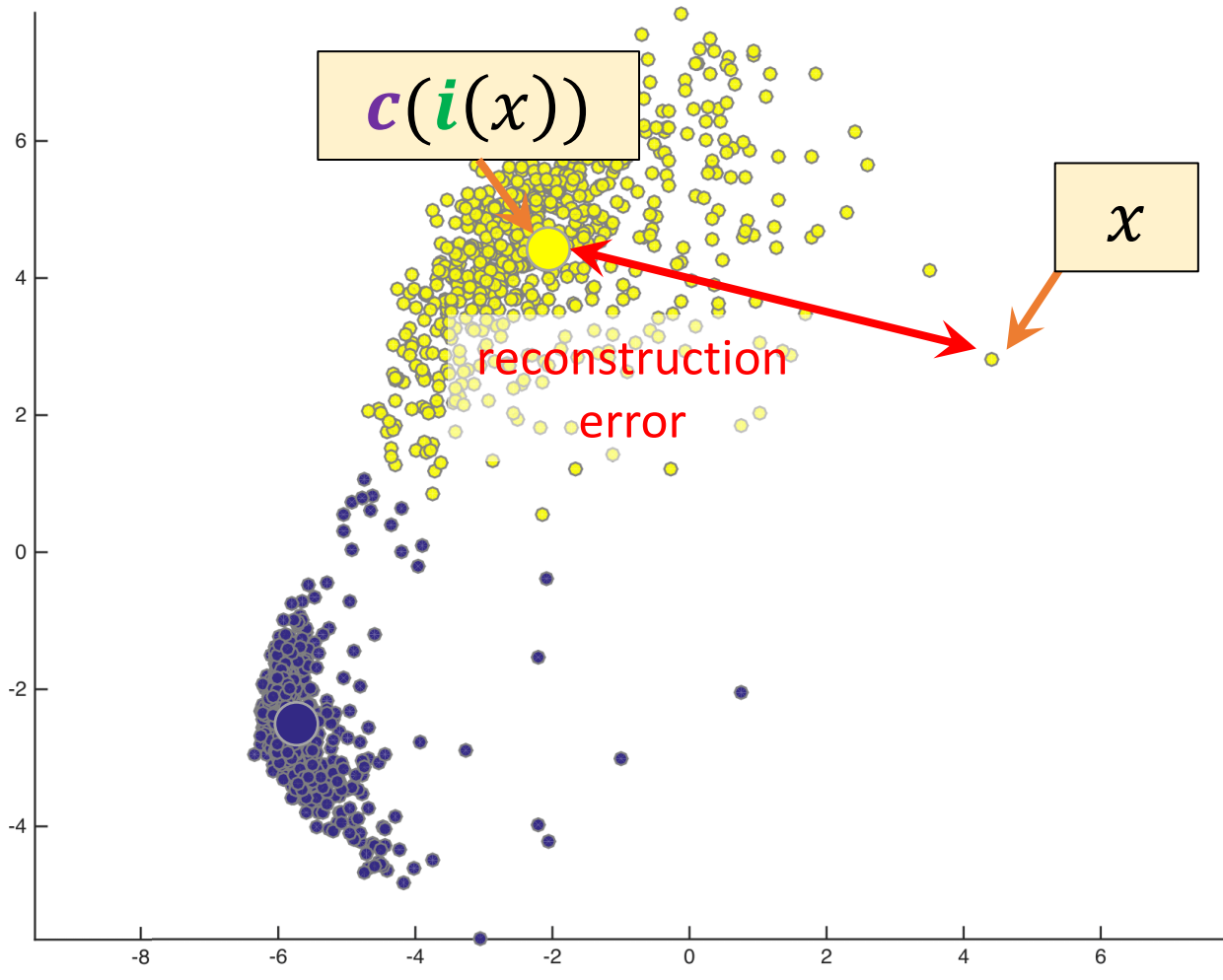


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

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

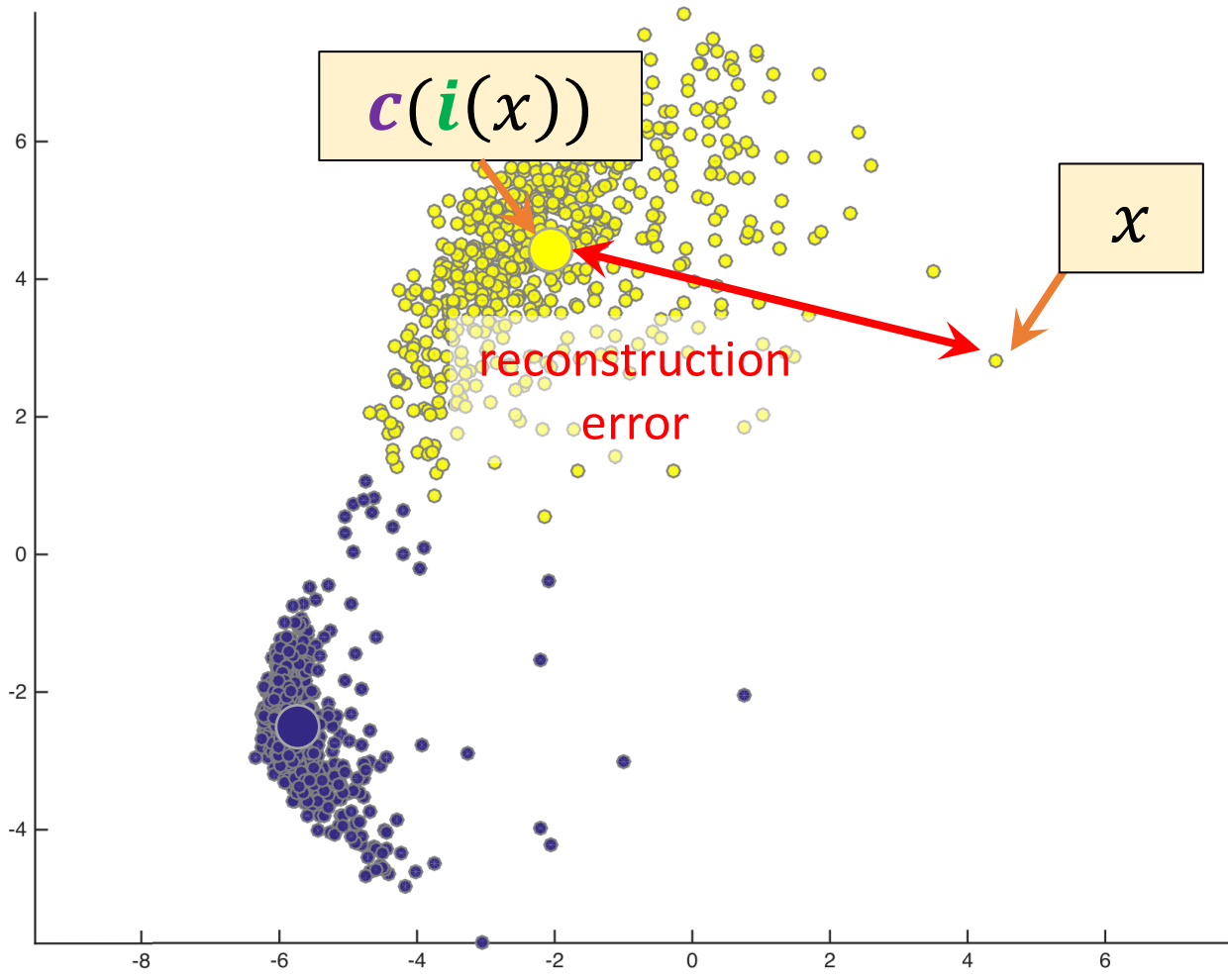


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

K-means clustering

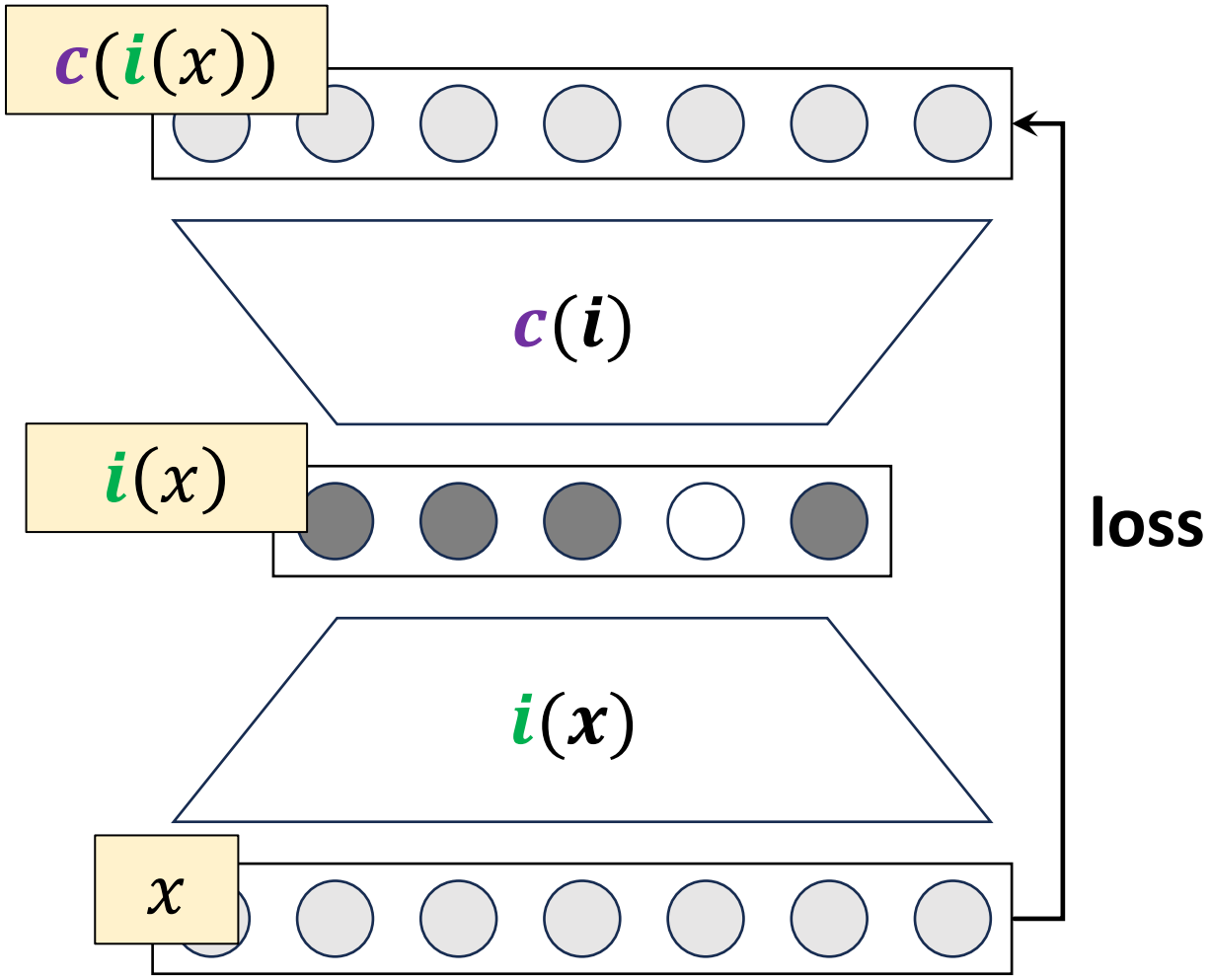
Loss function:

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

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

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

Repeat until converge



K-means clustering

Loss function:

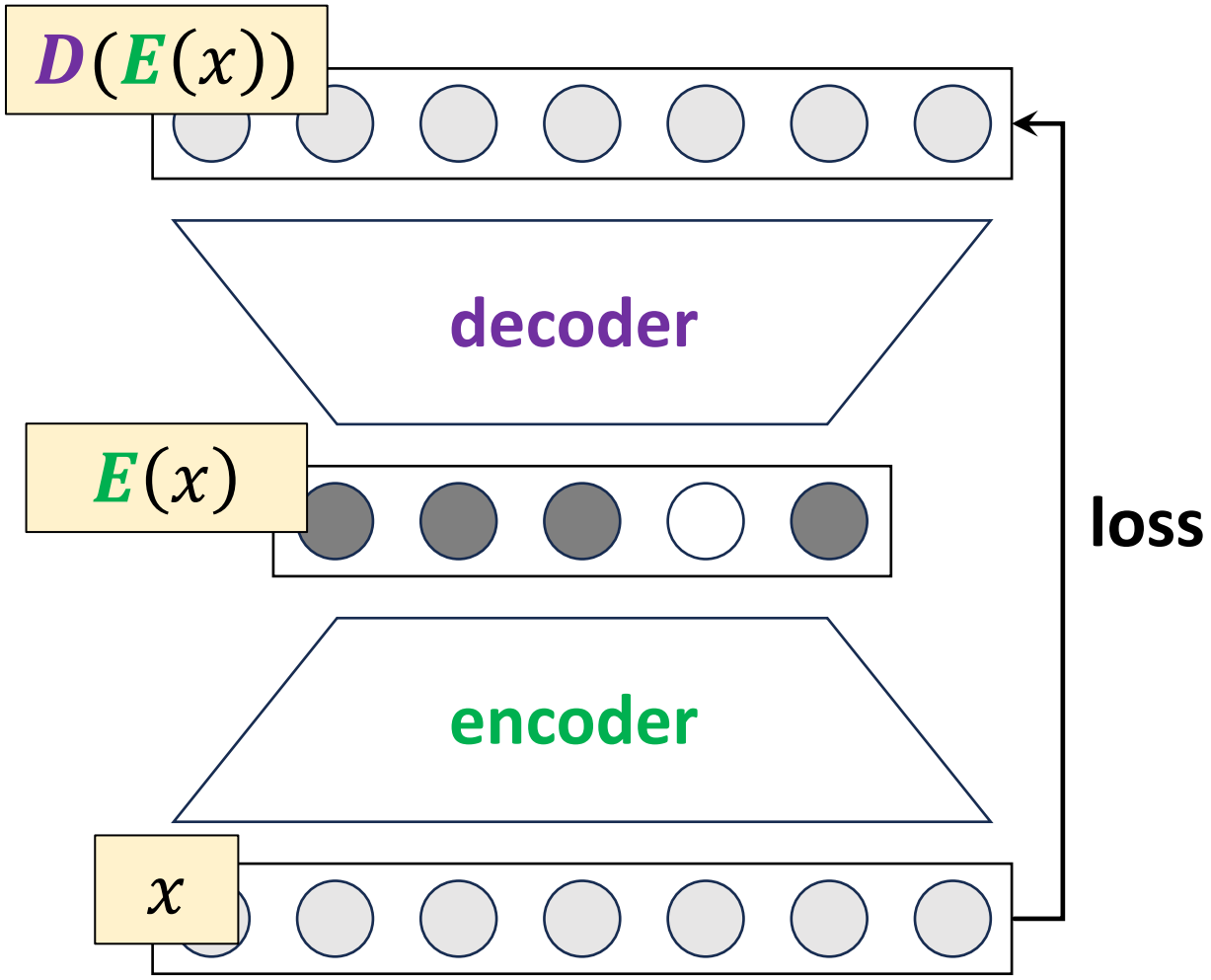
$$\min_{\boldsymbol{D},\boldsymbol{E}} \sum_{x} \|x - \boldsymbol{D}(\boldsymbol{E}(x))\|^2$$

