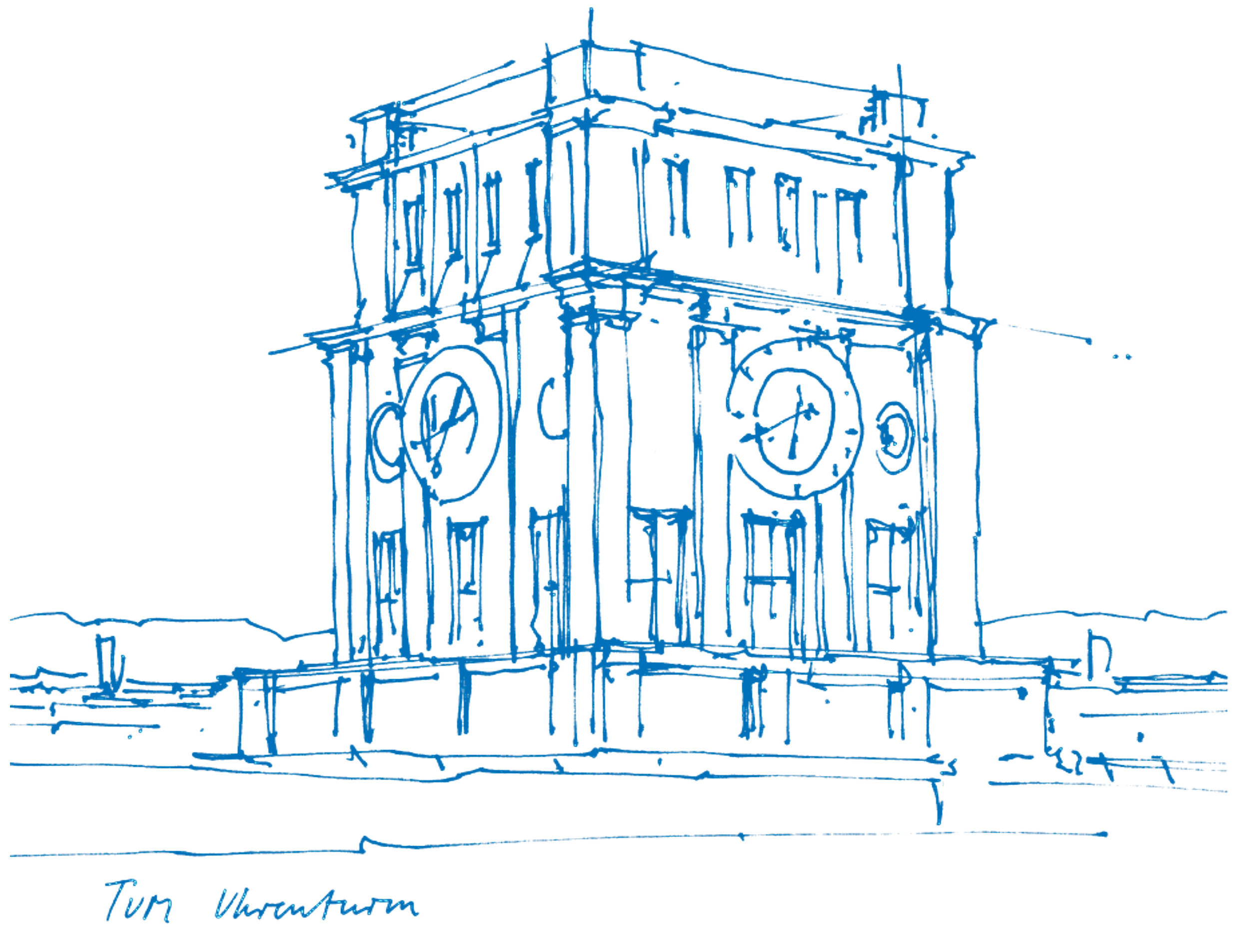
Computer Vision III:
Unsupervised learning

Nikita Araslanov
09.01.2024



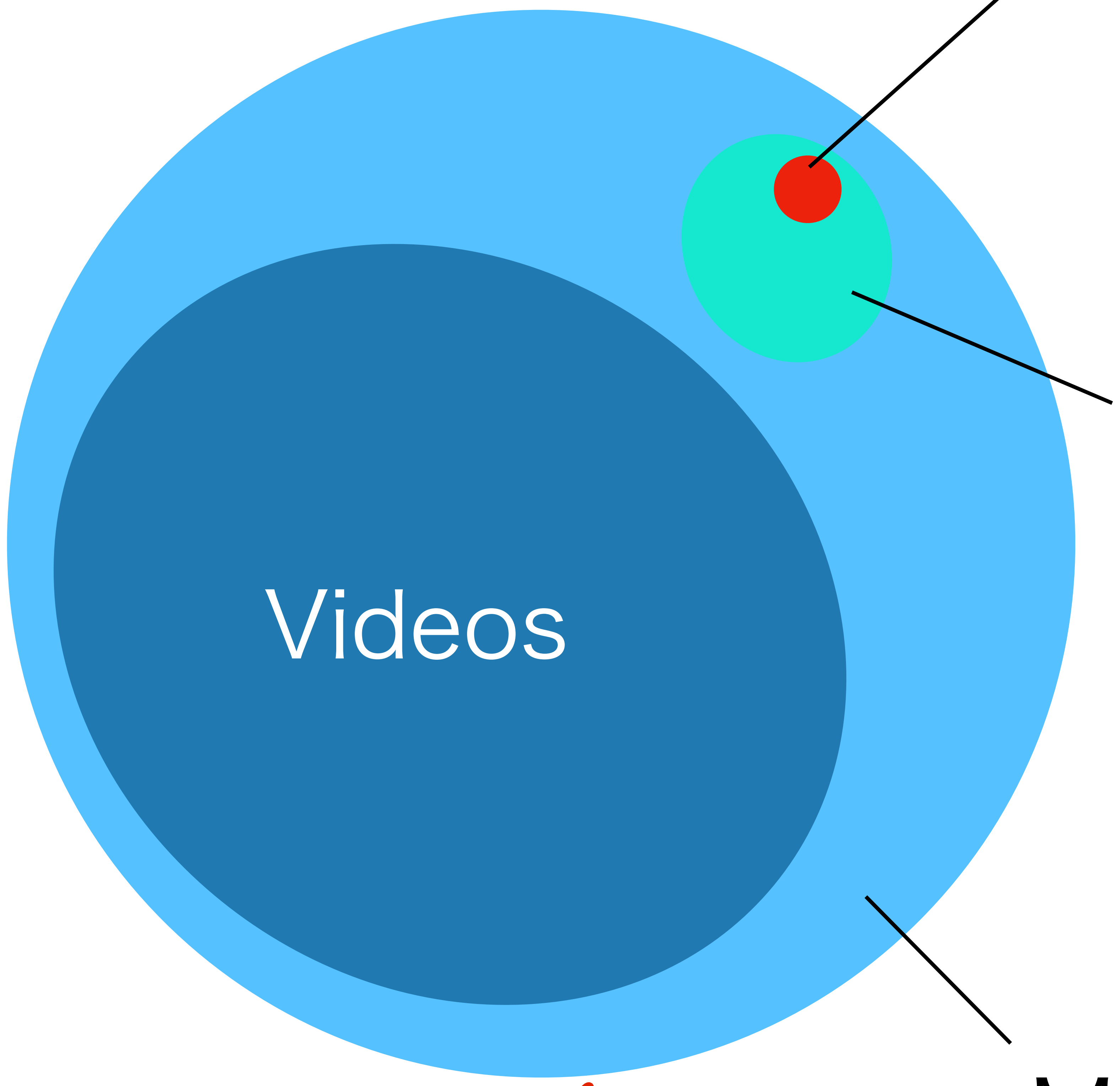
Course progress

1. Introduction
2. Object detection 1
3. Object detection 2
4. Single object tracking
5. Multiple object tracking
6. Semantic segmentation
7. Instance & panoptic segmentation
8. Video object segmentation
9. Transformers
10. **Unsupervised DST** Today
11. Semi-supervised DST (next week)

Limited supervision

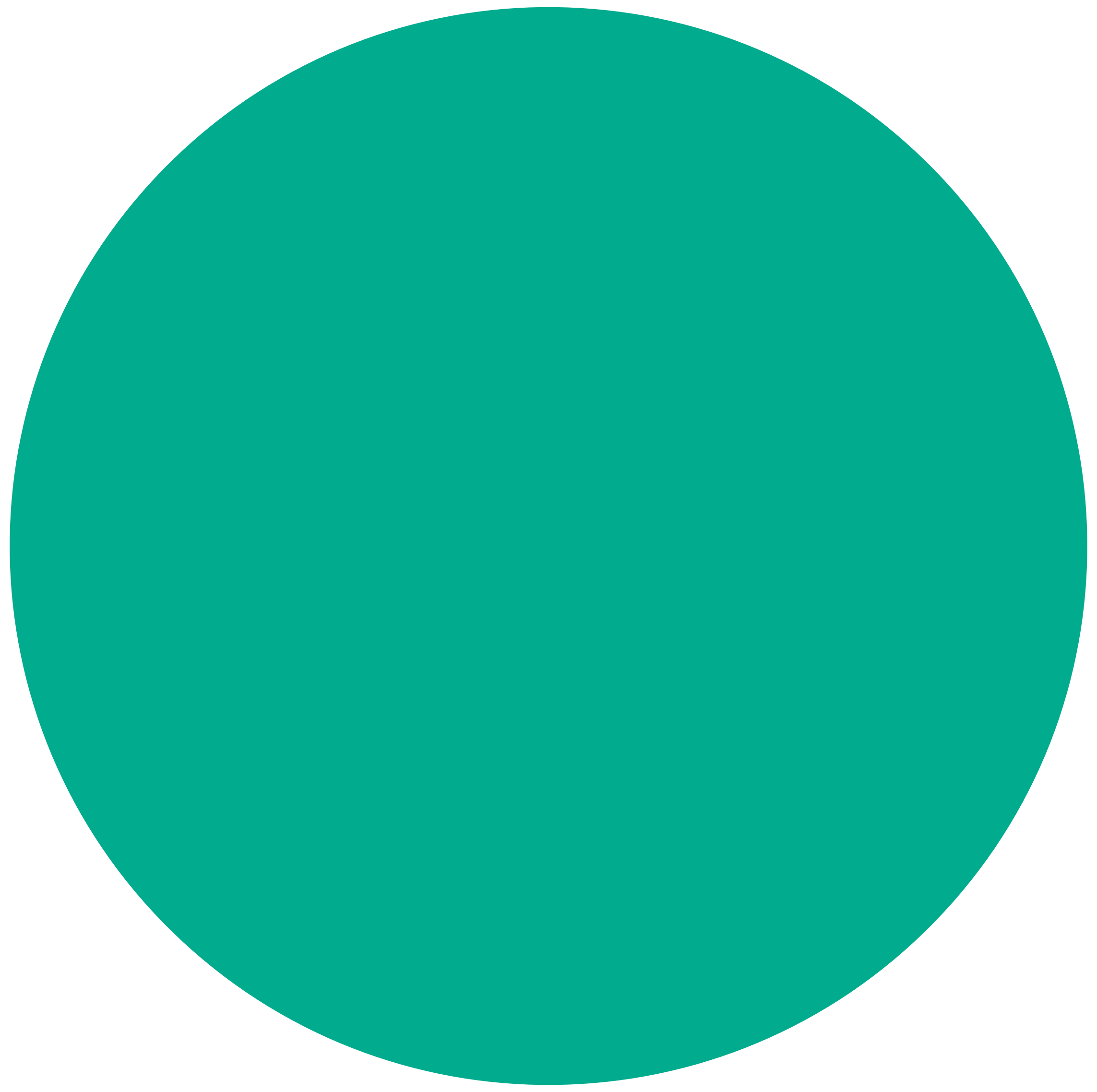
Real image data

Fine annotation labeled



Coarse annotation
(e.g. image-level labels)