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

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

K-means performs Autoencoding

- w/ one-hot encoding
- w/ coupled encoder & decoder

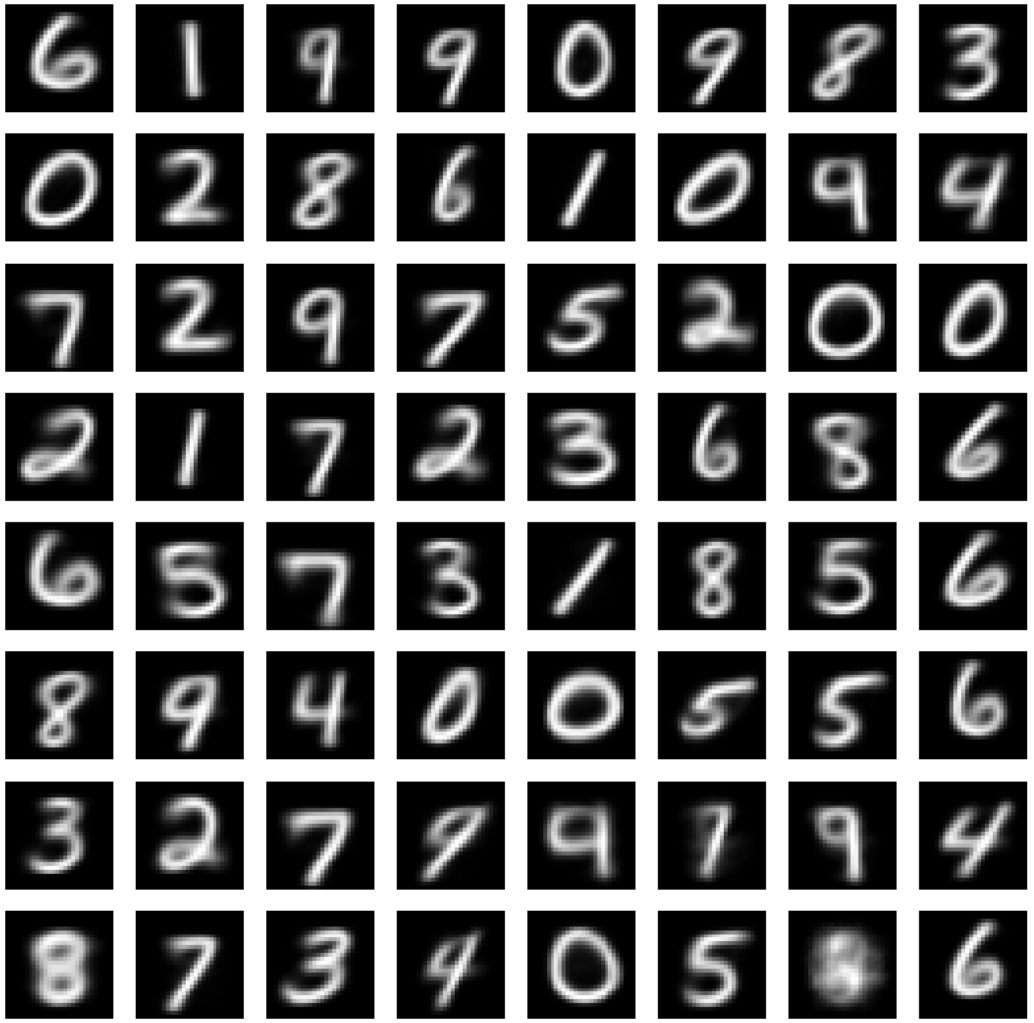


Example: K-means on MNIST digits

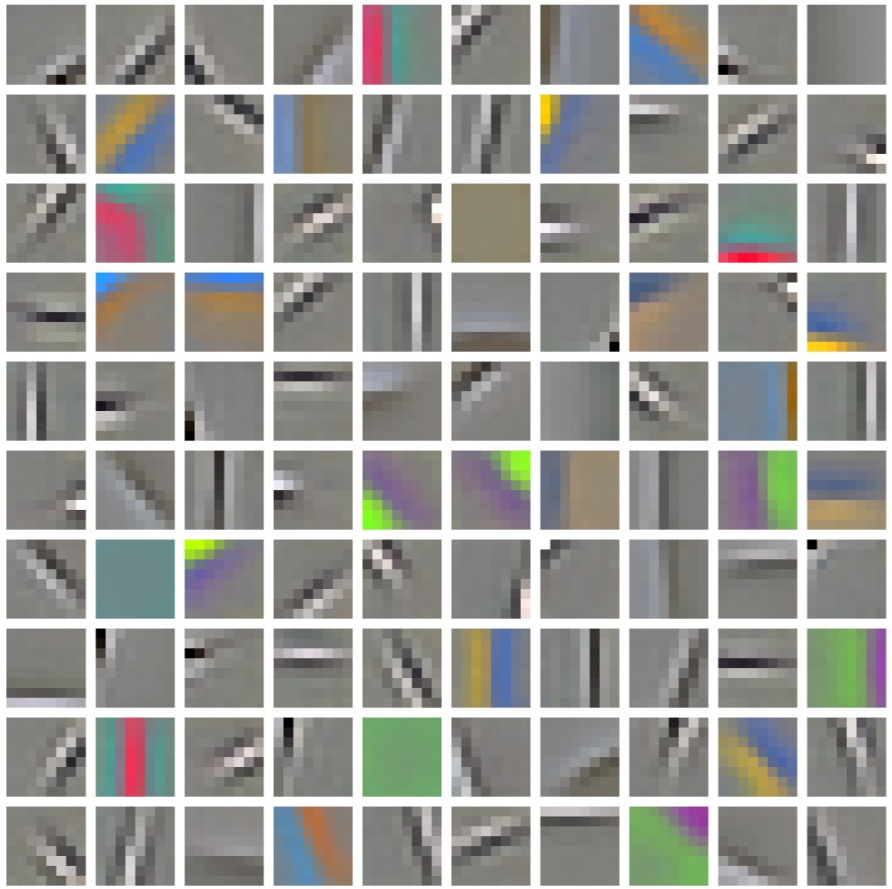
These are not real images.

These are the learned **representations**.

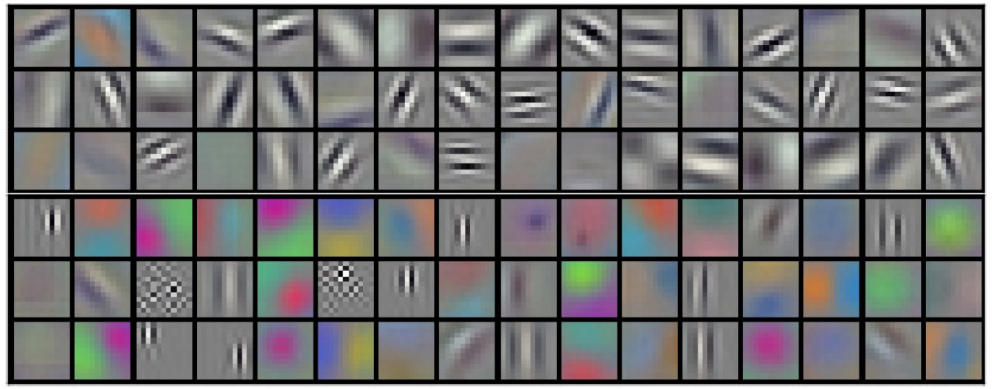
Visualized centers with k = 64



Example: K-means on 8×8 image patches



K-means
(w/ PCA whitening)

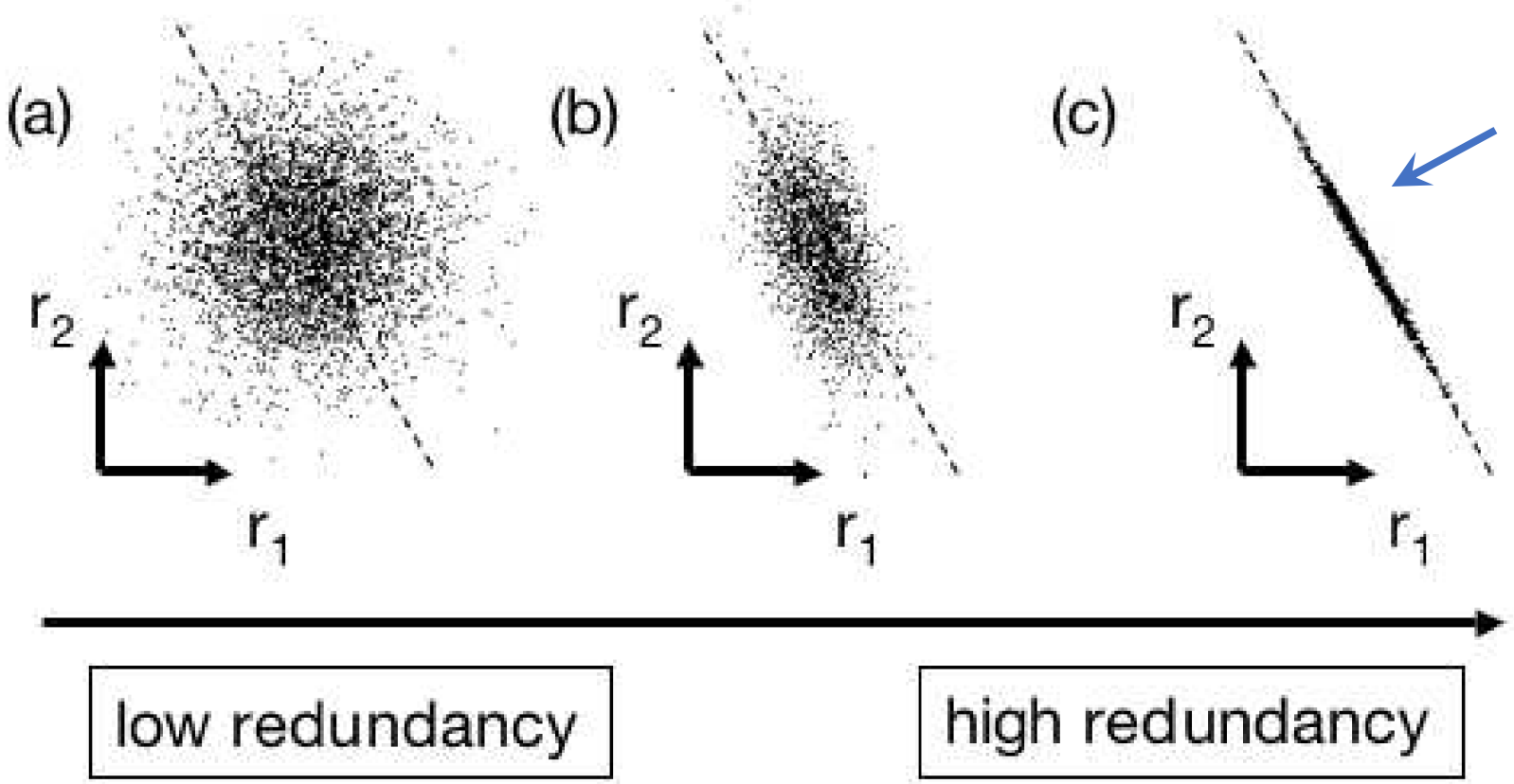


1st layer of AlexNet
(supervised learning)

Reconstruction: classical yet highly powerful

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

Principal Component Analysis (PCA)



One can approximately use r_1 to **represent** r_2.

(r_1 , r_2): not a good **representation**.

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

Principal Component Analysis (PCA)

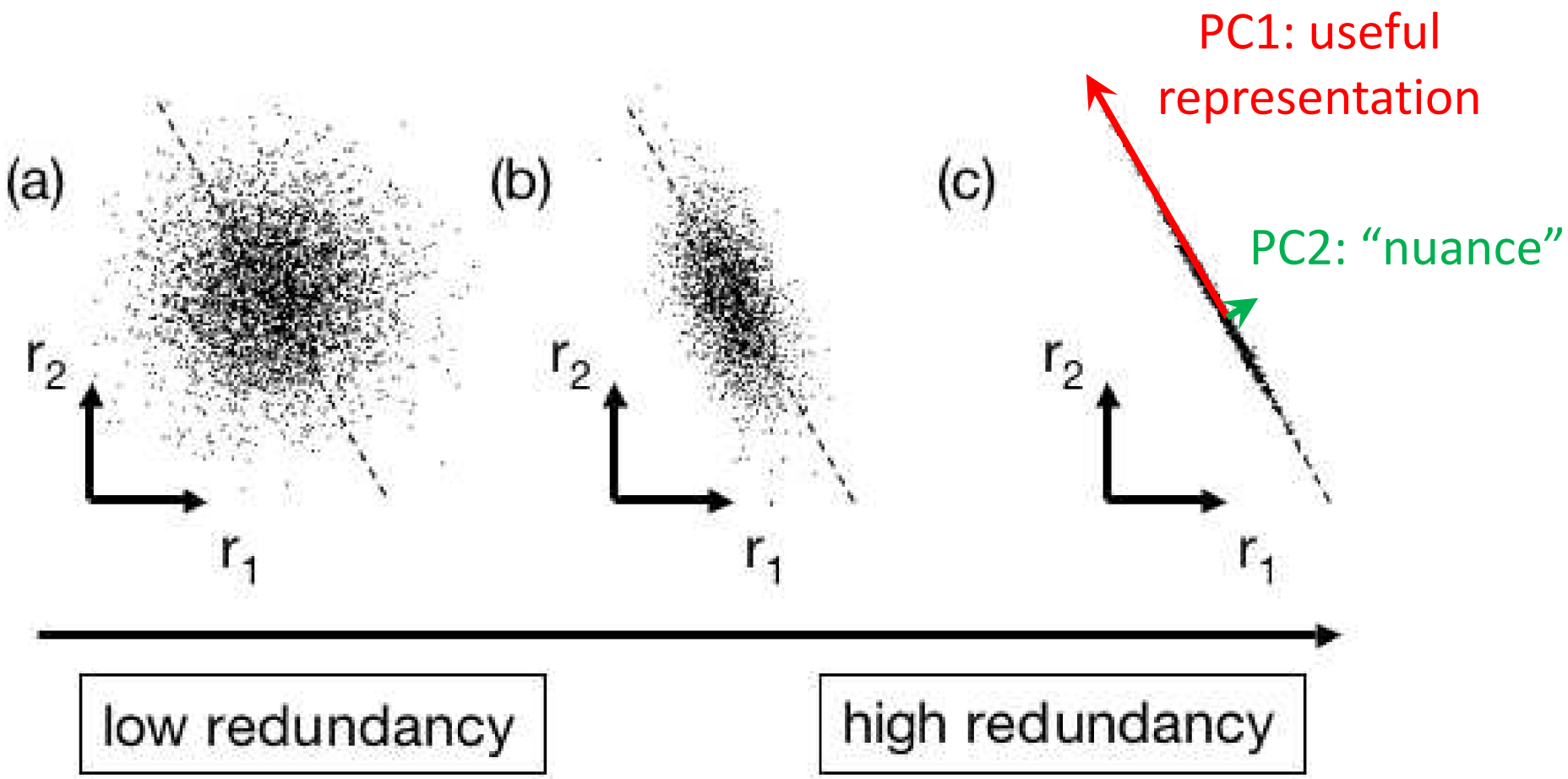


Figure credit: Jonathon Shlens, “A Tutorial on Principal Component Analysis”, 2005