Synthetic data

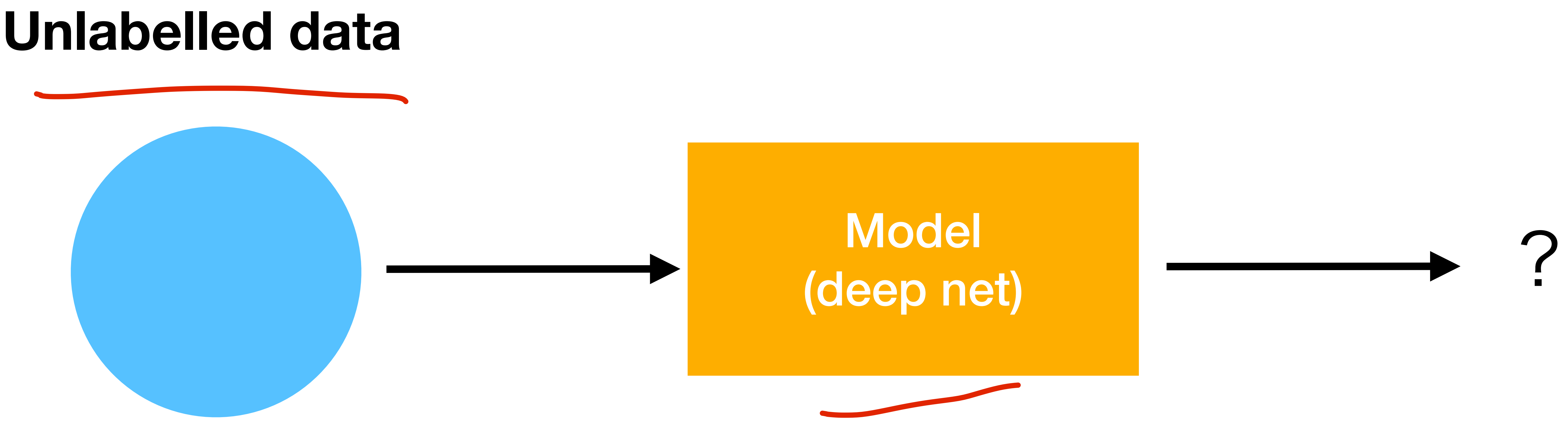


Goal: Make use of unlabelled data

Unsupervised learning

Learning without labels

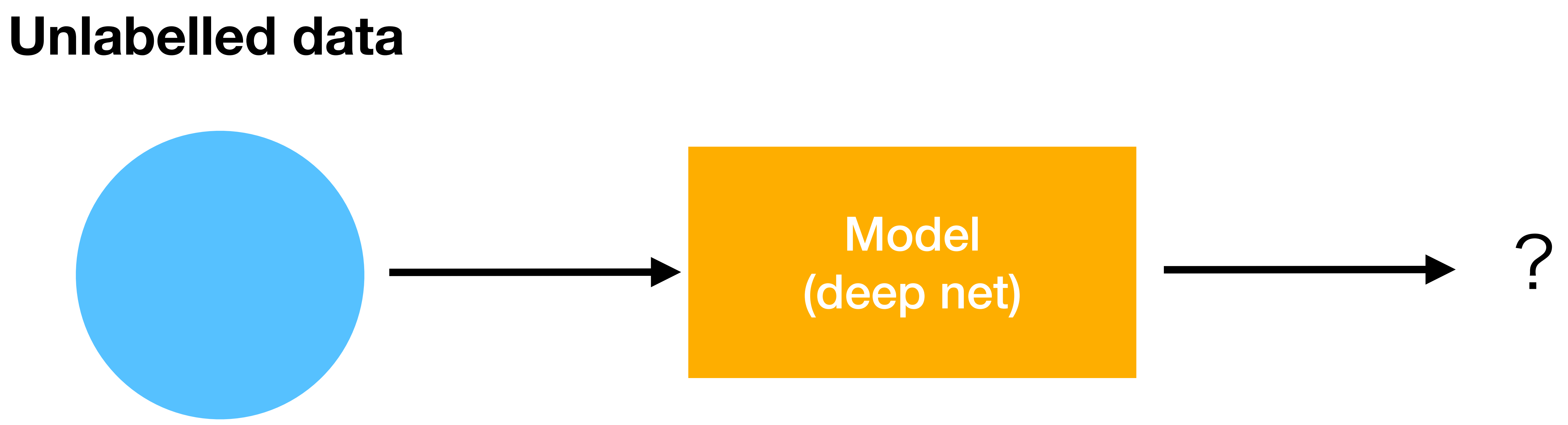
Suppose we do not have any labelled data:



Learning without labels

Suppose we do not have any labelled data:

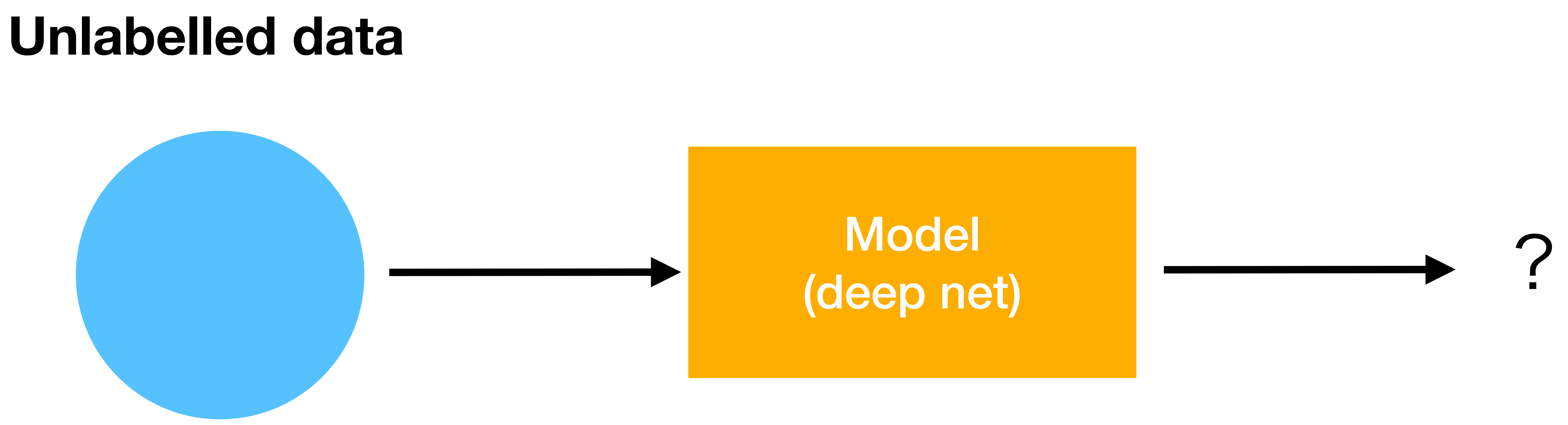
- Can we hope to learn anything useful from the unlabelled data?



Learning without labels

Suppose we do not have any labelled data:

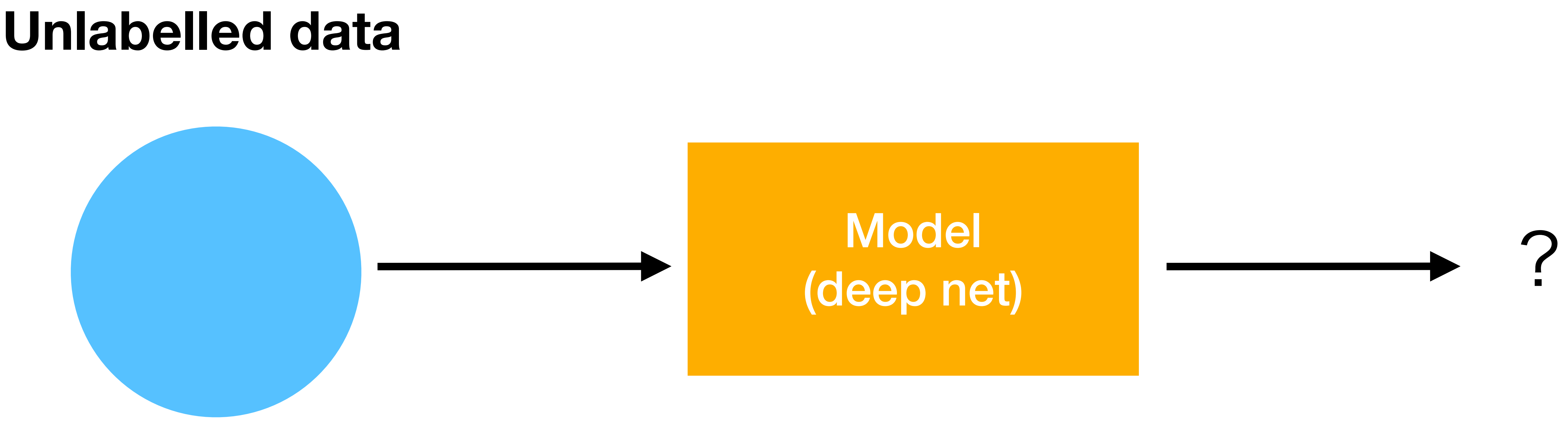
- Can we hope to learn anything useful from the unlabelled data?
- What is “useful” here?



Learning without labels

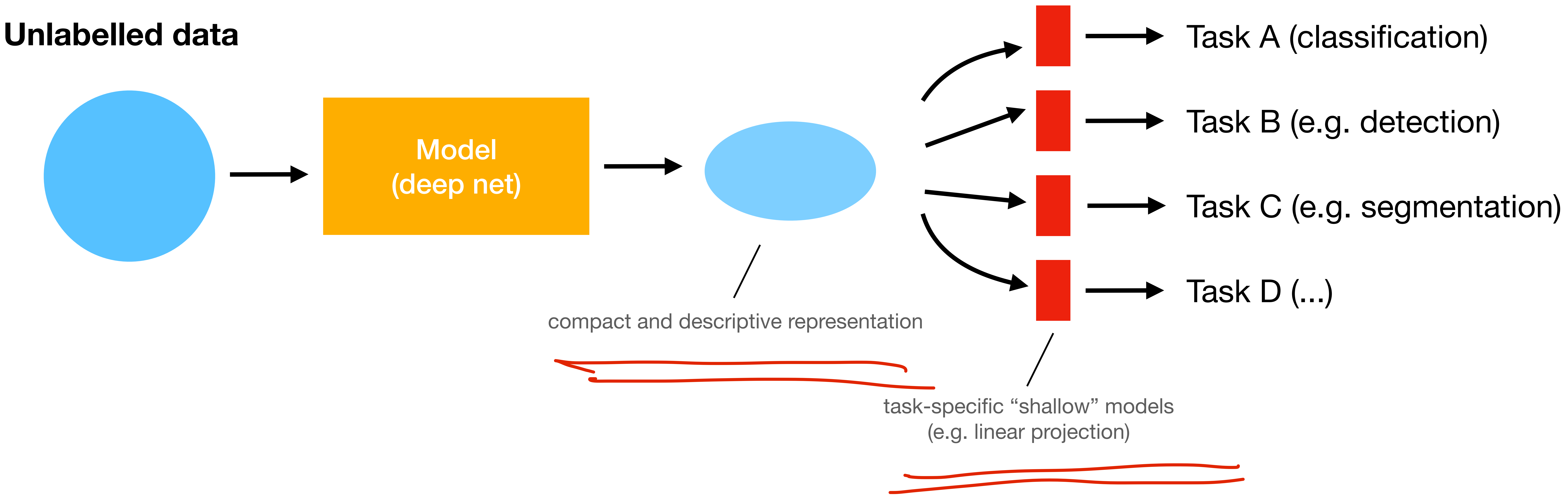
Suppose we do not have any labelled data:

- Can we hope to learn anything useful from the unlabelled data?
- What is “useful” here?
- Compact, yet descriptive representation.



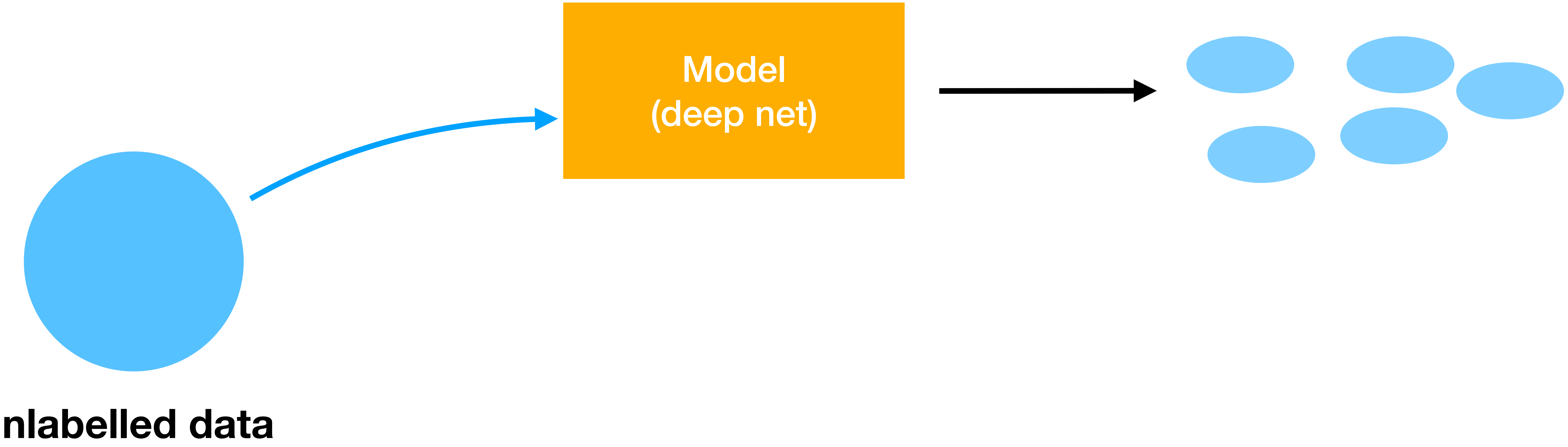
Learning without labels

Compact, yet descriptive representation:



Learning without labels

Idealised example – linearly separable embeddings (e.g. in classification):



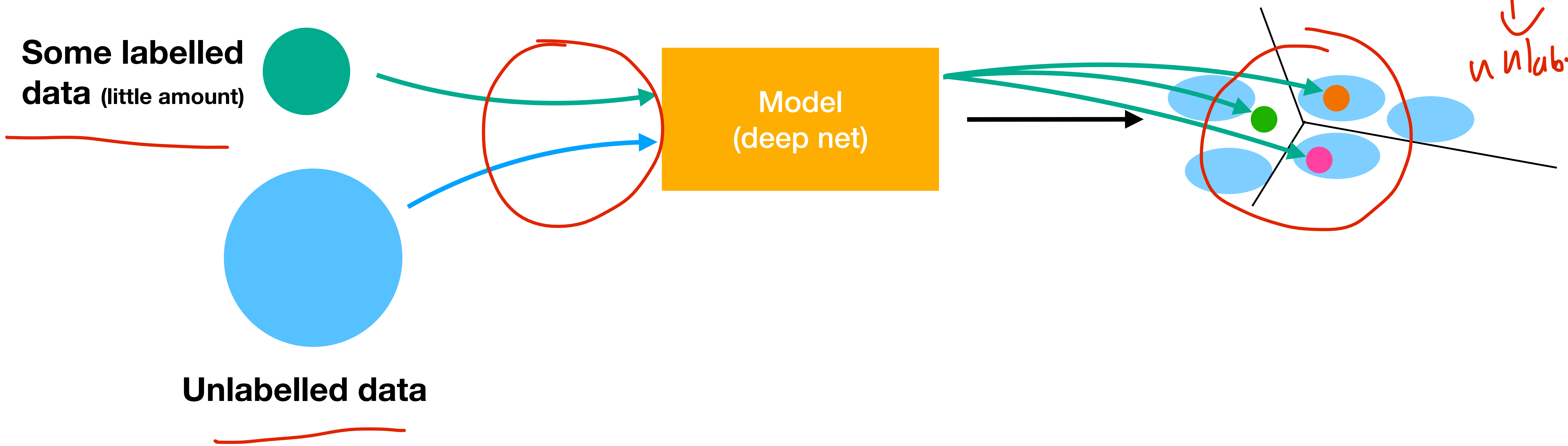
Learning without labels

Idealised example – linearly separable embeddings (e.g. in classification):

labeled → decision bound

↓

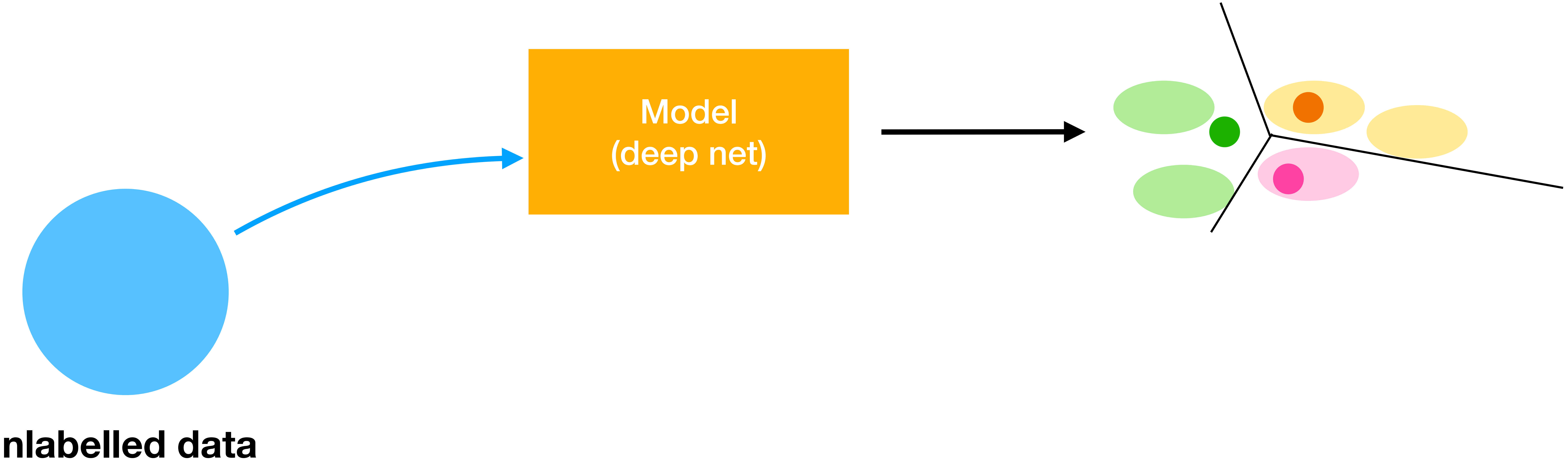
unlabeled



- Meaningful labels with a few labeled examples and a linear classifier.

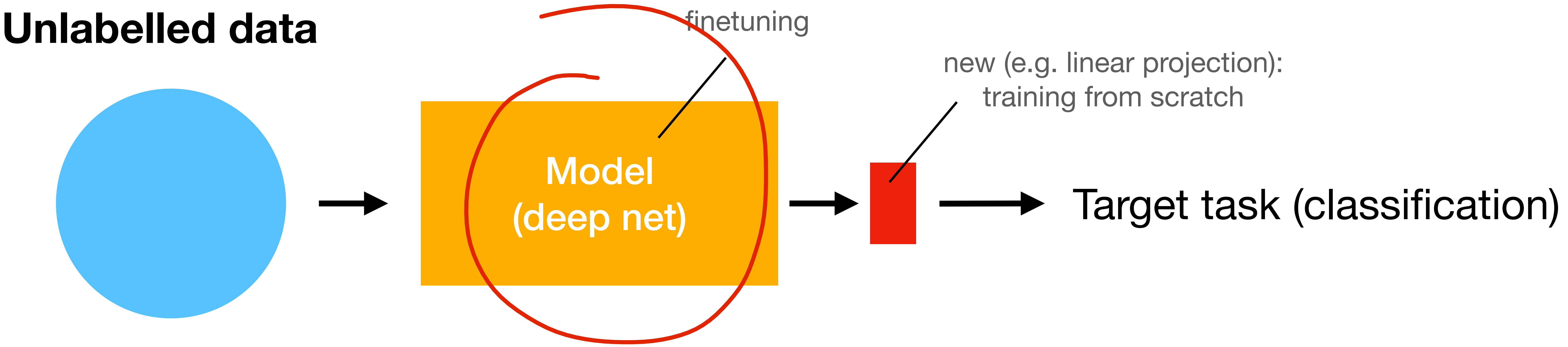
Learning without labels

Idealised example – linearly separable embeddings (e.g. in classification):



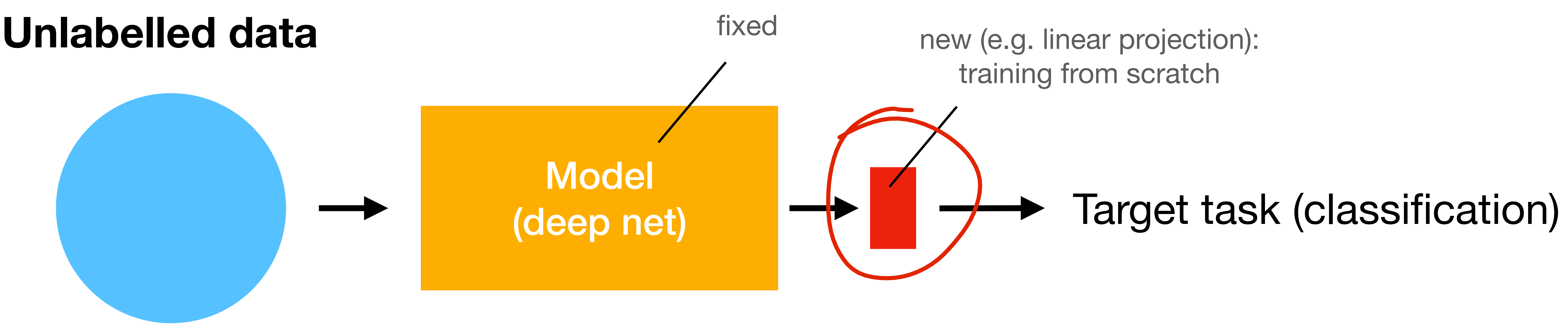
- Meaningful labels with a few labeled examples and a linear classifier.

Evaluating SSL models



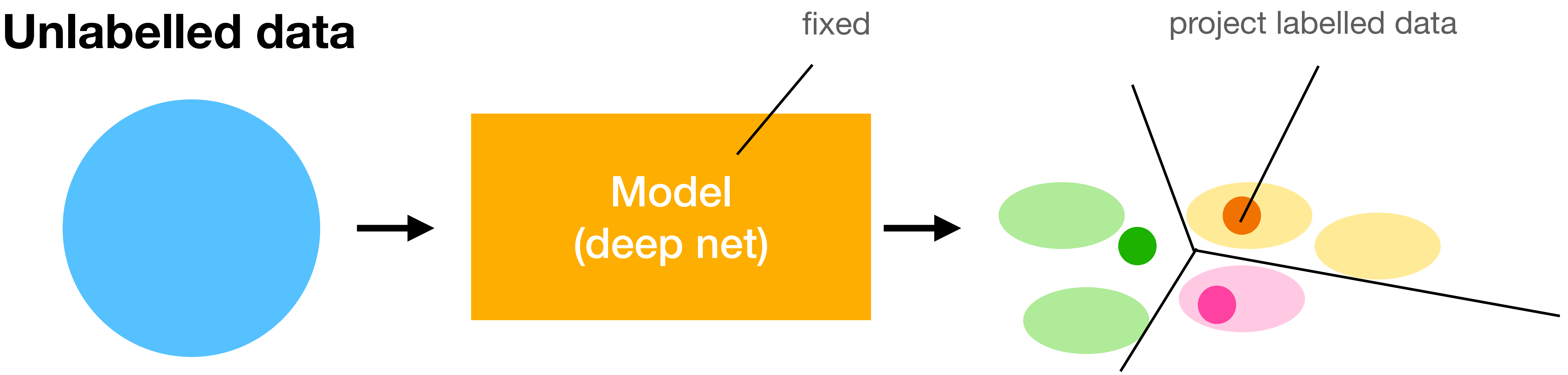
- **Fine-tuning** on the downstream tasks:
  - either all or only few last layers.
- Pros: Typically leads to best task performance (e.g. accuracy, IoU).
- Cons: Our model becomes task-specific; cannot be re-used for other tasks.

Evaluating SSL models



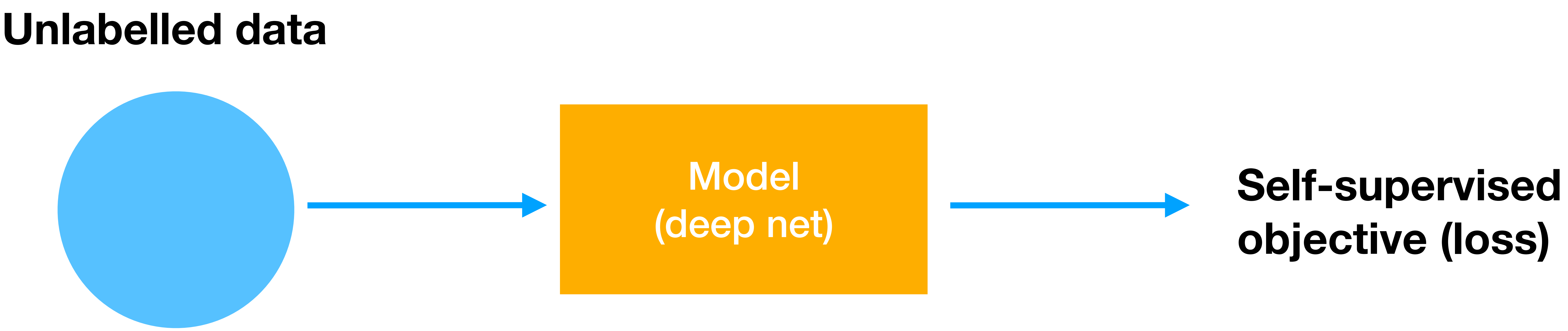
- **Linear probing:**
  - only learn the new linear projection. The model parameters remain fixed.
- Pros: We can re-use the model for other tasks by training multiple linear projections.
- Cons: Typically worse task accuracy than fine-tunining.

Evaluating SSL models



- **k-NN classification:**
  - project labelled data in the embedding space.
  - classify datapoints based on the class of its k nearest neighbours.
- Pros: Same as linear probing (versatility), but no learning is necessary.
- Cons: Prediction can be a bit costly
  - linear search complexity due to high feature dimensionality.

Self-supervised learning



How do we train deep models without labels?

- Goal: define training objectives with some relation to our target objective;
- By training the model on these objectives, we hope to learn something useful about our data.