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

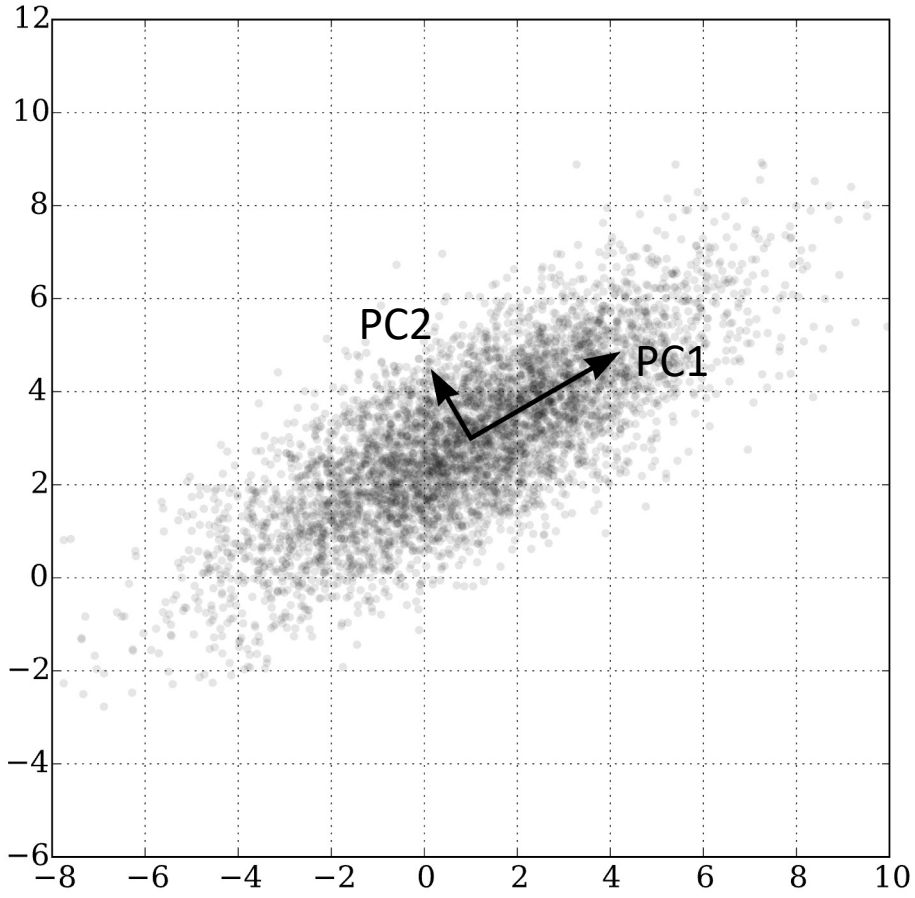


figure credit: https://en.wikipedia.org/wiki/Principal_component_analysis

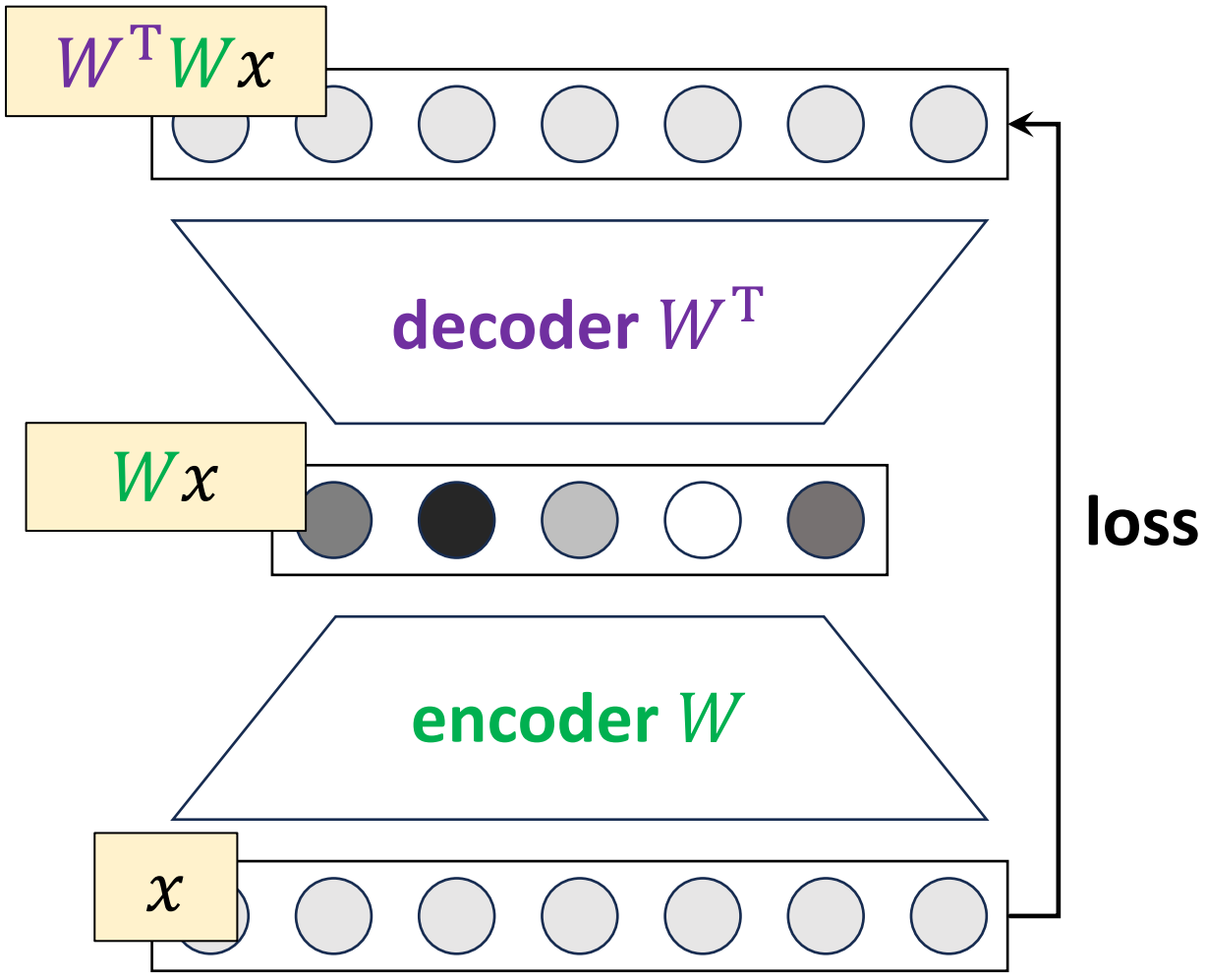
Principal Component Analysis (PCA)

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



Example: PCA on face data (Eigenfaces)

These are not real 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

Reconstruction: classical yet highly powerful

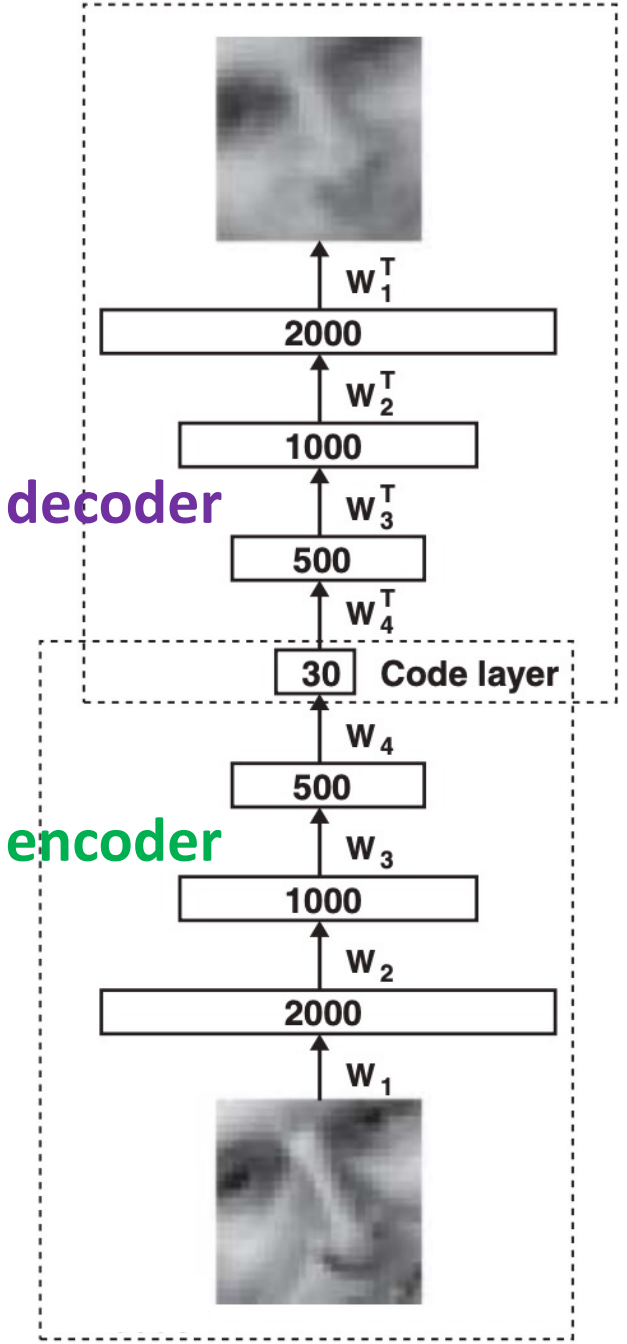
- K-means clustering
- Principal Component Analysis (PCA)
- Autoencoder

Autoencoder [2006]

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

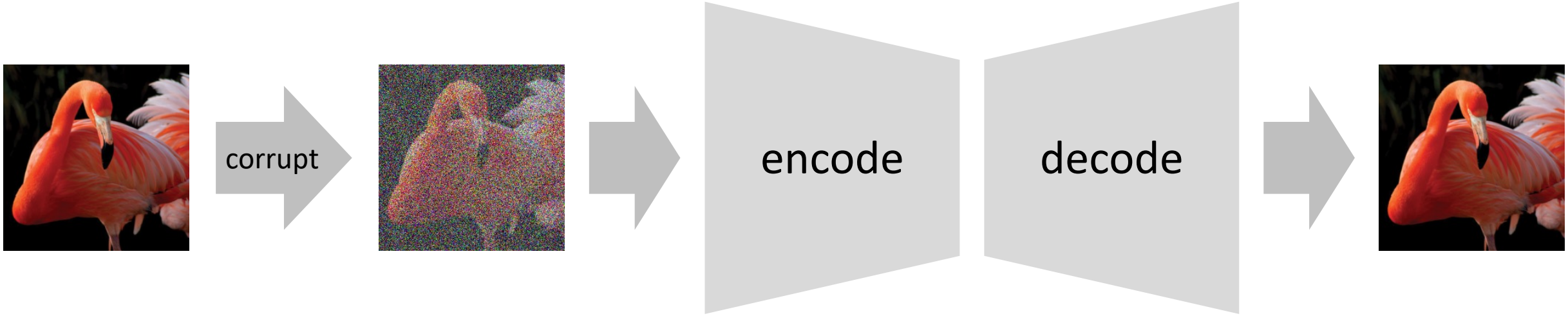
- **encoder** & **decoder**: deep neural nets
- optimized by BackProp
- a lower-dimensional latent as a "bottleneck"
- reconstruct the input

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

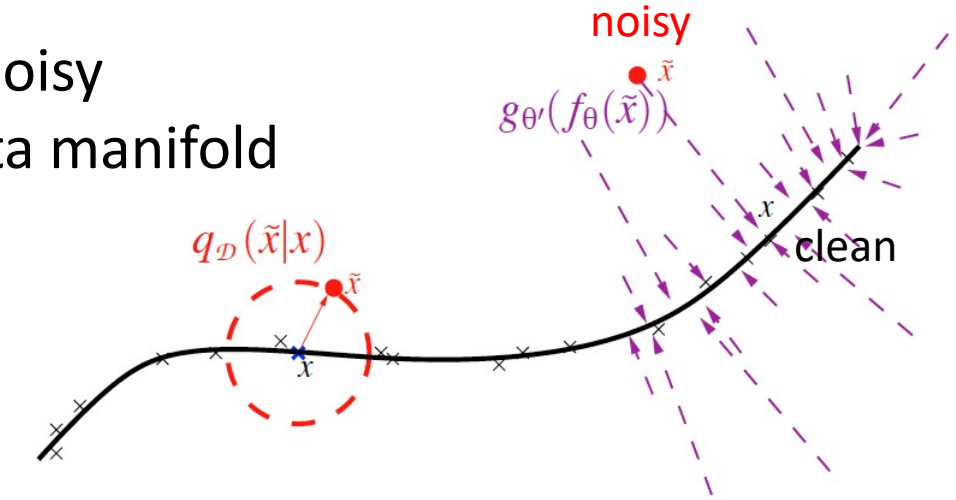


Hinton and Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks", Science 2006

Denoising Autoencoder (DAE) [2010]



learning by projecting noisy
samples back to the data manifold



Vincent, et al., “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion”, JMLR, 2010

Masked Image Modeling [2016]

Context Encoders

- Autoencoding with masking
- in pixel space
- using ConvNet

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

Masked Language Modeling [2018]

BERT (Bidirectional Encoder Representations from Transformers)