Categories of self-supervision

- Pretext tasks
- Contrastive learning
- Non-contrastive learning

Categories of self-supervision

- **Pretext tasks**
- Contrastive learning
- Non-contrastive learning

Pretext tasks

Idea: Solve a different task with available (generated) supervision

Caveats:

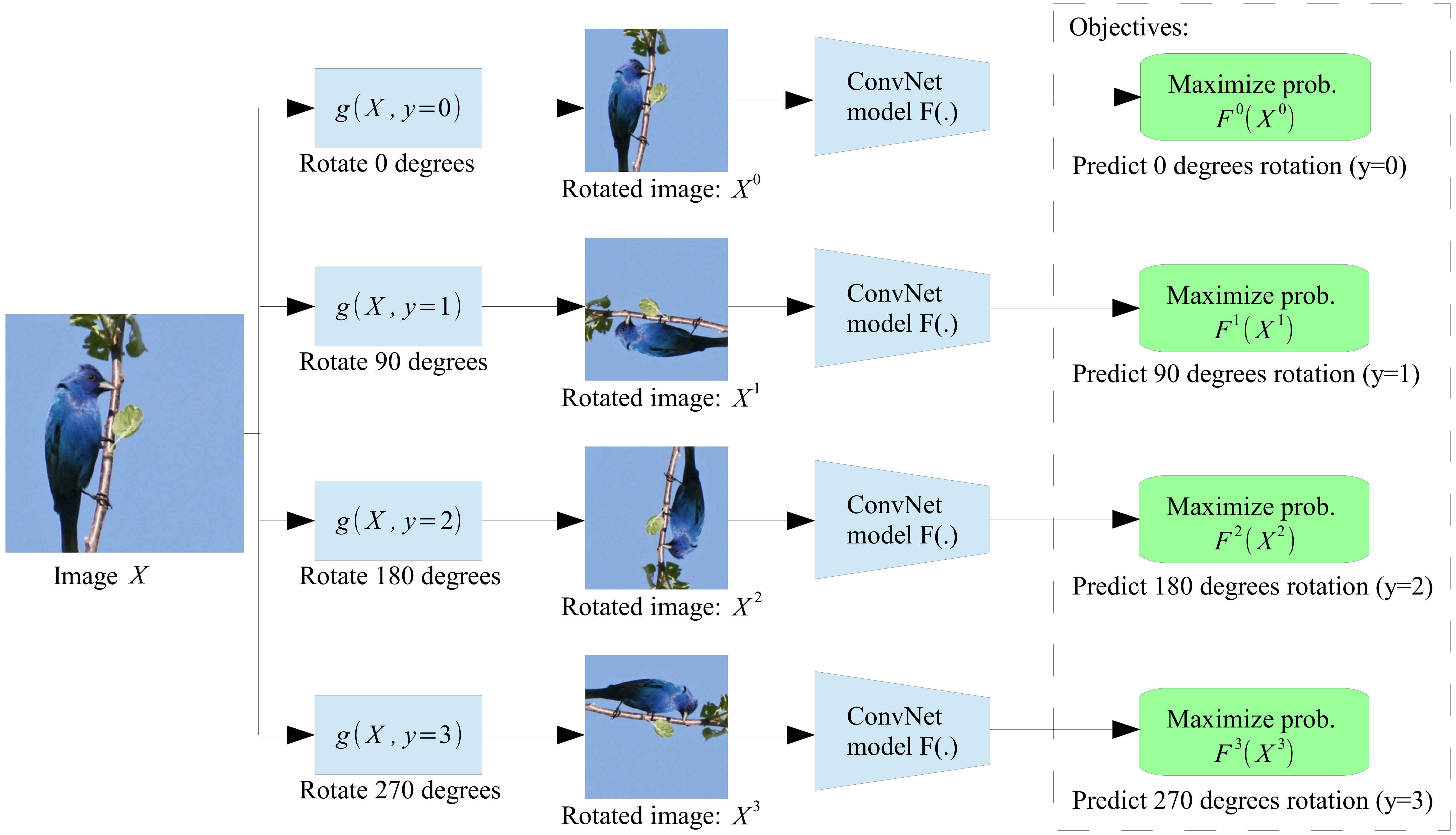
- The task should have some relation to our goal task;
- Finding an effective pretext task has been a heavily researched;
- The deep net will always try to cheat, i.e. find “shortcut” solutions:
    - see Geirhos et al., “Shortcut learning in deep neural networks” (2020).

Pretext Task: Rotation

- Task: predicting image rotation.
- Training process outline:
  - Quantise rotation angles (e.g. 0, 90, 180, 270 – 4 classes).
  - Sample an image from the training dataset.
  - Rotate the image by one of the pre-defined angles.
    - This defines our “ground-truth” label.
  - Train the network to predict the correct rotation class.

Gidaris et al., “Unsupervised representation learning by predicting image rotations” (2018).

Pretext Task: Rotation



CE Loss

QUIZ: Are we always expected to minimise this loss?

Gidaris et al., "Unsupervised representation learning by predicting image rotations" (2018).

Pretext Task: Rotation

- This leverages the photographic bias in typical image datasets
  - i.e. photographed objects have prevalent orientation.
- Otherwise, there is no canonical pose, hence the rotation angles are meaningless
  - A thought experiment: add all rotated images to the original dataset.

- 这利用了典型图像数据集中的摄影偏差

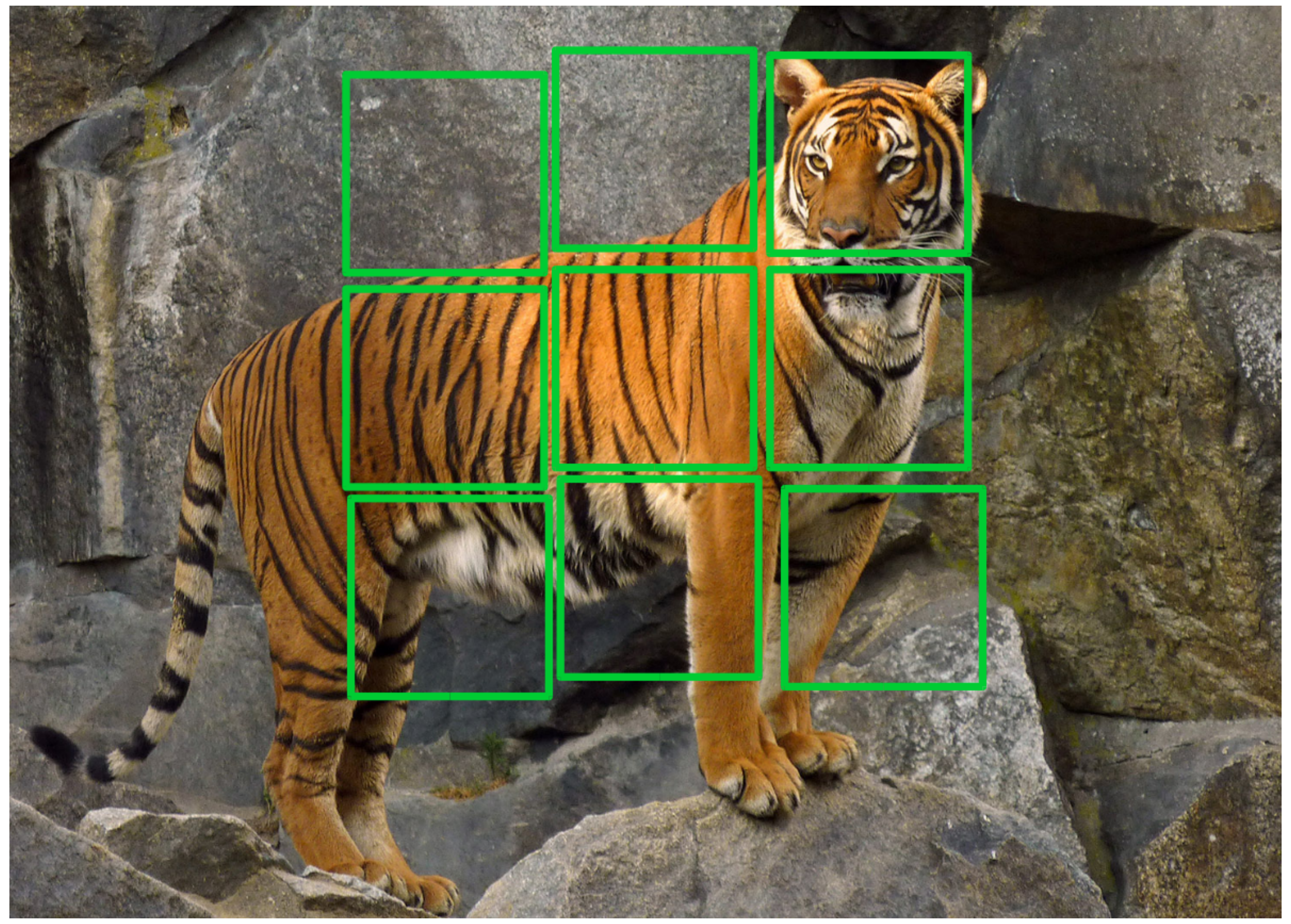
- 即拍摄的物体具有普遍的方向性。

- 否则，就不存在典型的姿势，因此旋转角度也就失去了意义

- 一个思想实验：将所有旋转过的图像添加到原始数据集中。

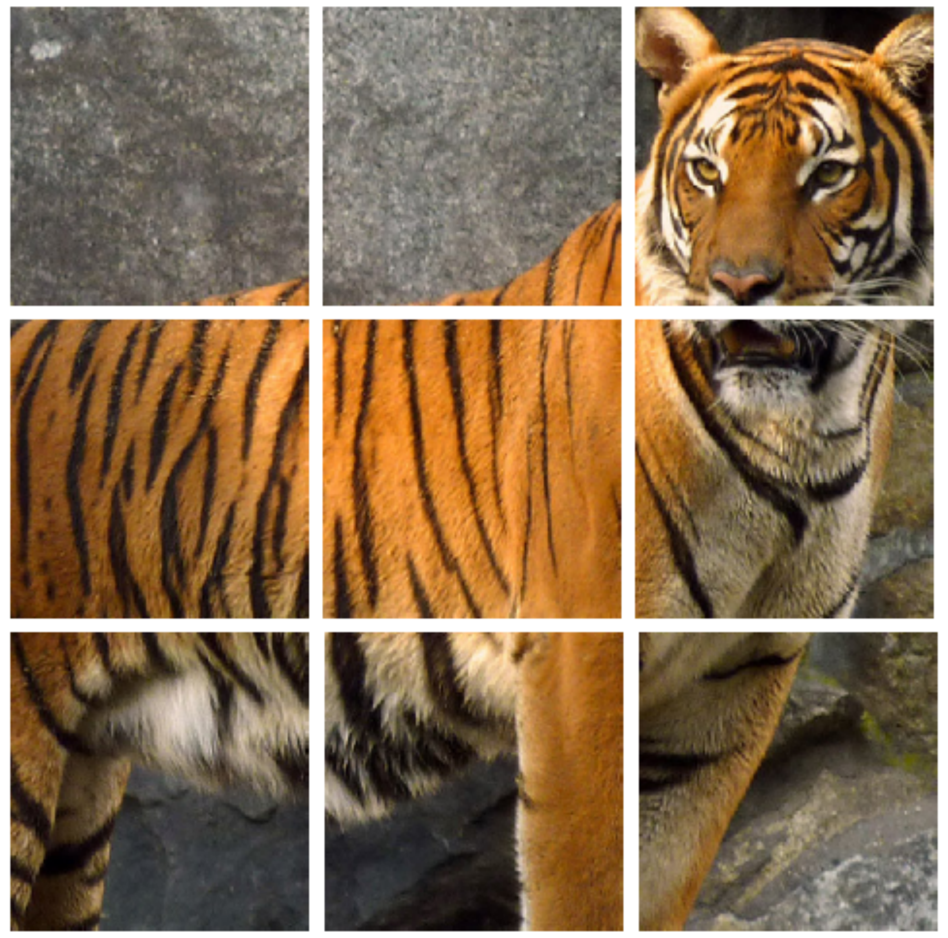
Pretext Task: Jigsaw puzzle

1. Extract patches



2. Input: shuffle

3. Output: reconstruct



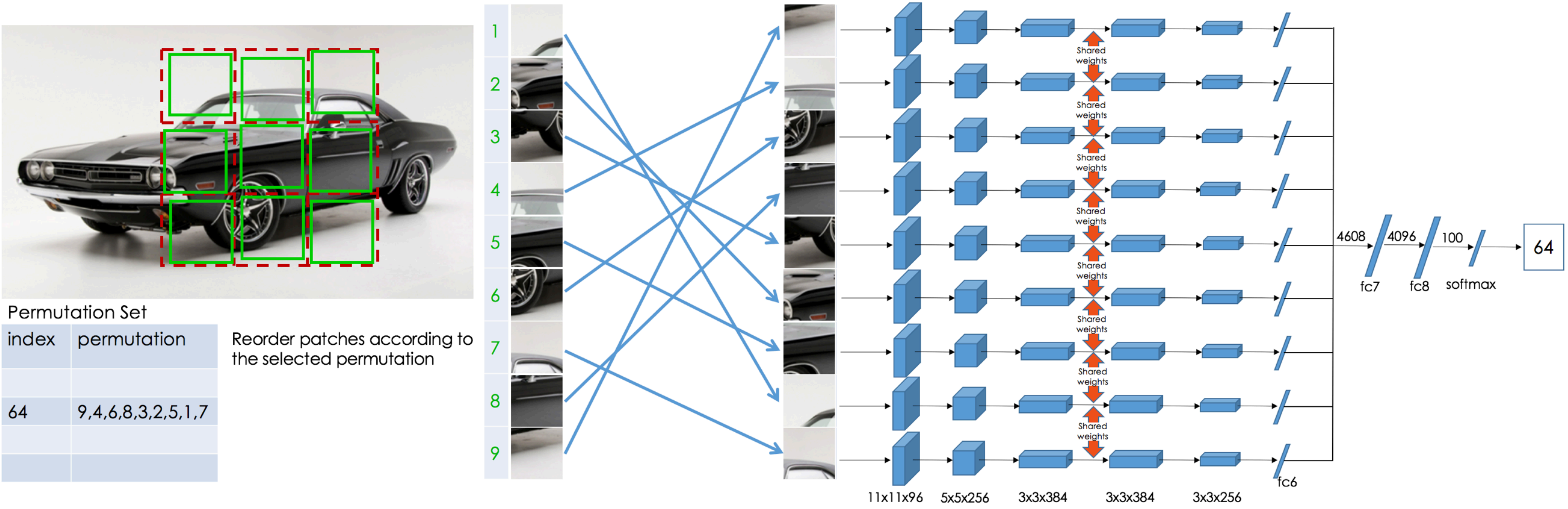
- Solving this task requires the model to learn spatial relation of image patches.

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

Pretext Task: Jigsaw puzzle

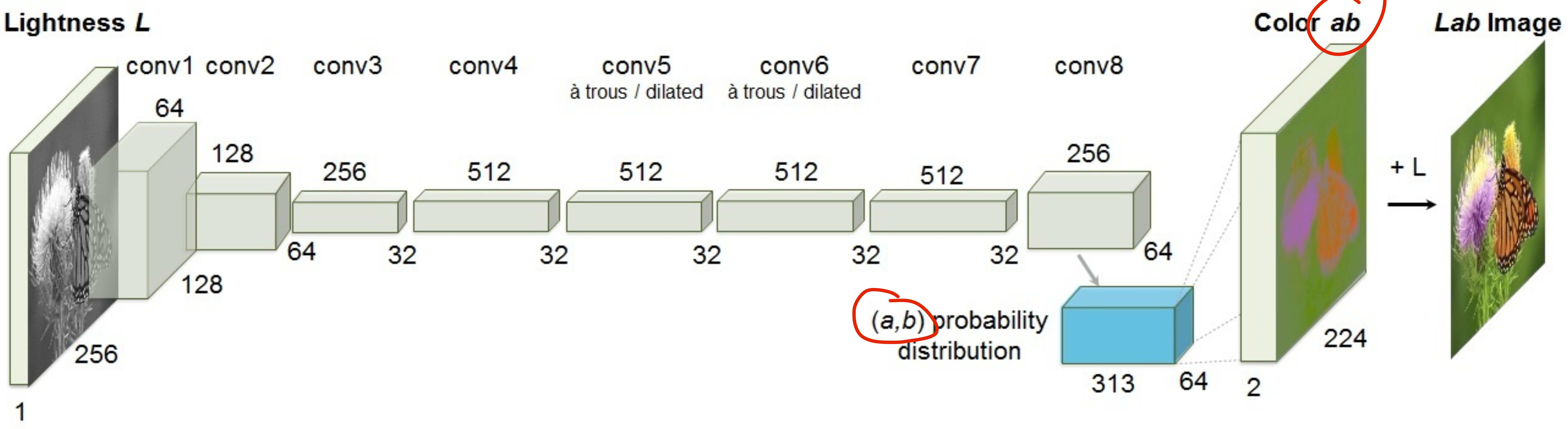
- Cast this task as a classification problem:
  - every permutation defines a class.

QUIZ: How many classes shall we define to solve a 9-piece puzzle?



Colorization

- Predicting the original colour of the images (in CIELAB colour space):
  - Intuition: proper colourisation requires semantic understanding of the image.



Zhang et al., “Colorful image colorization”. In ECCV, 2016.

Colorization

细微差别：

-这与从灰度图像预测 RGB 值不同（由于着色的多模态性）。

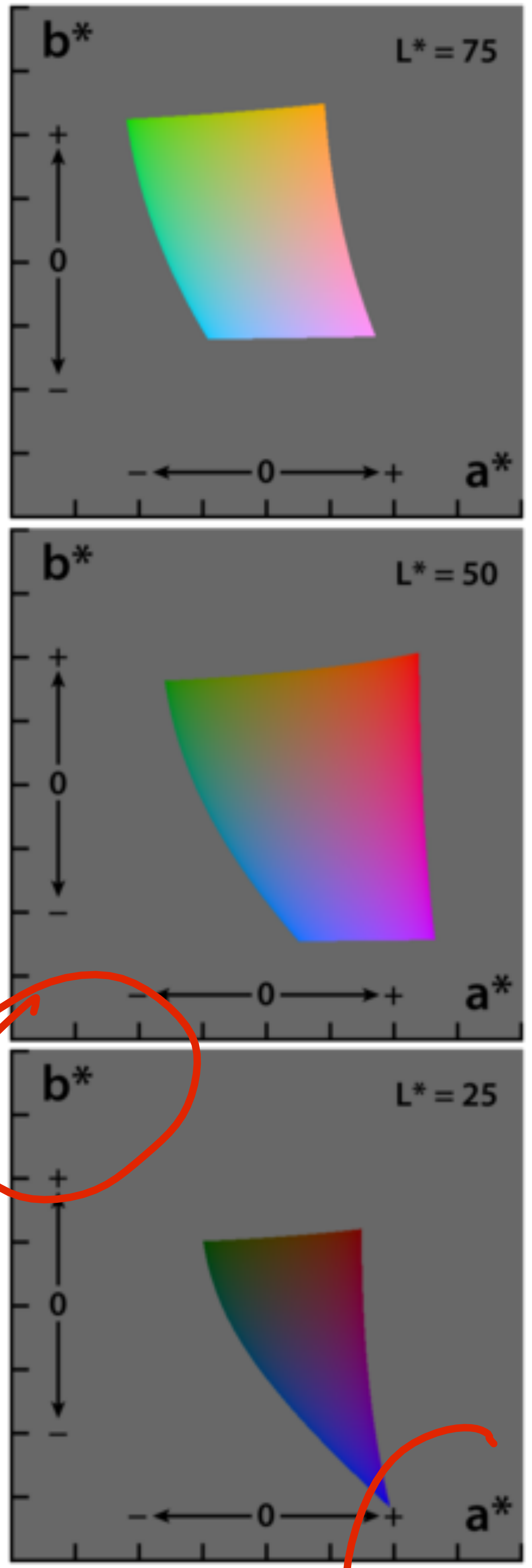
-相反，我们在 Lab 色彩空间中进行操作。回顾：
- L 代表感知亮度；
- a 和 b 表示四种独特的颜色（红、绿、蓝、黄）。

-实验室中的距离更有意义。

-我们将着色视为一个多项式分类问题。

Nuances:

- This is not the same as predicting RGB values from a greyscale image (due to multimodality of colorisation).
- Instead, we operate in Lab colour space. Recall:
  - L stands for perceptual lightness;
  - a and b expresses four unique colours (red, green, blue, yellow).
- Distances in Lab are more perceptually meaningful.
- We cast colorisation as a multinomial classification problem.



Zhang et al., “Colorful image colorization”. In ECCV, 2016.

Colorization

Application to legacy black-and-white photos:

Zhang et al., “Colorful image colorization”. In ECCV, 2016.

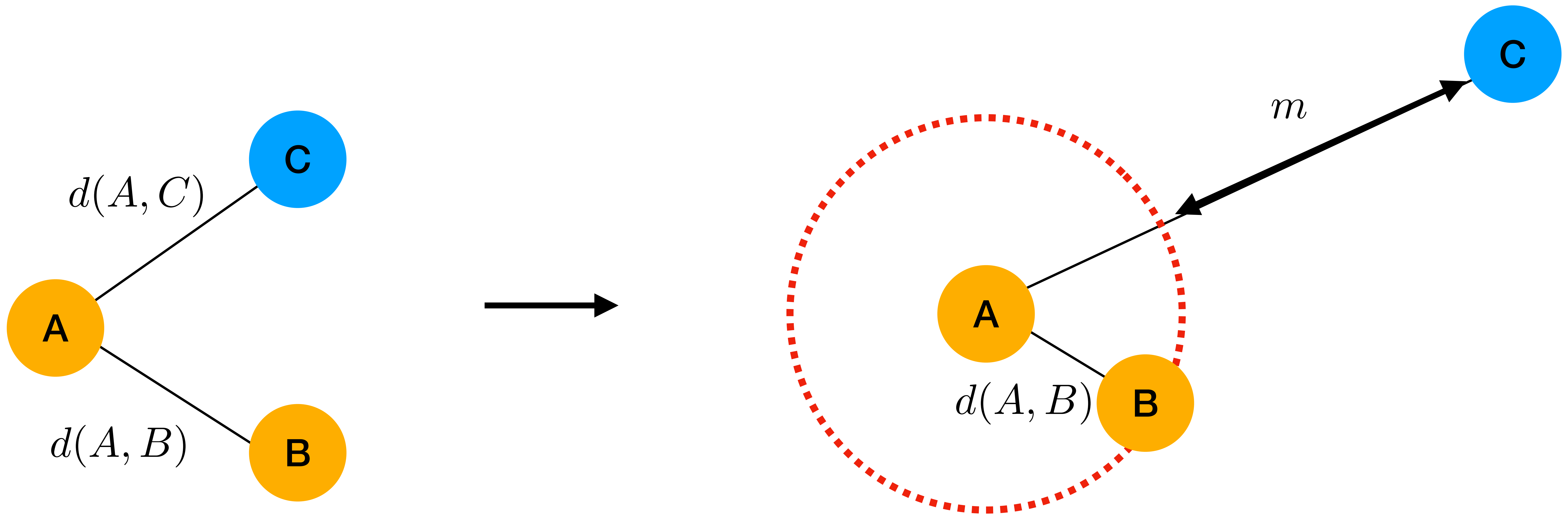
Categories of self-supervision

- Pretext tasks
- **Contrastive learning**
- Non-contrastive learning

Recall metric learning

$$\mathcal{L}(A, B, C) = \max(0, \|f(A) - f(B)\|^2 - \|f(A) - f(C)\|^2 + m)$$

Intuitive idea:

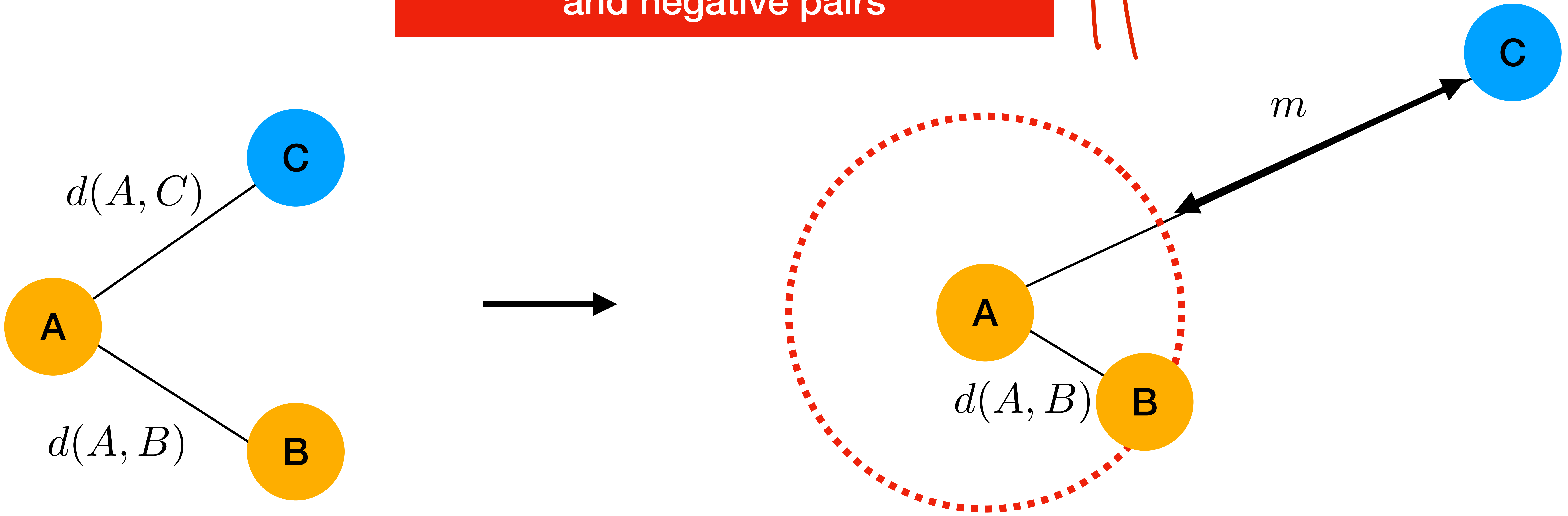


Recall metric learning

$$\mathcal{L}(A, B, C) = \max(0, \|f(A) - f(B)\|^2 - \|f(A) - f(C)\|^2 + m)$$

Intuitive idea:

We need labels to define positive and negative pairs



Contrastive learning

- Contrastive learning is an extension of metric learning to unsupervised scenarios.
- Idea:
  - Use data augmentation (e.g. cropping) to create a positive pair of the same image;
  - Use other images to create (many) negative pairs.