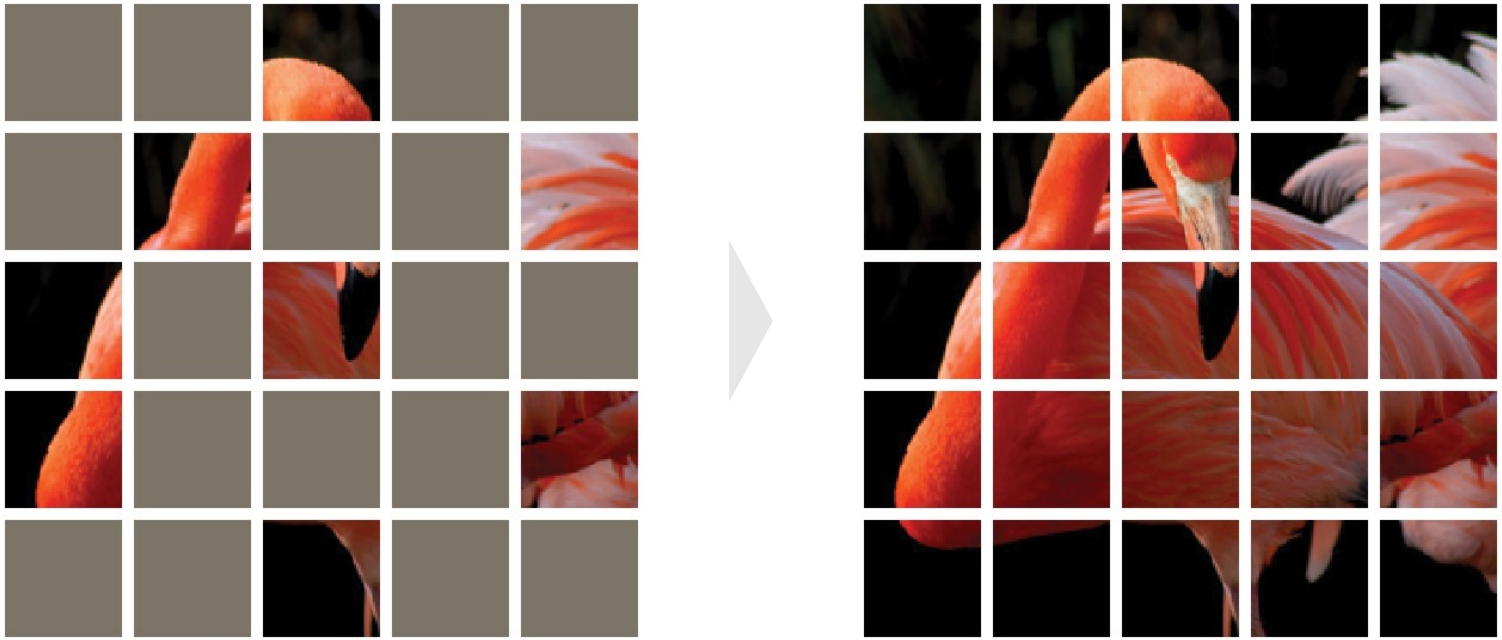
- Autoencoding with masking
- in token space
- using Transformers

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

Masked Image Modeling [2022]

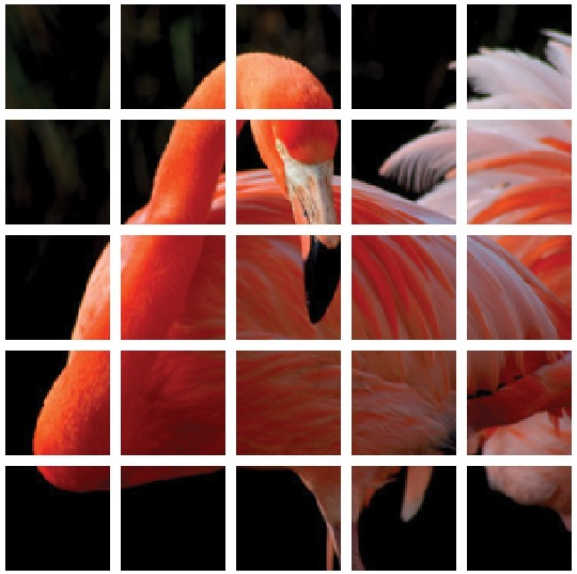
MAE (Masked Autoencoder)

- Autoencoding with masking
- in token/patch space
- using Transformers



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

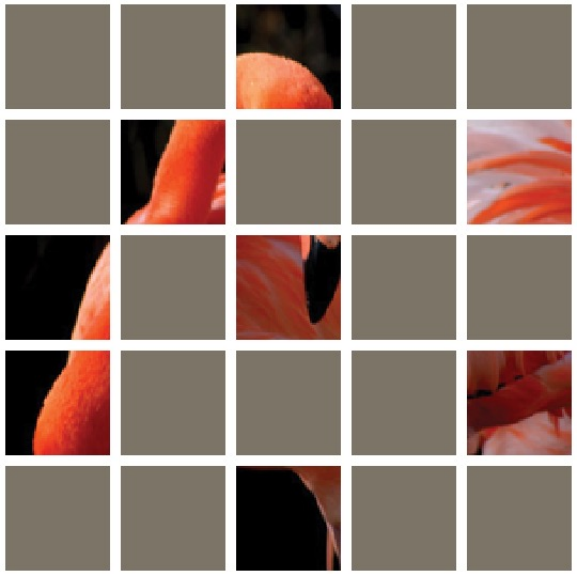
Masked Autoencoder (MAE)



patches as visual tokens
(Vision Transformer)

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

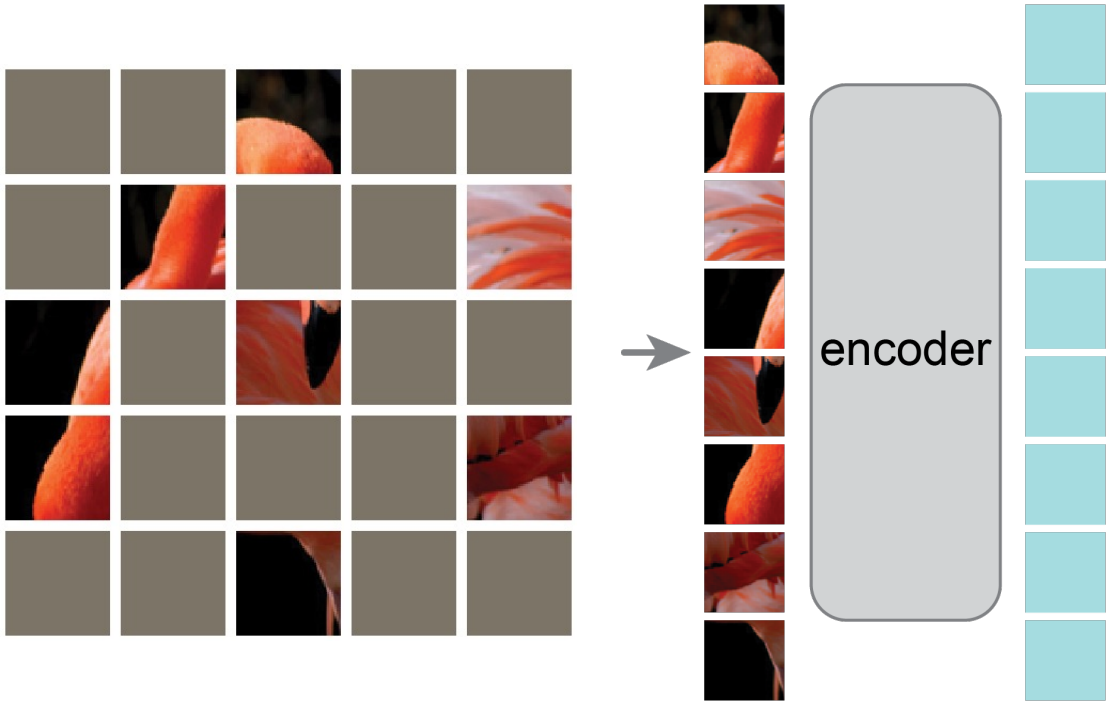
Masked Autoencoder (MAE)



random masking

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

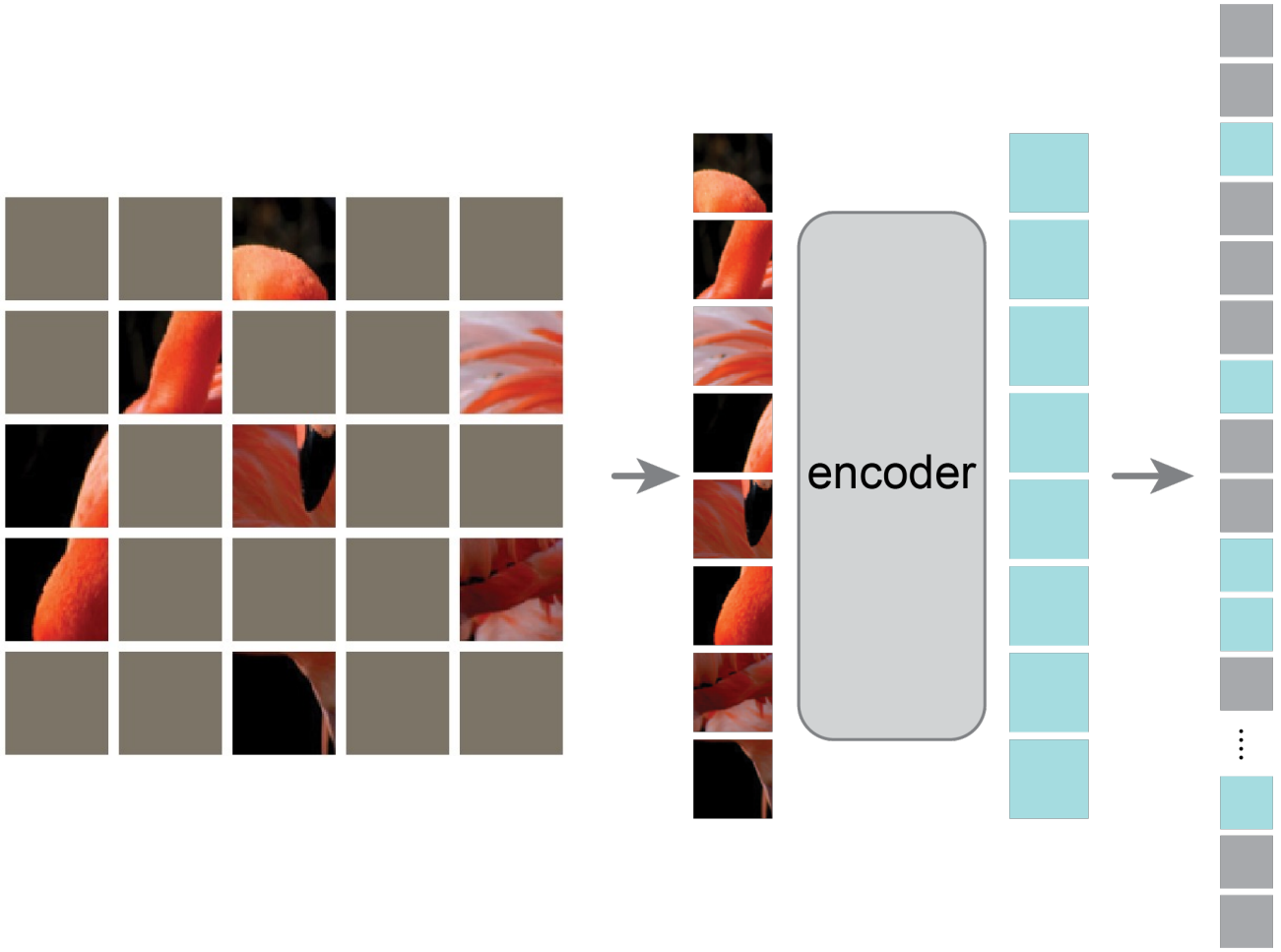
Masked Autoencoder (MAE)



encode visible patches
w/ Transformer

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

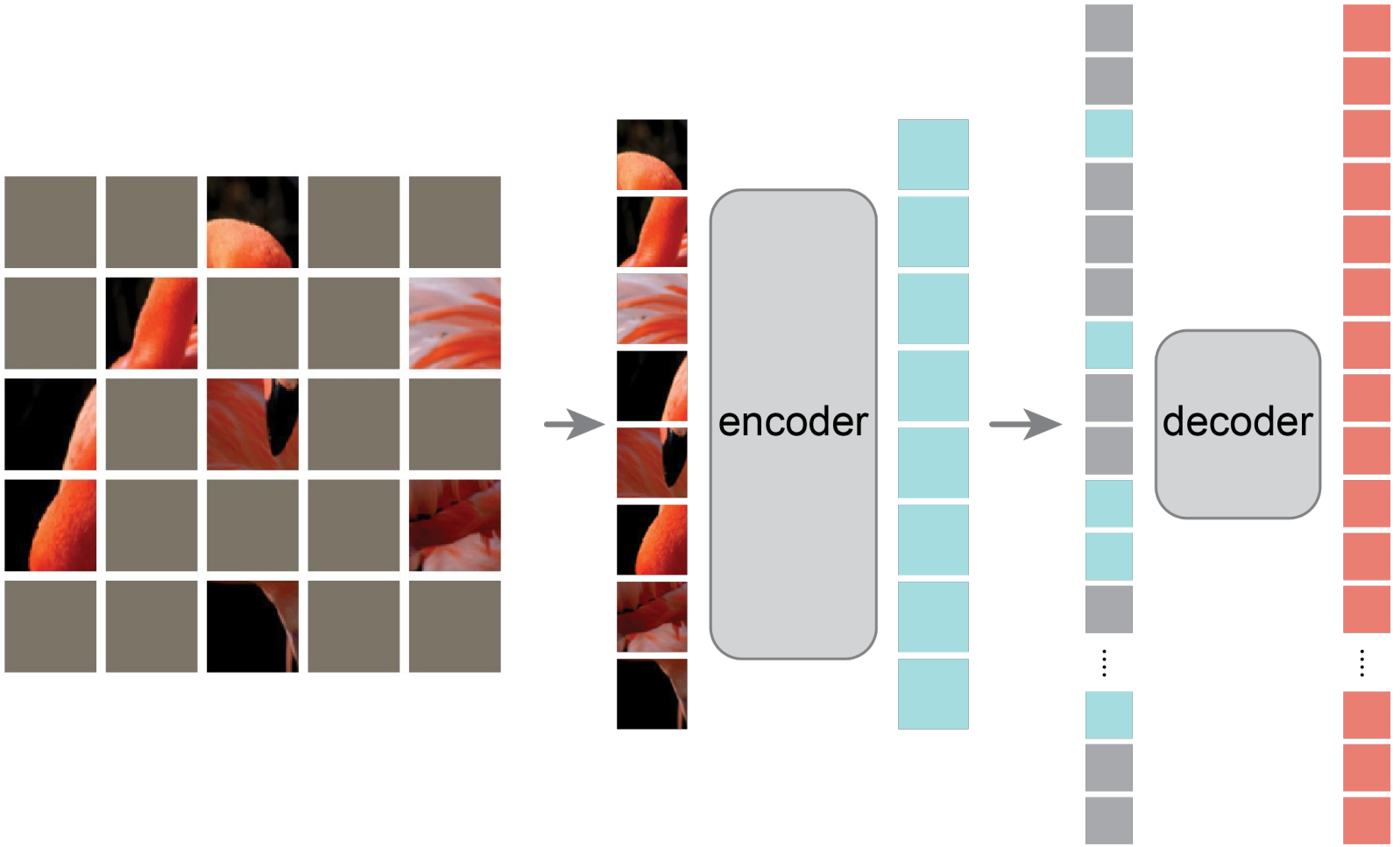
Masked Autoencoder (MAE)