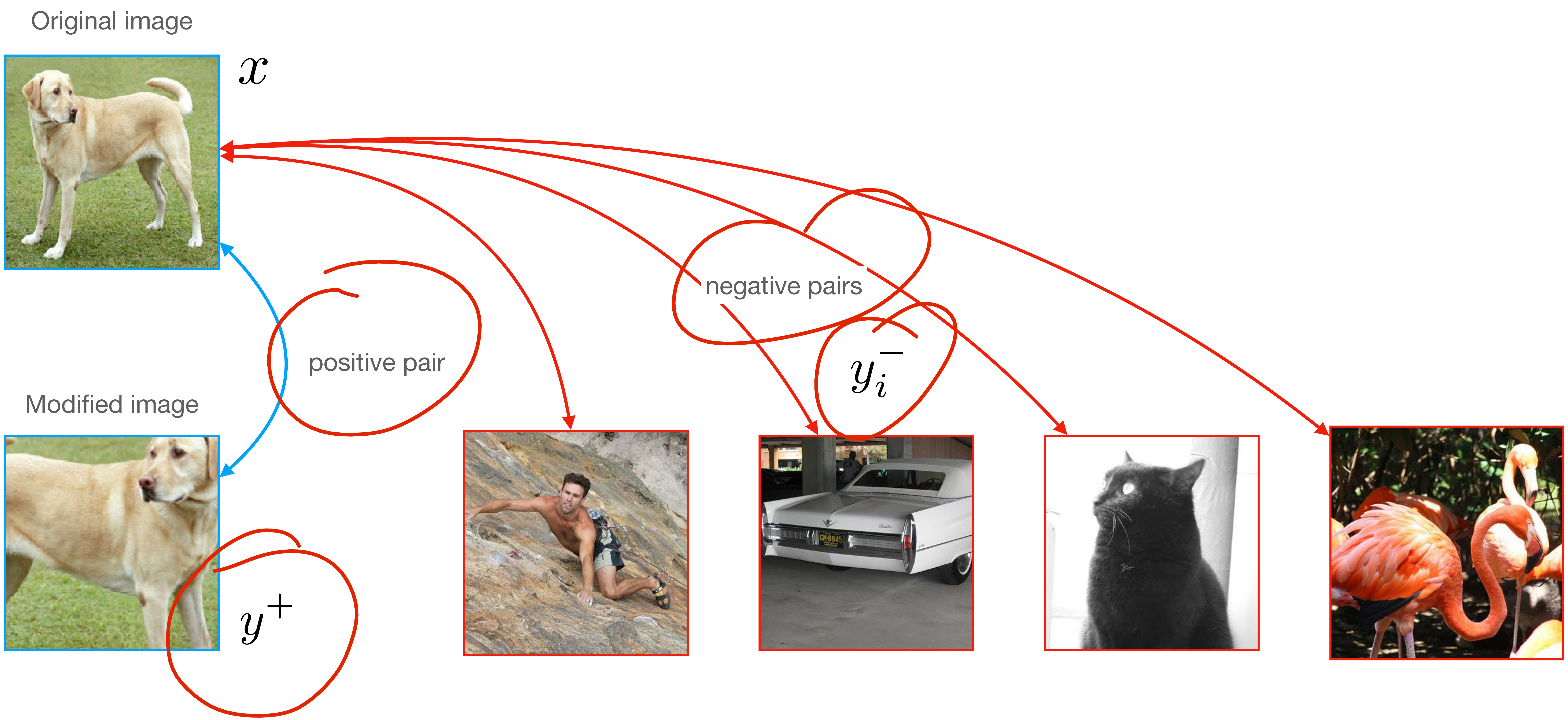
- 对比学习是度量学习在无监督情况下的延伸。

-理念

- 利用数据扩增（如裁剪）创建同一图像的正对图像；

- 使用其他图像创建（许多）负对。

Contrastive learning: Example



Contrastive learning: Example

- Represent each image a single feature vector (e.g. last layer in a CNN).
- Consider **cosine similarity** of two such vectors,:

$$d(x, y) = \frac{x \cdot y}{\|x\| \|y\|}$$

Note: $d(x, y) \in [-1, 1]$

- For a given set $\{x, y^+, \{y_i^-\}_{i=1,\dots,n}\}$ compute contrastive score:

$$s(x) = e^{d(x,y^+)/\tau} / \left(e^{d(x,y^+)/\tau} + \sum_{i=1}^{n} e^{d(x,y_i^-)/\tau}\right)$$

Contrastive learning: Example

$$s(x) = e^{d(x,y^+)/\tau} / \left(e^{d(x,y^+)/\tau} + \sum_{i=1}^{n} e^{d(x,y_i^-)/\tau}\right)$$

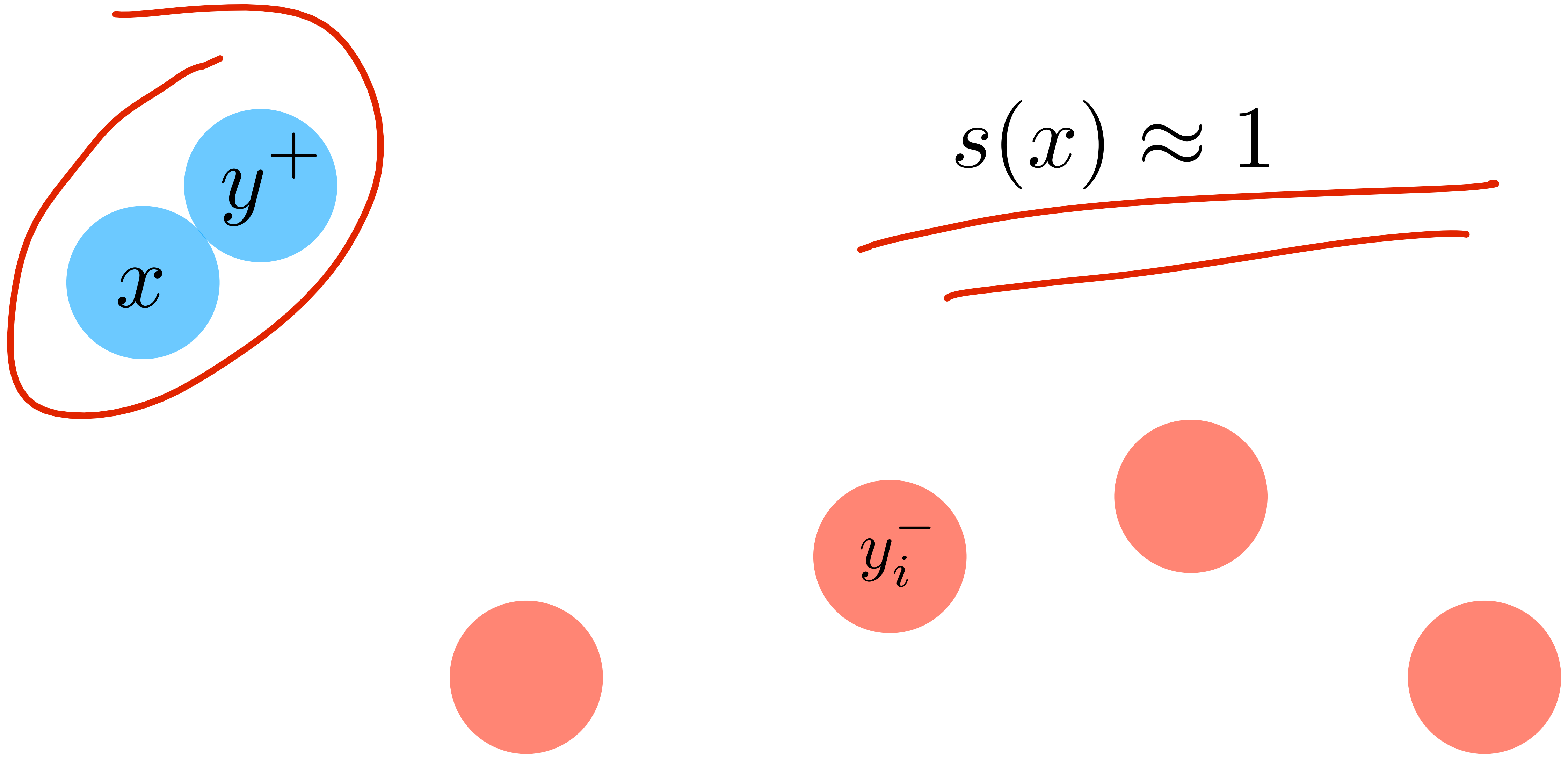
Observations:

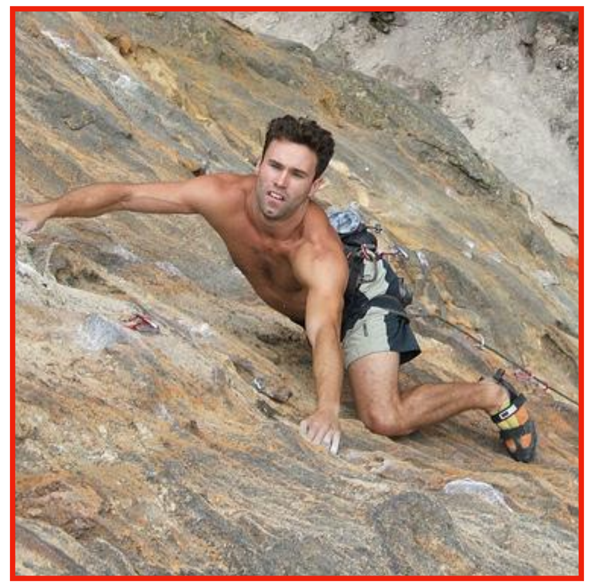
- Temperature τ – a hyperparameter (usually between 0.01 and 1.0).
- What is the range?
- What does it mean when it reaches maximum/minimum?
- We clearly want to maximise this value! (many implementations)
- Example loss: $-\log s(x)$

Contrastive learning: Example

$s(x) \approx 1$

Original image

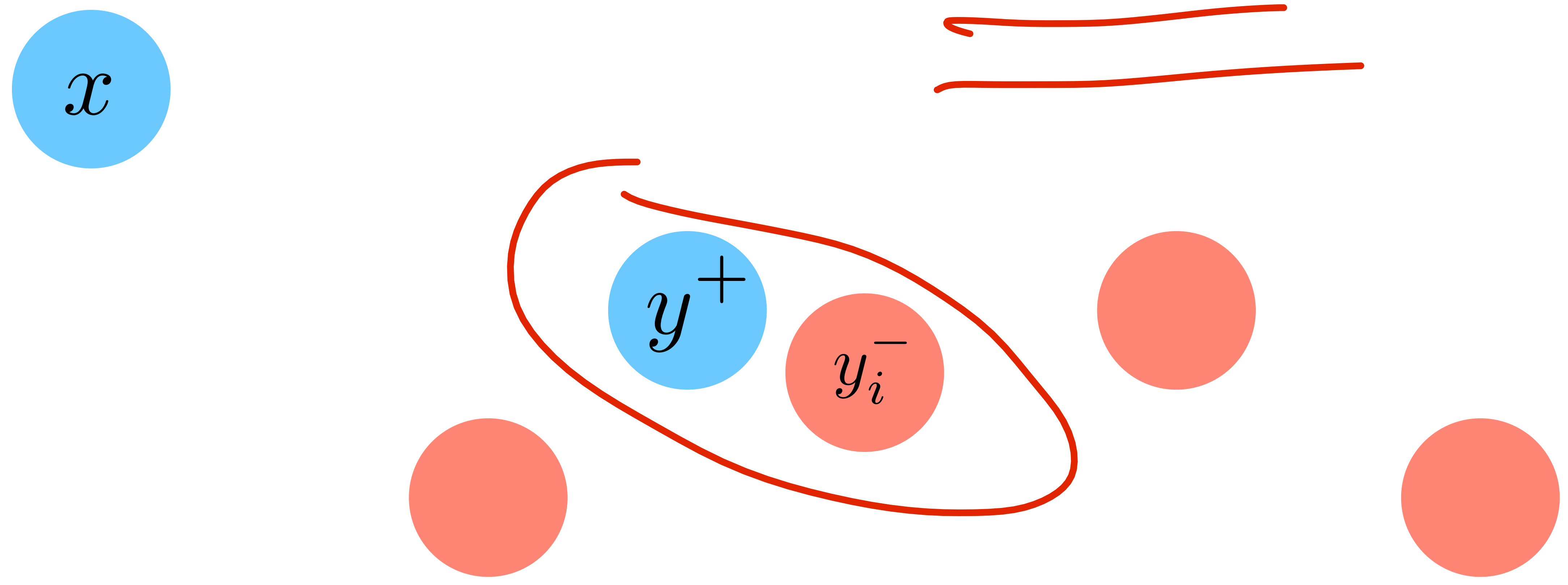


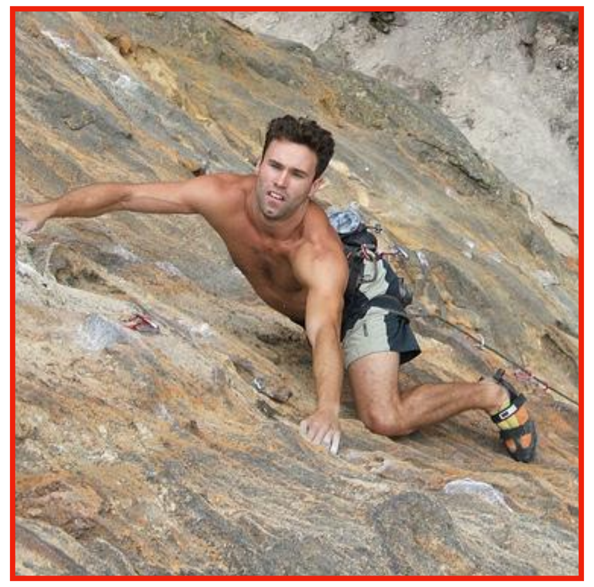


Contrastive learning: Example

Original image

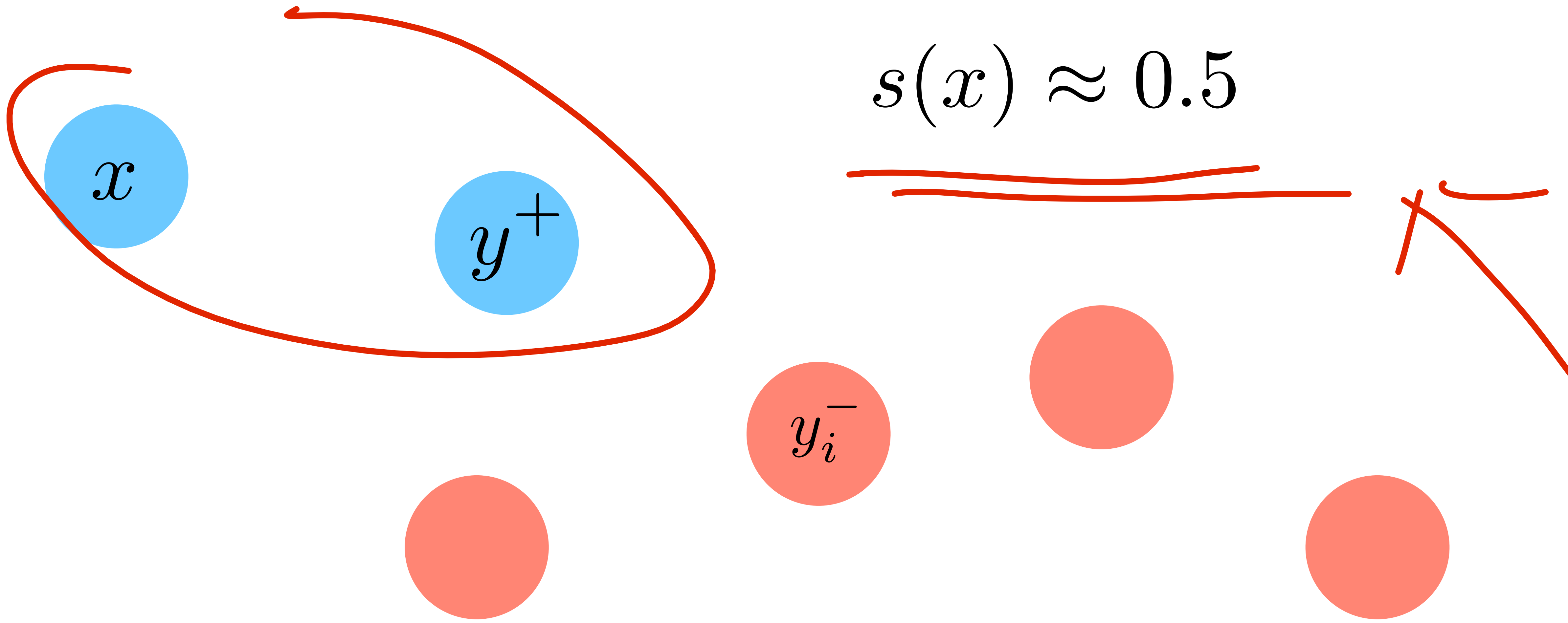
$s(x) \approx 0$



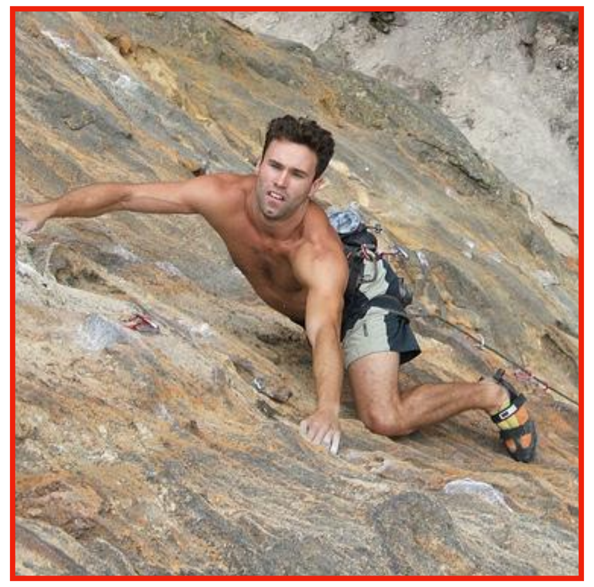


Contrastive learning: Example

Original image



$$s(x) \approx 0.5$$



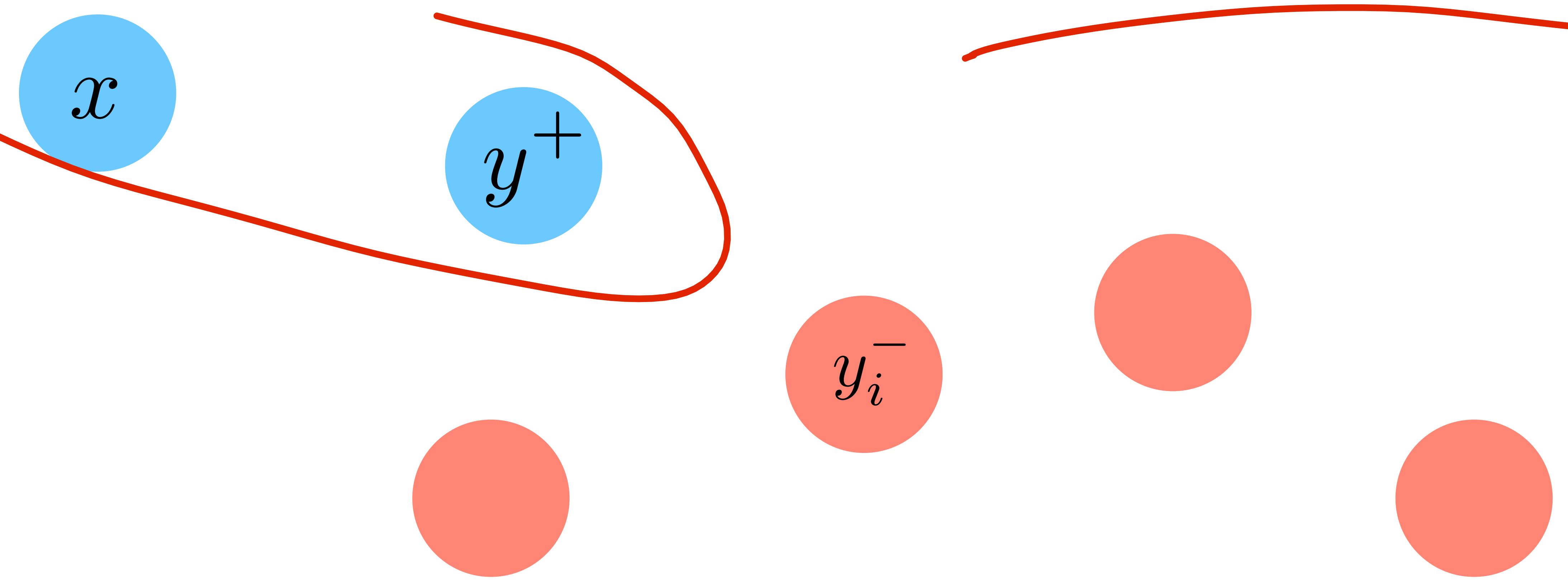
Contrastive learning: Example

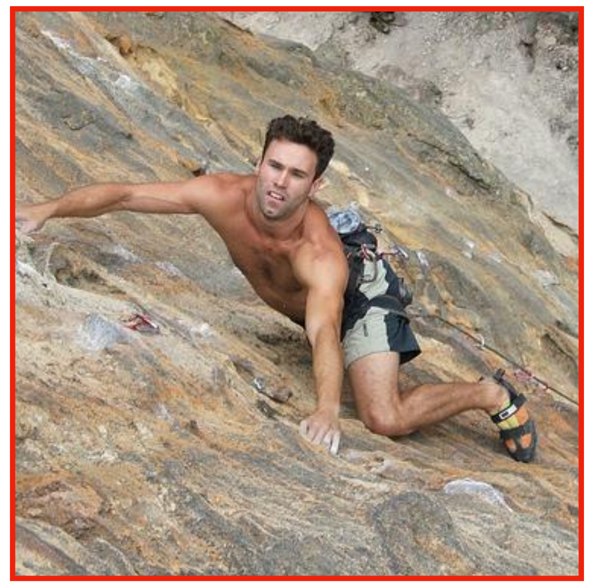
$S(x) = 0.5$

temp ↓

$s(x) \approx 1$ very low temperature

Original image

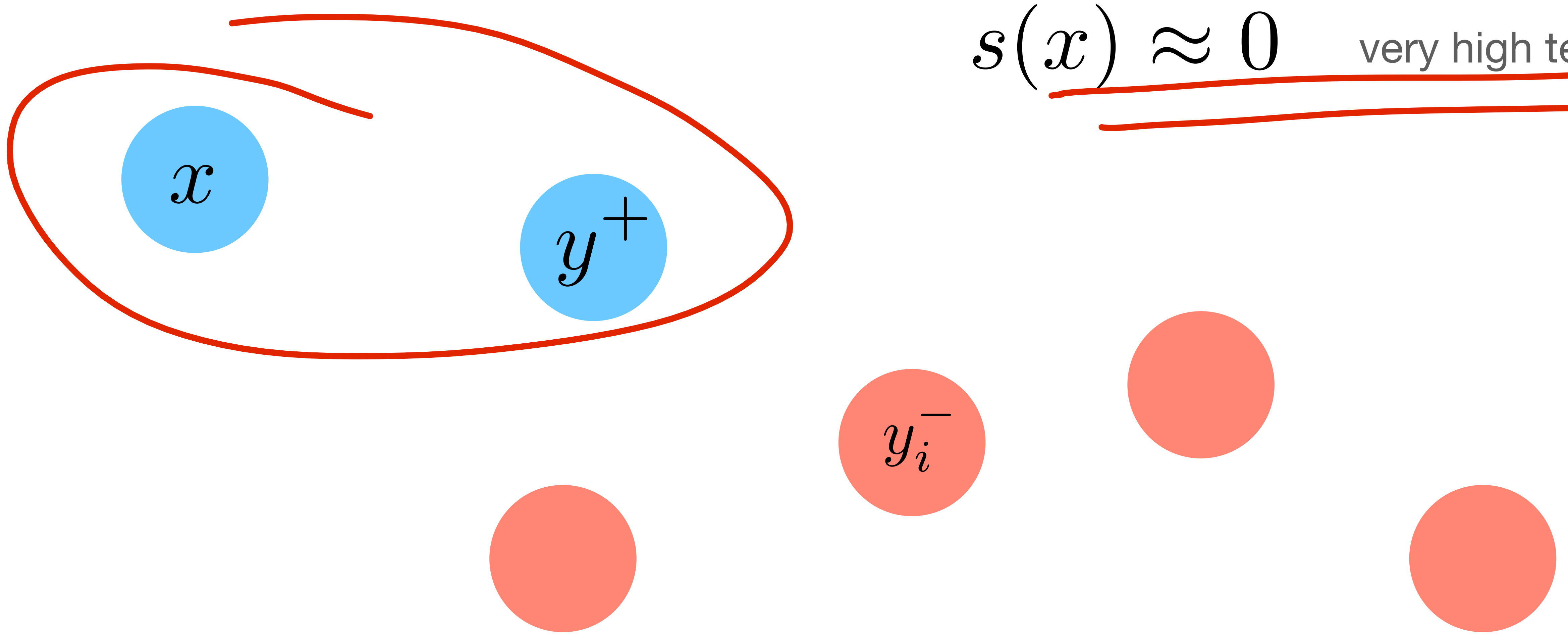


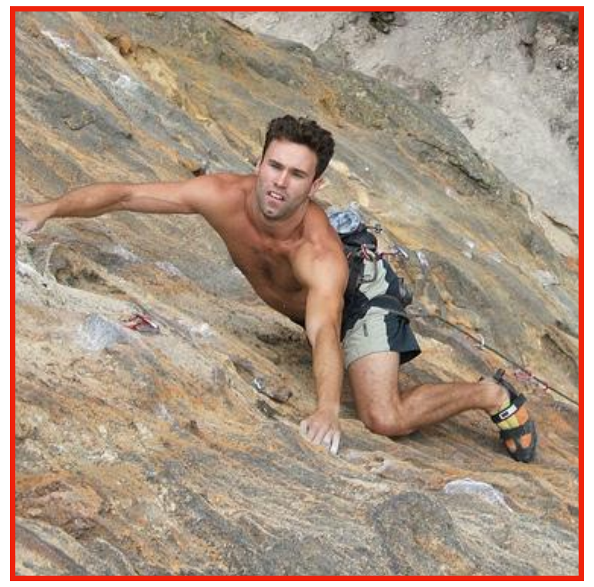


Contrastive learning: Example

Original image

$s(x) \approx 0$ very high temperature





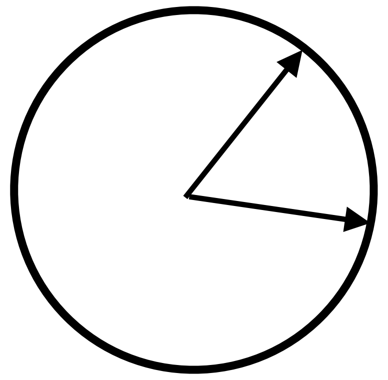
The temperature can be thought of as a soft margin.

Contrastive learning: Intuition

- Note that we normalise the feature embeddings:

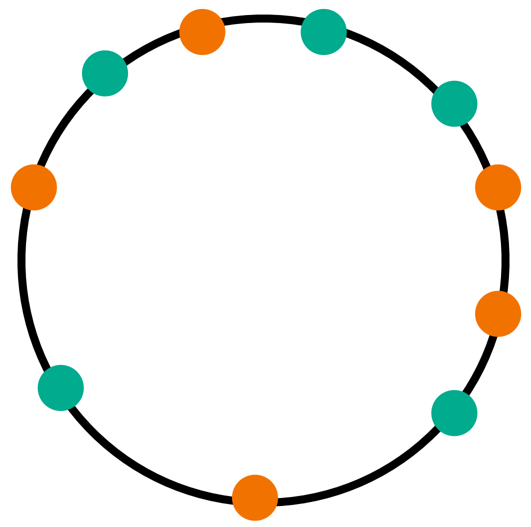
$$d(x, y) = \frac{x \cdot y}{\|x\|\|y\|}$$

- Every unit vector corresponds to a point on a unit sphere:



- The goal of contrastive learning is to cluster the representation on the sphere:

initially:



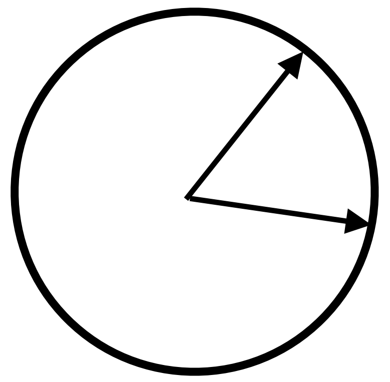
Wang and Isola, “Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere”. In ICML 2020.