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

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

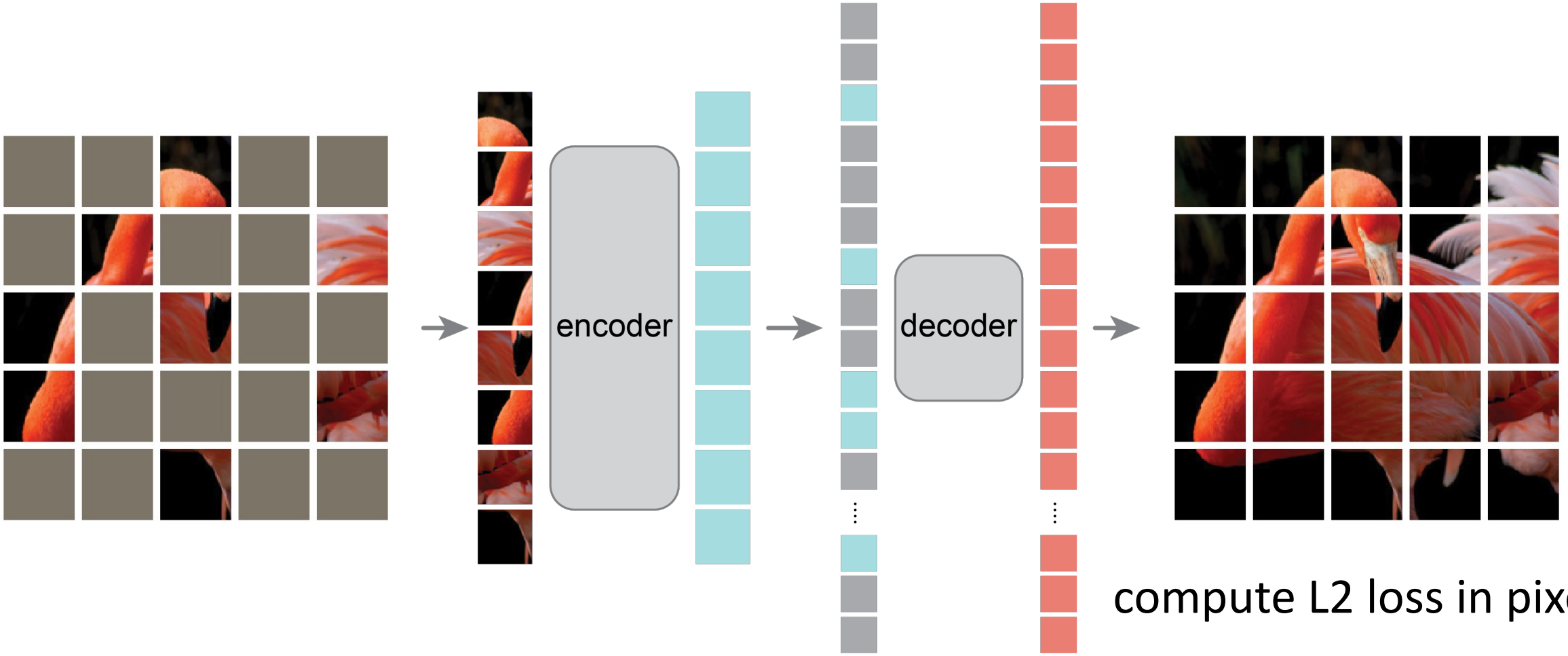
Masked Autoencoder (MAE)



predict the unknown

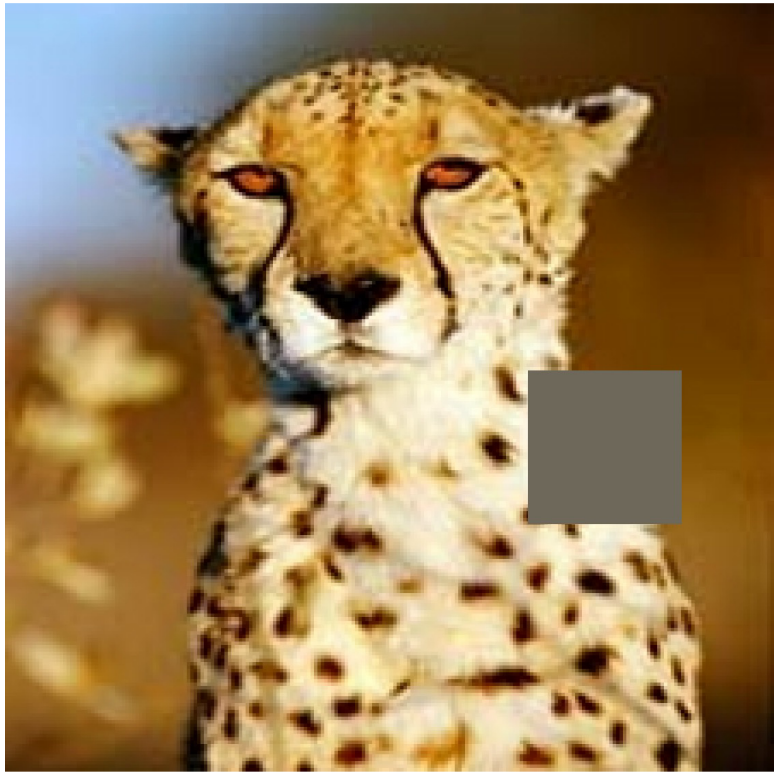
He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

Masked Autoencoder (MAE)

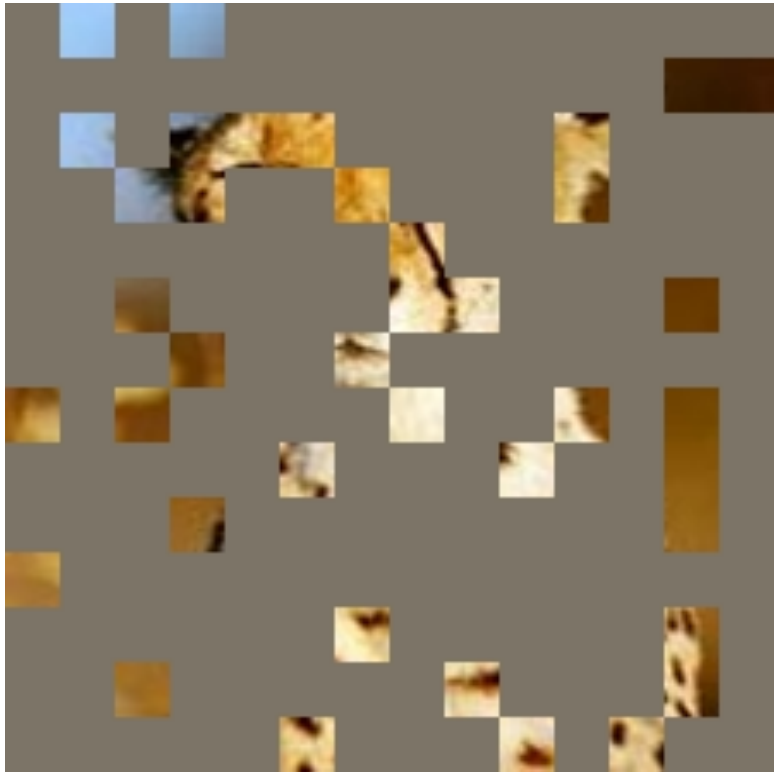


He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

How to encourage learning good representations?



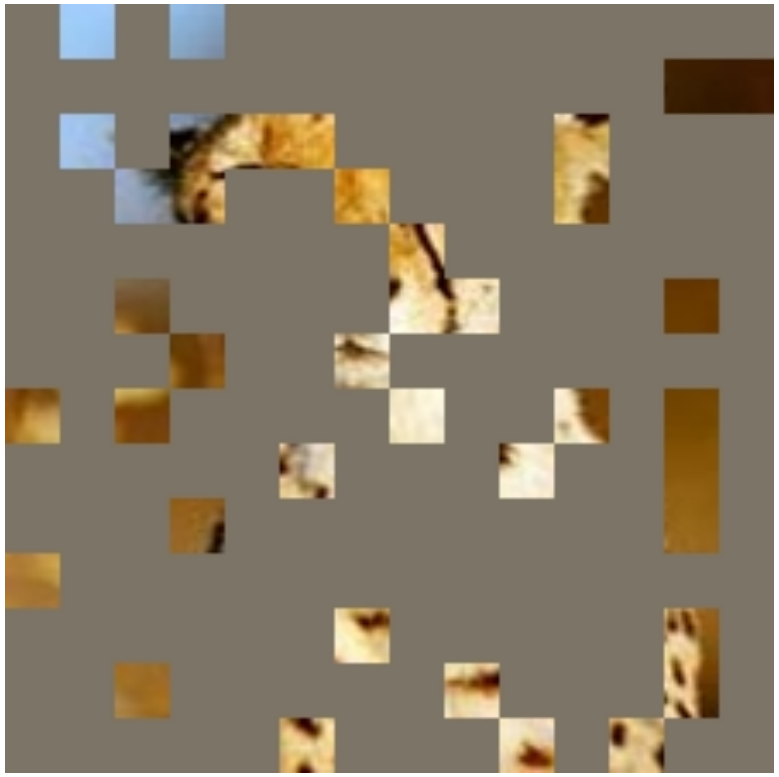
- predicting a small portion may not require high-level understanding



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

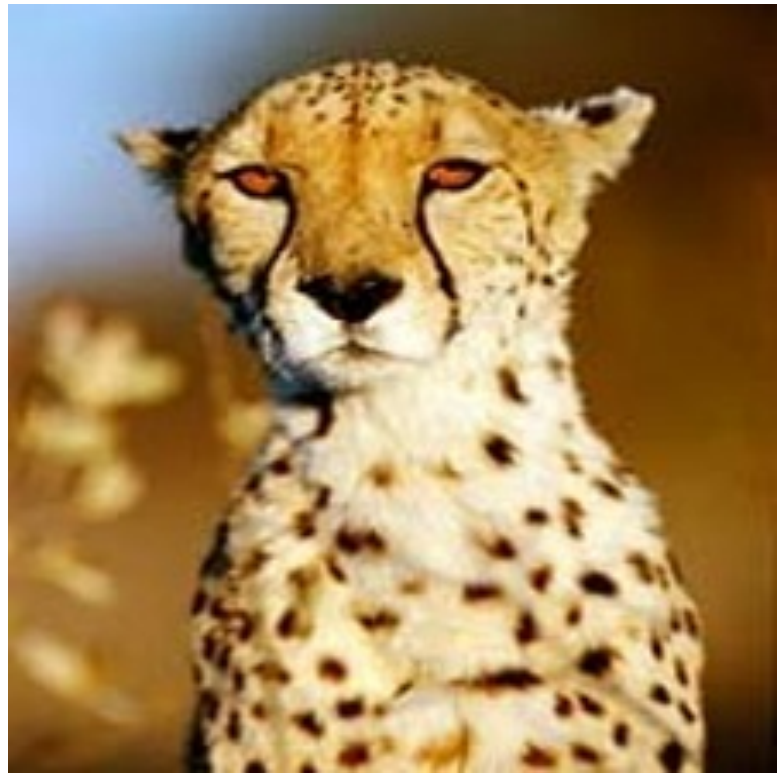
He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

How to encourage learning good representations?



input

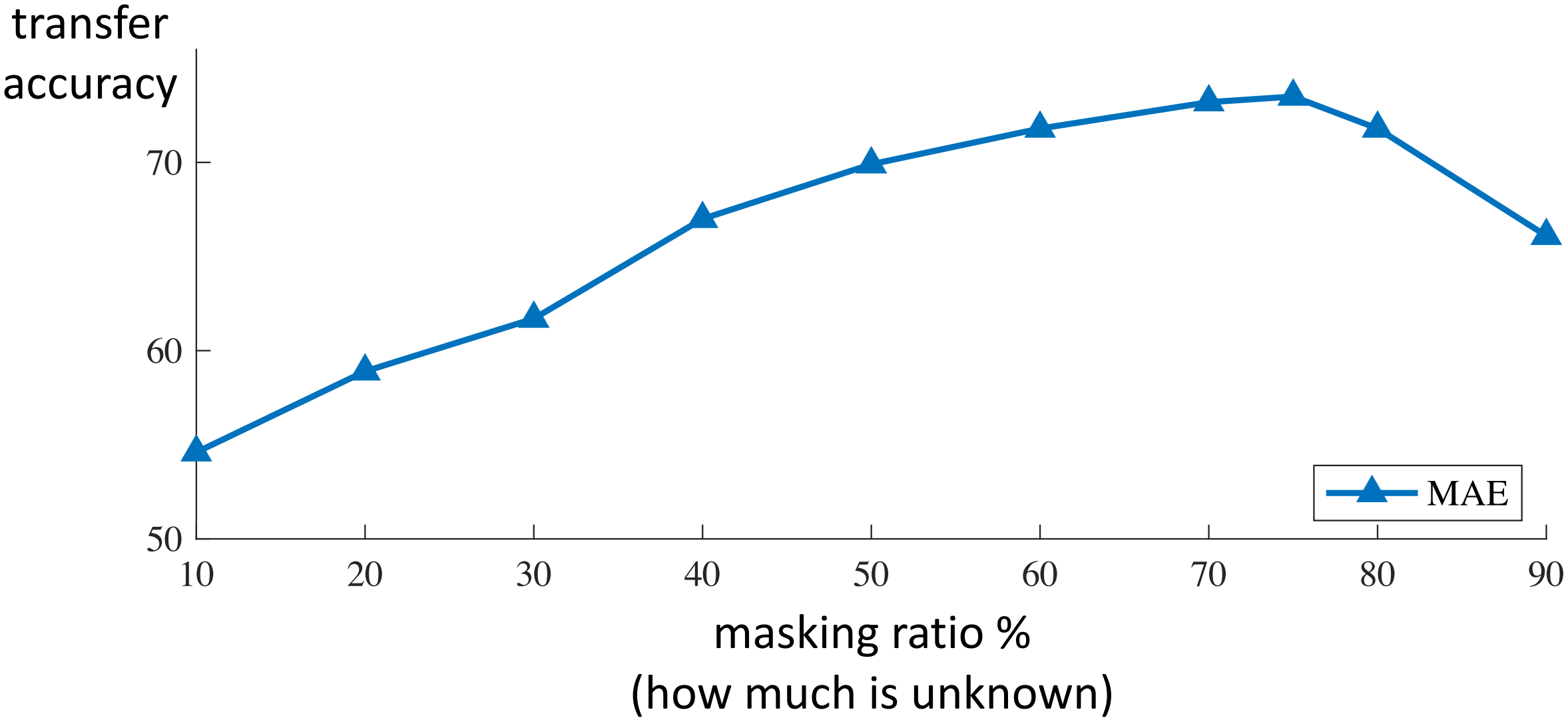
MAE prediction



original

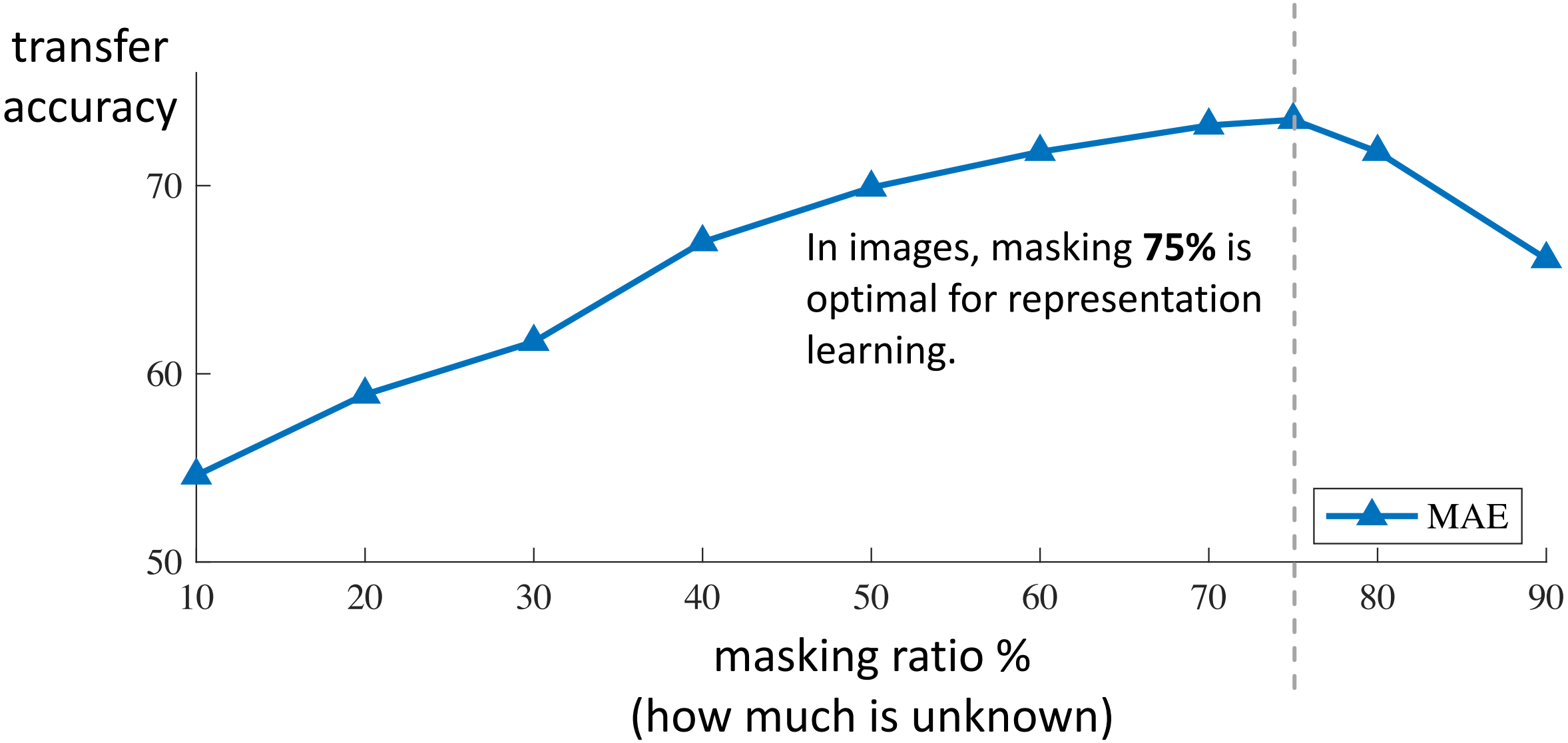
He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

How to encourage learning good representations?



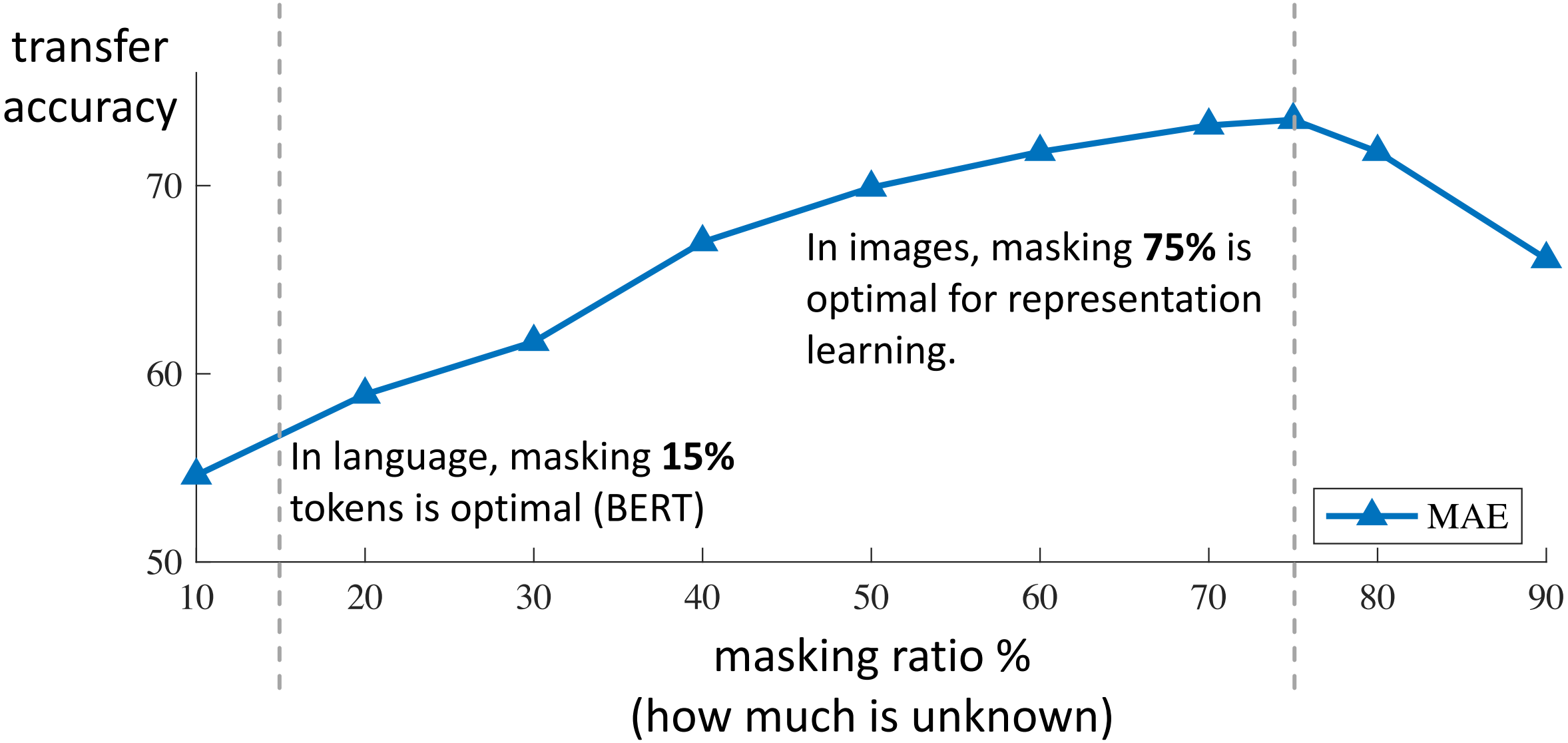
He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

How to encourage learning good representations?



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

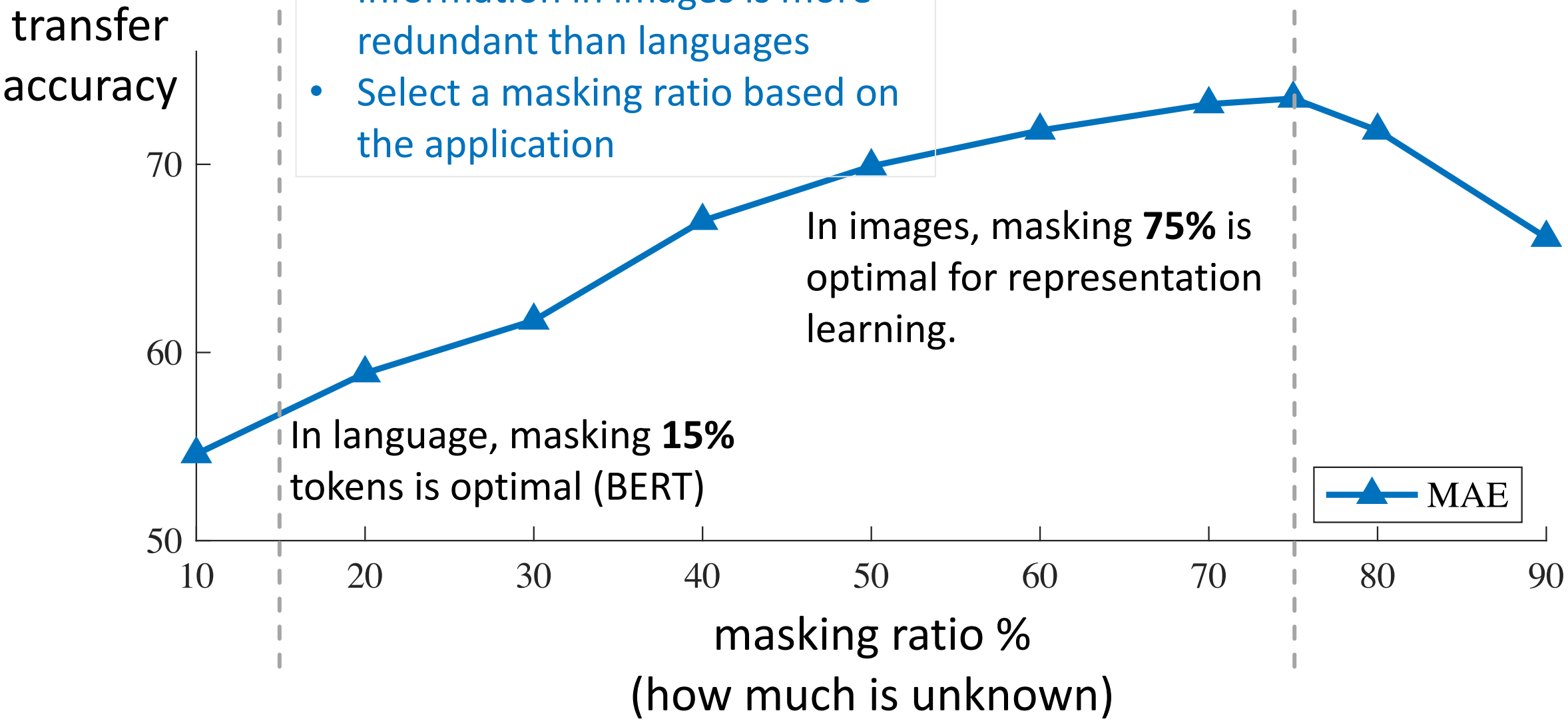
How to encourage learning good representations?



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

How to encourage learning good representations?

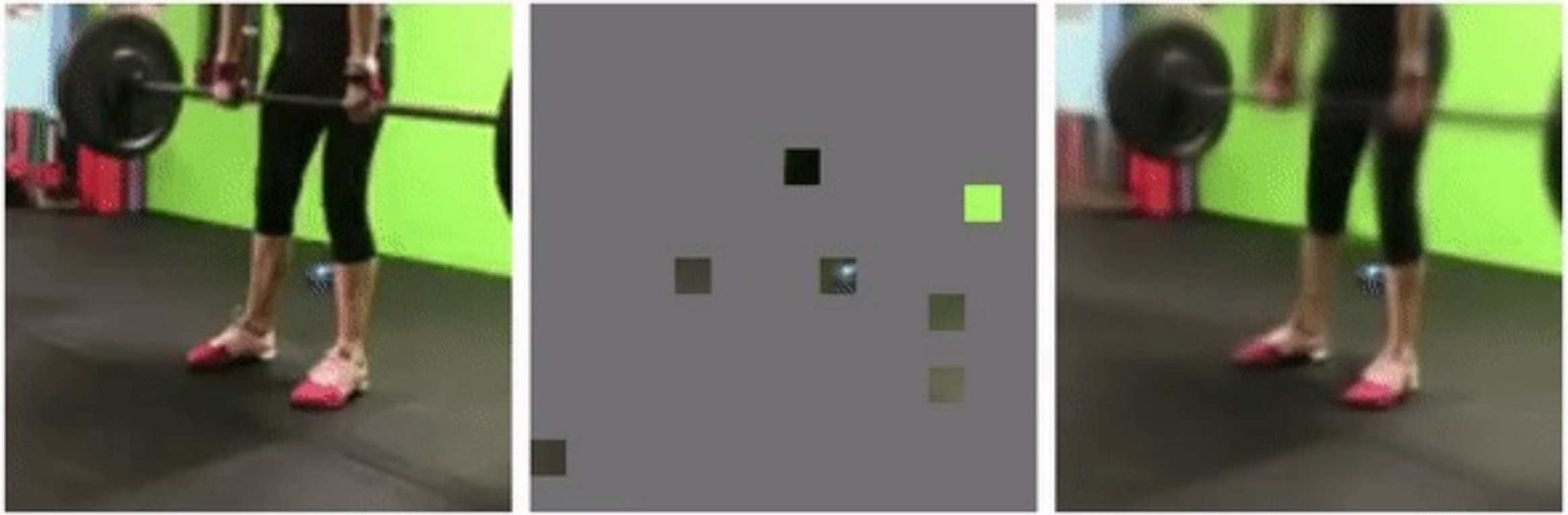
- Information in images is more redundant than languages
- Select a masking ratio based on the application



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

MAE on Videos

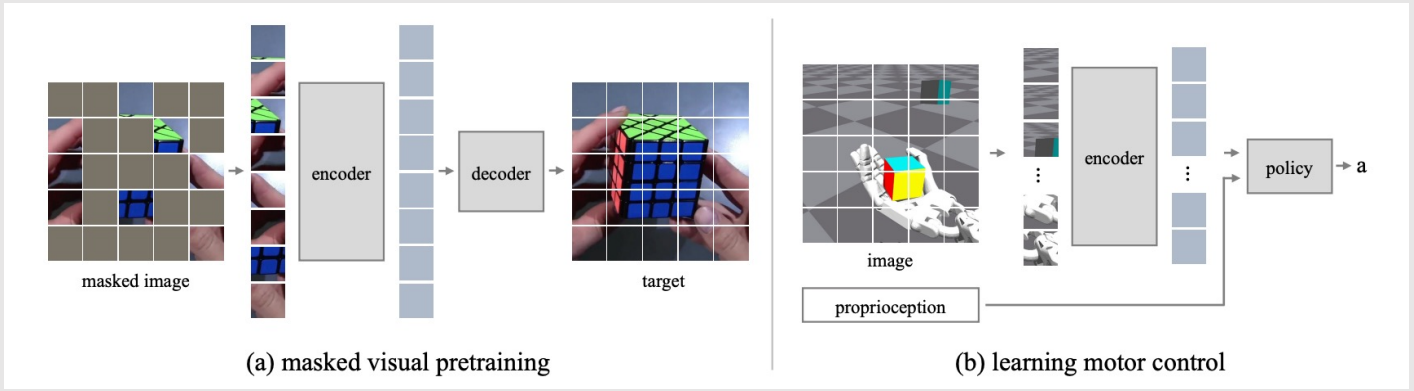
- 95% masking ratio
- videos are more redundant than images

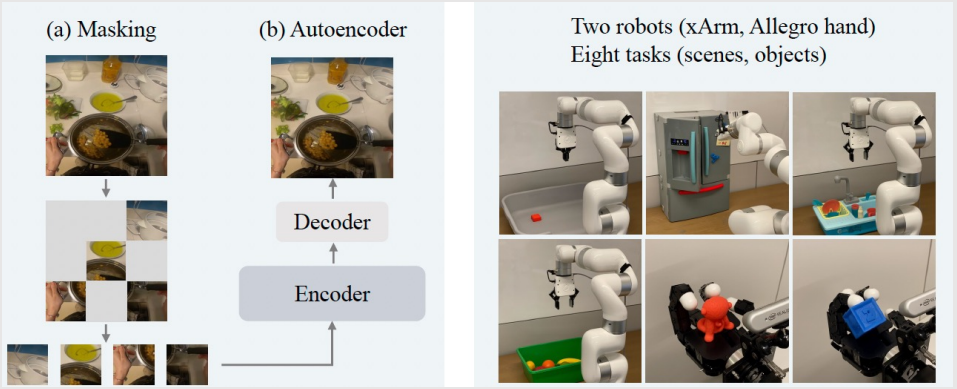


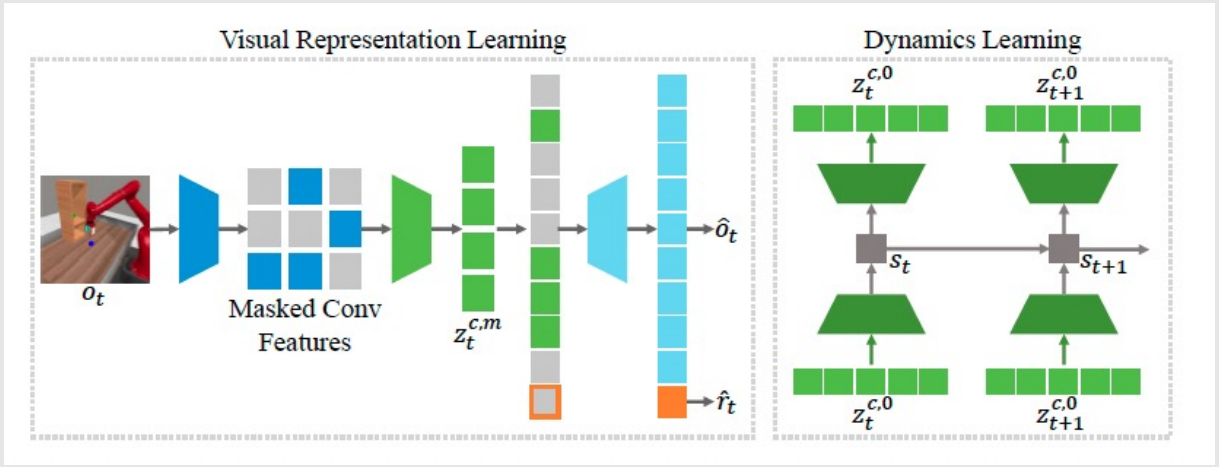
original | input | MAE prediction

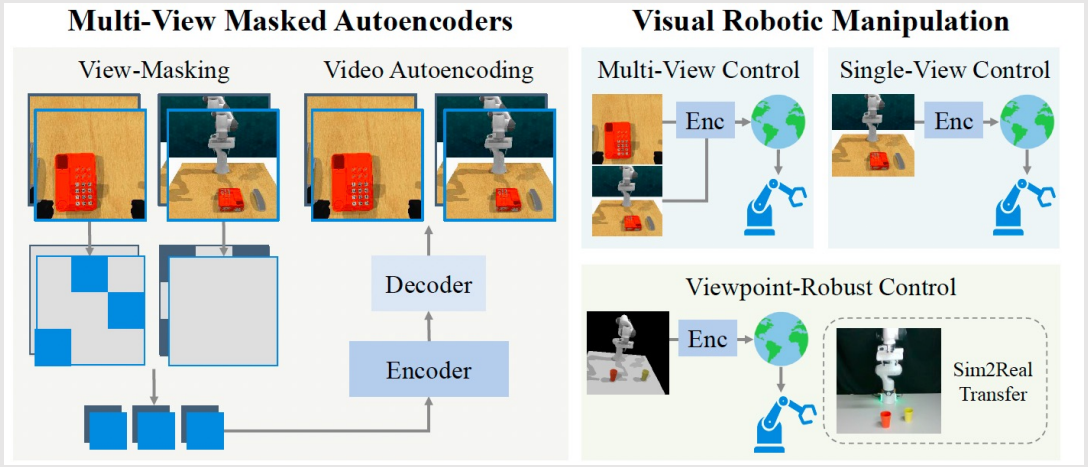
Feichtenhofer, et al., “Masked Autoencoders As Spatiotemporal Learners”, NeurIPS 2022

Masked Autoencoder: Applications





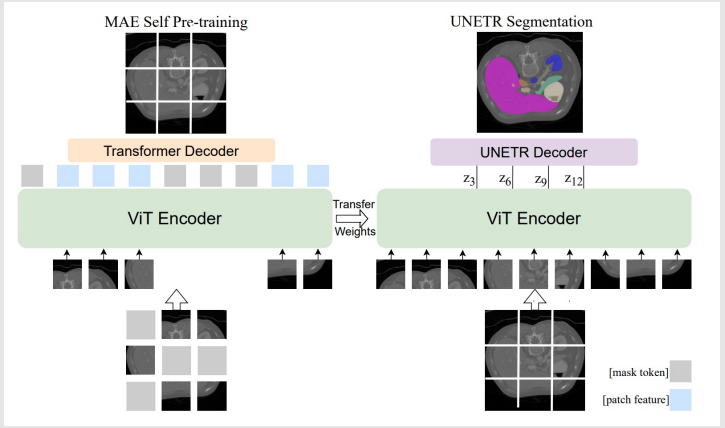


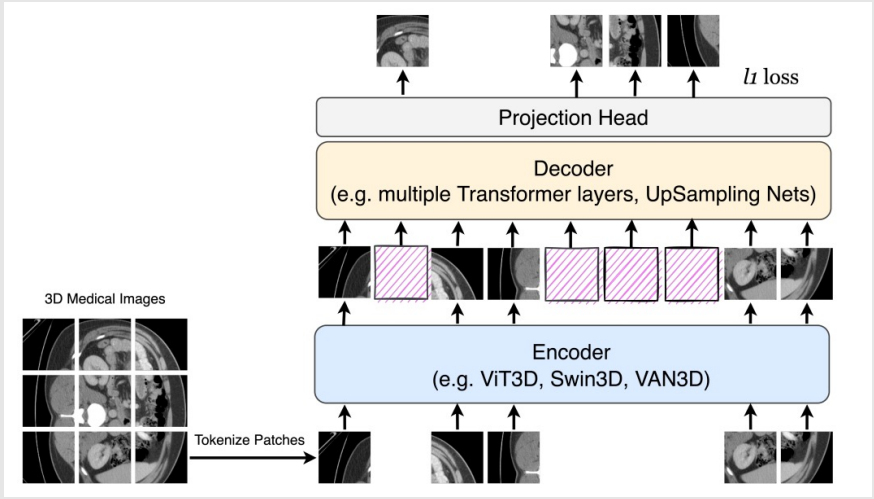


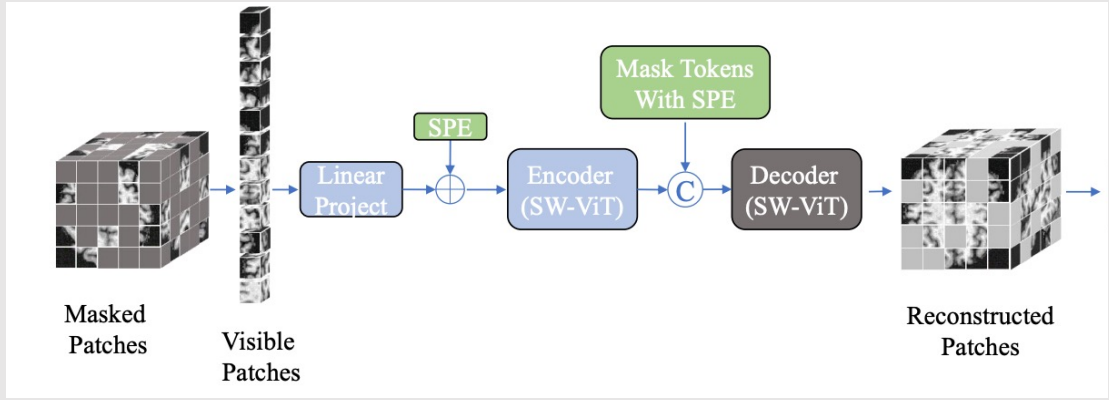
- **Robotics**

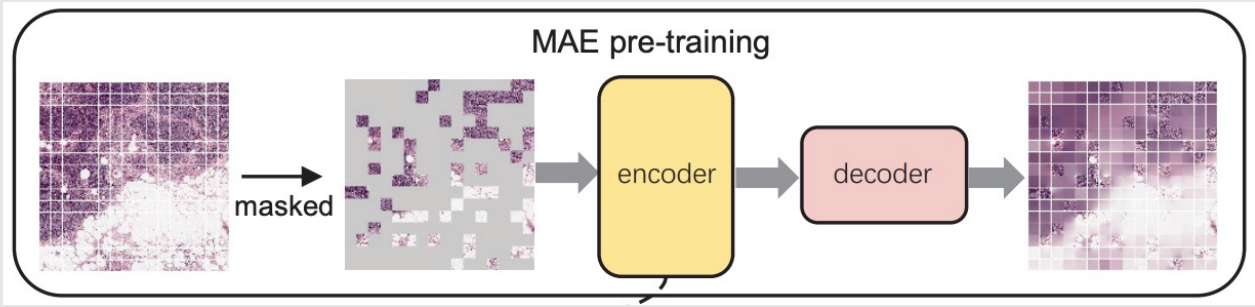
Xiao+, ’22; Radosavovic+, ’22; Seo+, ’22; Seo+, ’23

Masked Autoencoder: Applications





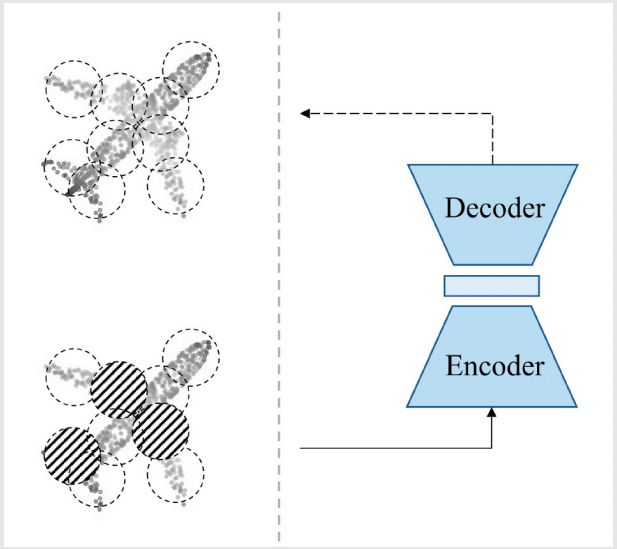


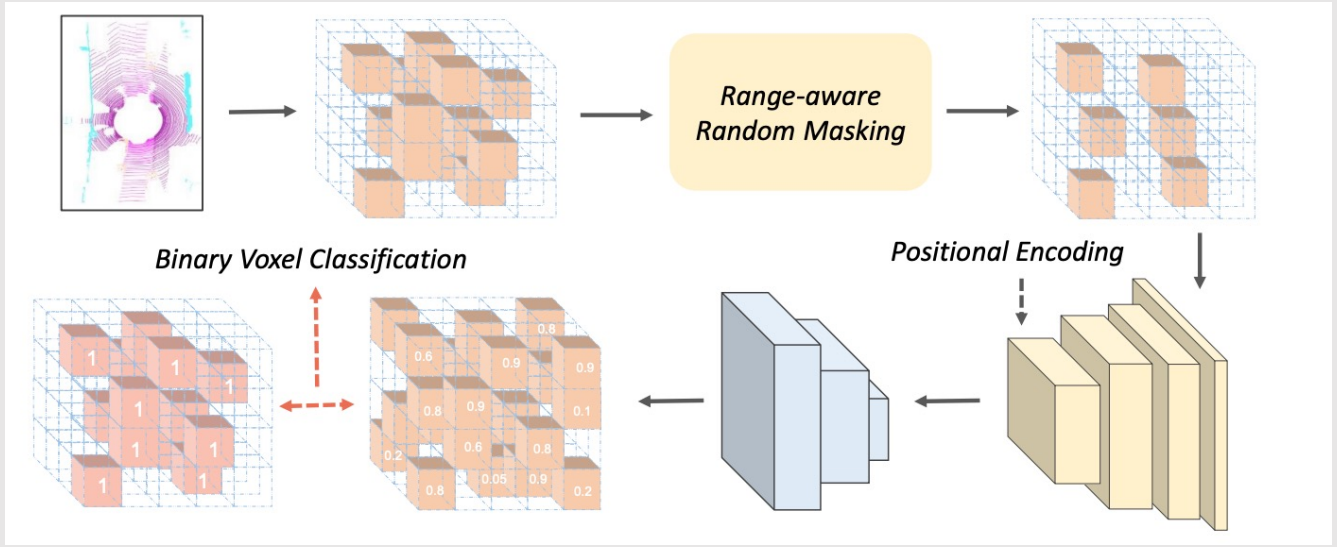


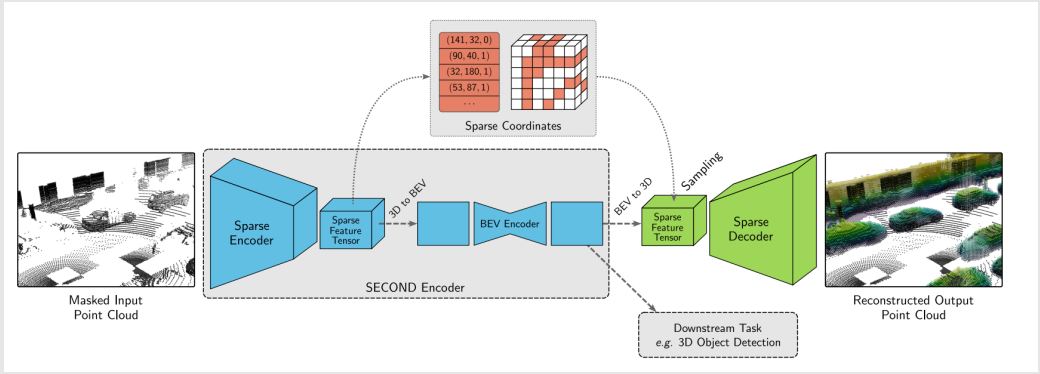
- **Medical Imaging**

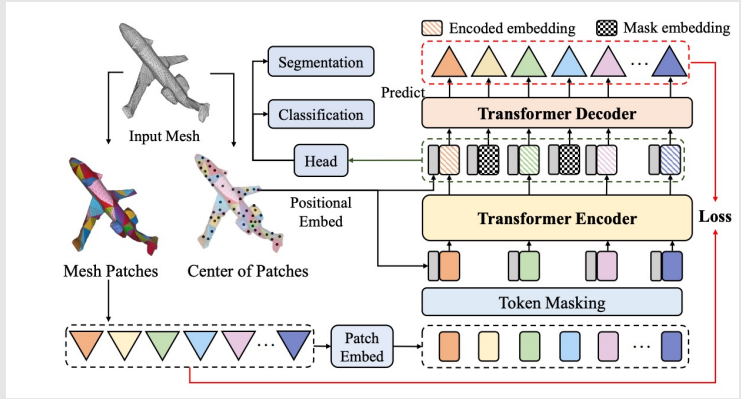
Zhou+,'22; Chen+,'22; Huang+,'22; An+,'22

Masked Autoencoder: Applications





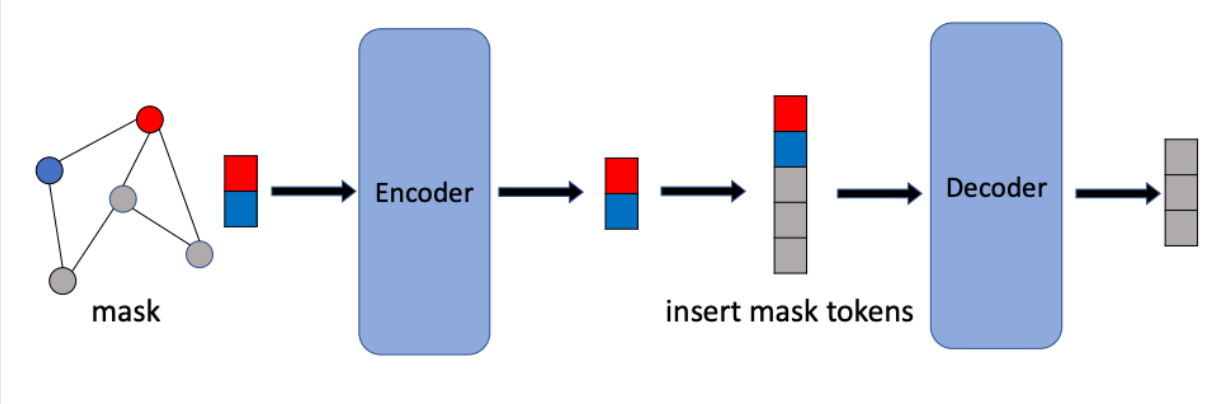


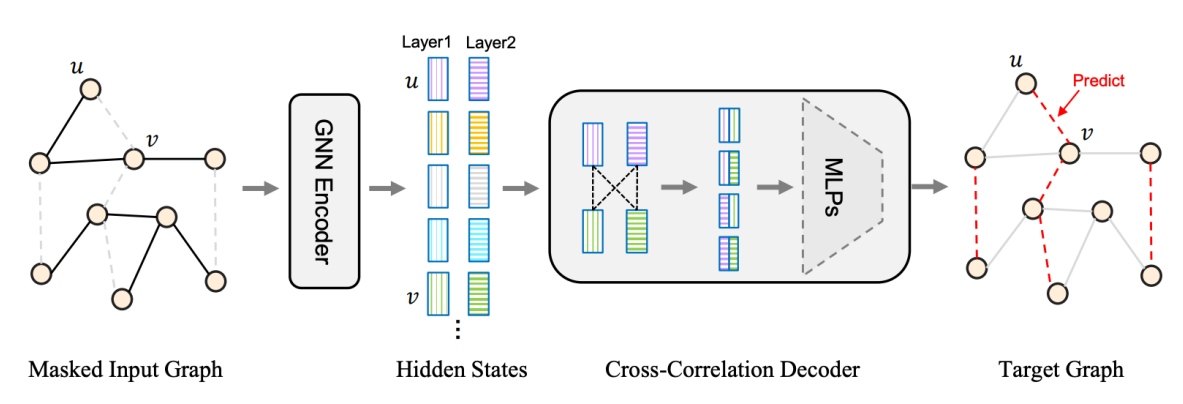


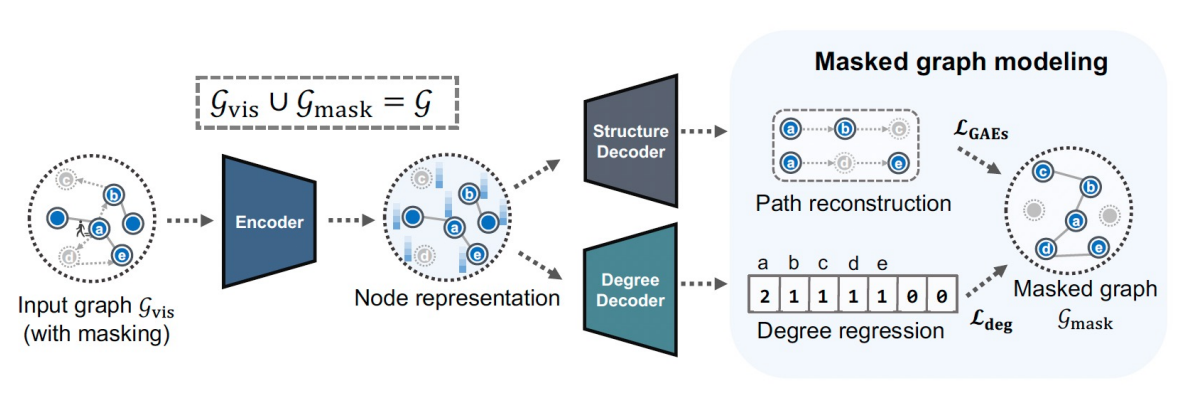
- **3D Geometry**

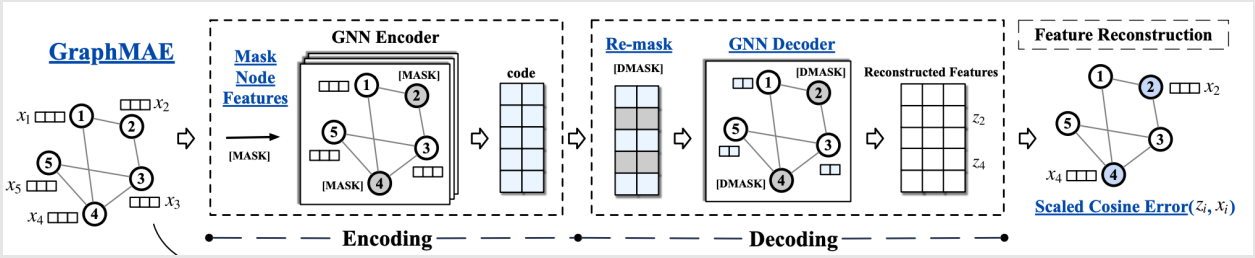
Pang+, ’22; Liang+, ’22; Min+, ’22; Krispel+, ’22

Masked Autoencoder: Applications





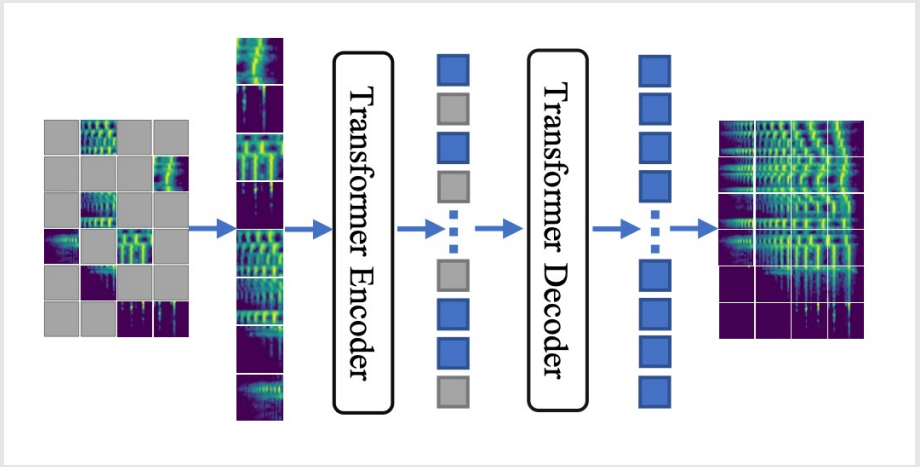


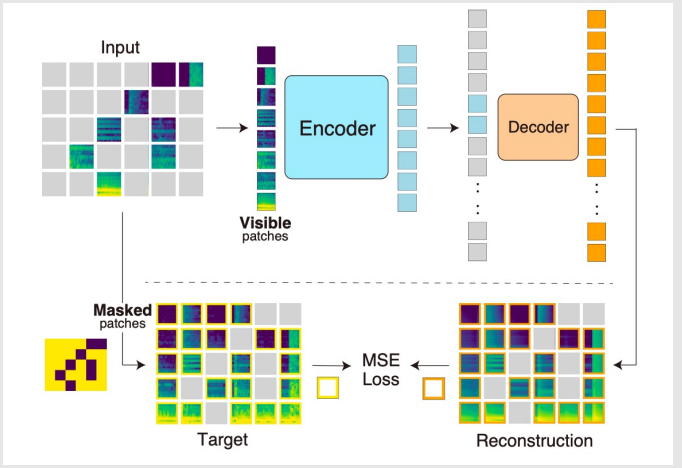


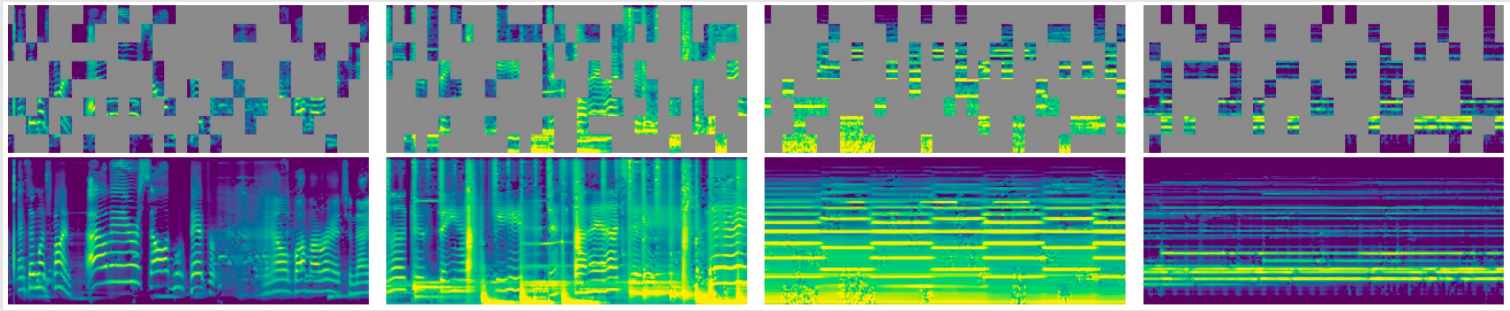
- **Graphs**

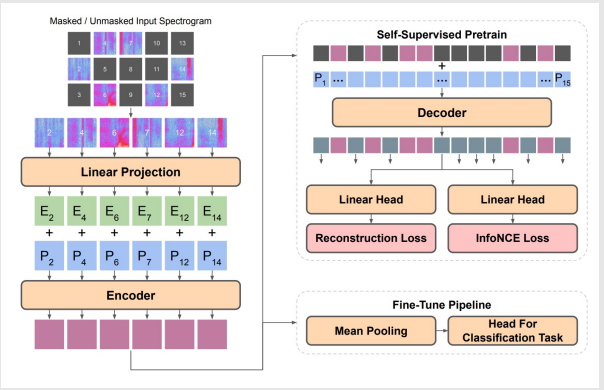
Tan+, '22; Zhang+, '22; Hou+, '22; Li+, '22

Masked Autoencoder: Applications









- **Audio**

Baade+, ’22; Chong+, ’22; Niizumi, ’22; Huang+, ’22

Summary

Reconstruction-based Representation Learning

- Autoencoding (k-means, PCAs, AE, DAE, BERT, MAE, ...)
- (+) simple and easy to use
- (+) require little domain-knowledge
- (-/+) objective encourages preserving all nuances