Contrastive learning: Intuition

- Note that we normalise the feature embeddings:

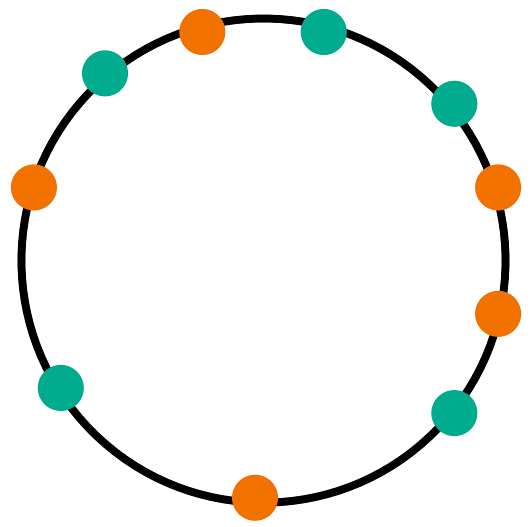
$$d(x, y) = \frac{x \cdot y}{\|x\| \|y\|}$$

- Every unit vector corresponds to a point on a unit sphere:

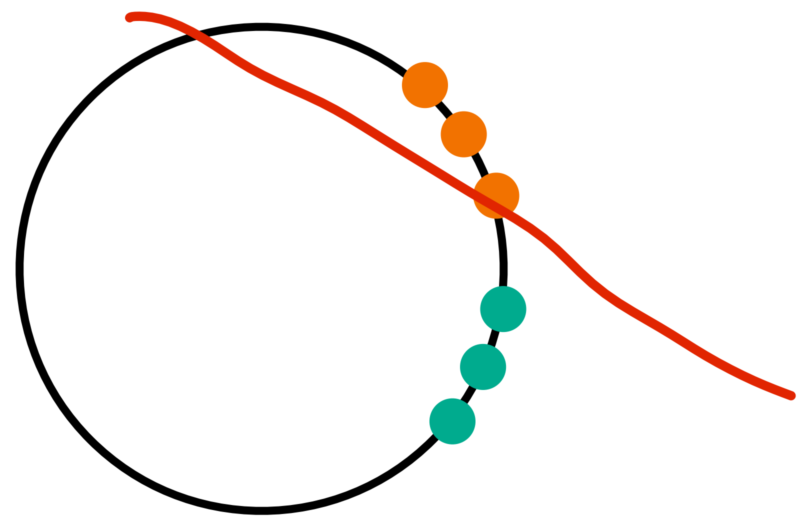


- The goal of contrastive learning is to cluster the representation on the sphere:

initially:



after training:



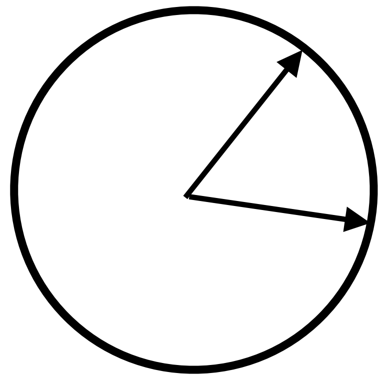
Wang and Isola, “Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere”. In ICML 2020.

Contrastive learning: Intuition

- Note that we normalise the feature embeddings:

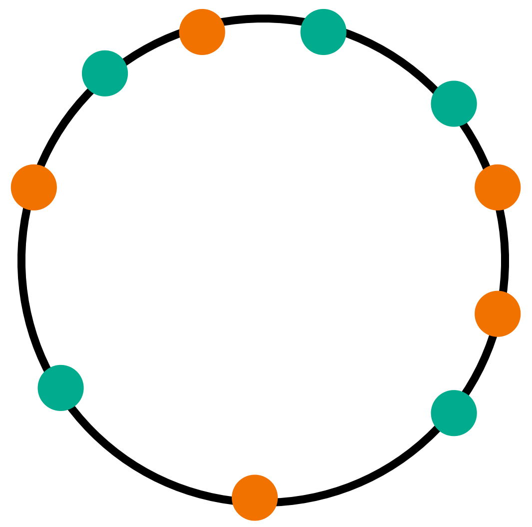
$$d(x,y) = \frac{x \cdot y}{\|x\|\|y\|}$$

- Every unit vector corresponds to a point on a unit sphere:

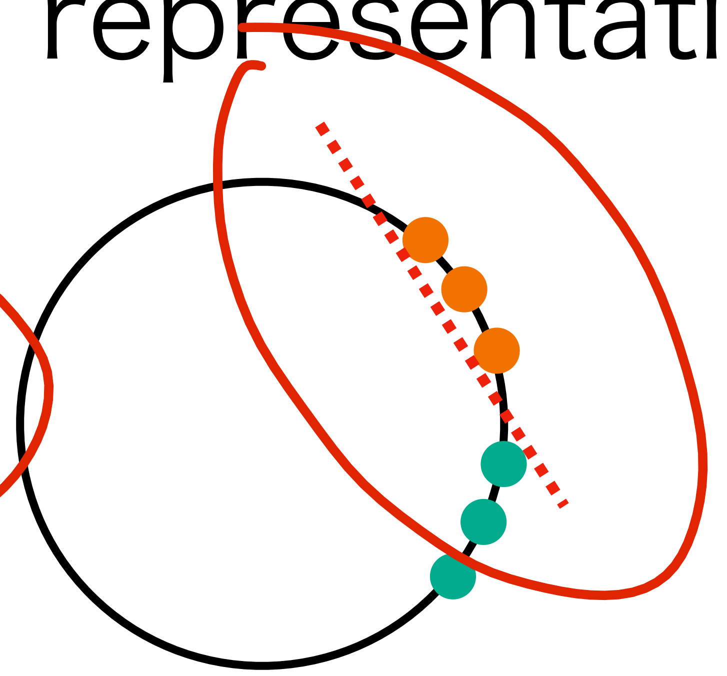


- The goal of contrastive learning is to cluster the representation on the sphere:

initially:



after training:



- The points on the sphere will be linearly separable!

Wang and Isola, “Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere”. In ICML 2020.

Deep Frameworks for SSL

- SimCLR (Chen et al., 2020)
- BYOL (Grill et al., 2020)
- MoCo (He et al., 2020, also MoCo v2, v3);
- DINO (Caron et al., 2021, also DINOv2)
- Masked Autoencoders (He et al., 2022)
- … and many more (BeiT, iBOT, SimMIM, I-JEPA…)

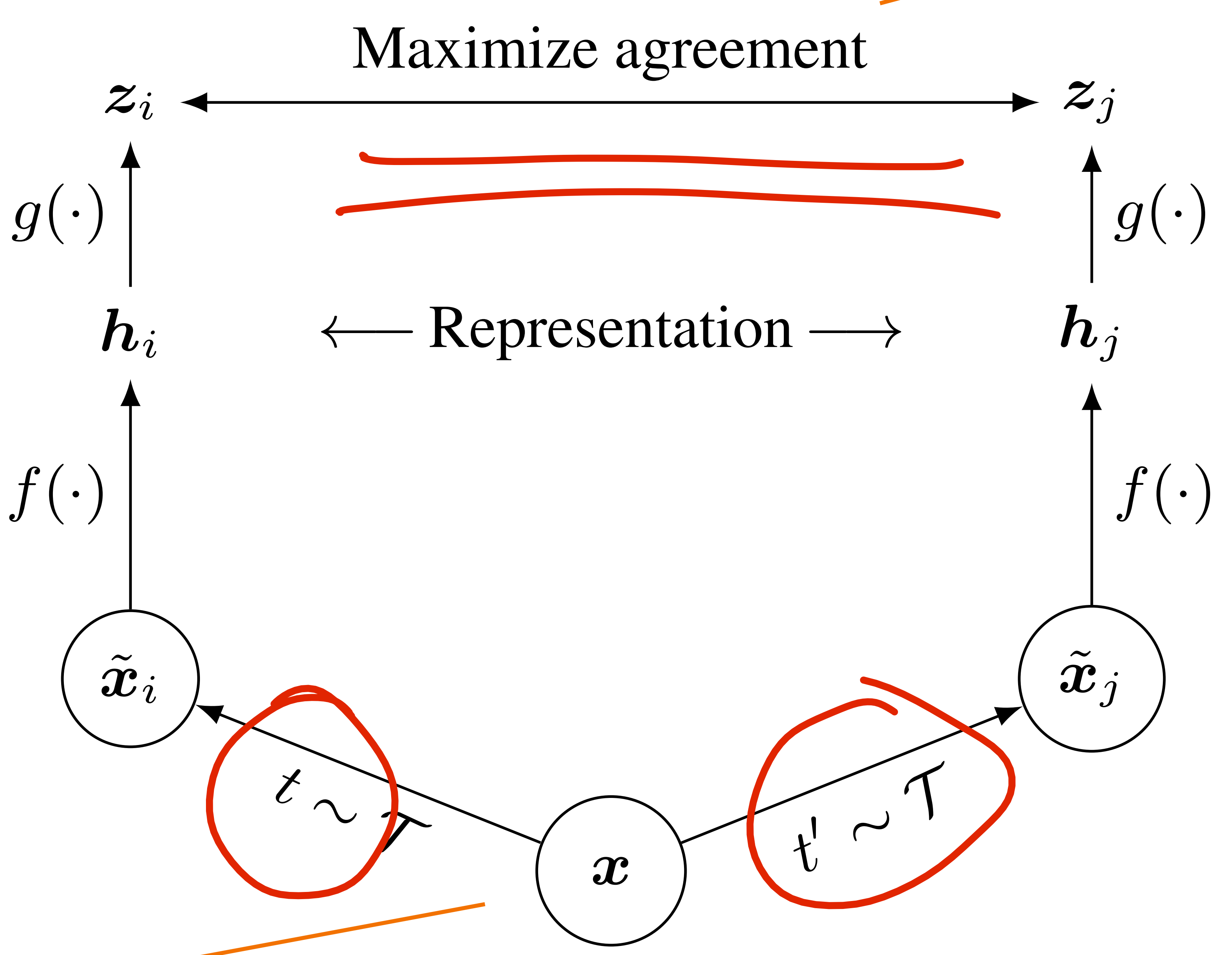
SimCLR

"A simple framework for contrastive learning"

contrastive loss

projection head (MLP)

base encoder (CNN)



input image

Chen et al., "A Simple Framework for Contrastive Learning of Visual Representations" (2020)

SimCLR: Generating Positive Pairs

Original image

Crop and resize

Crop, resize and flip

Greyscale

Colour jitter

Rotation

Cutout

Gaussian noise

Gaussian blur

Sobel filtering

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

SimCLR

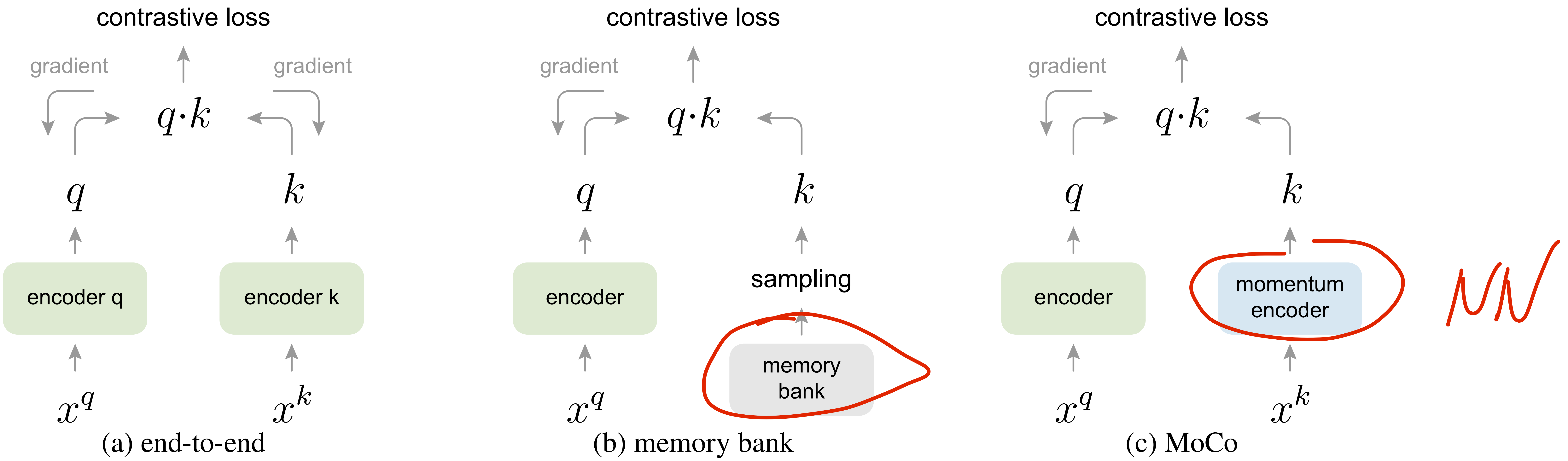
- Conceptually very simple.
- Best results require substantial computational resources:
  - batch size 8192 (16382 negative pairs).

- Why do we need a large batch size?
  - A subject of ongoing research (see Chen et al., NeurIPS '22).
  - Intuition: More negative samples reduce the gradient bias.

Chen et al., "A Simple Framework for Contrastive Learning of Visual Representations" (2020)

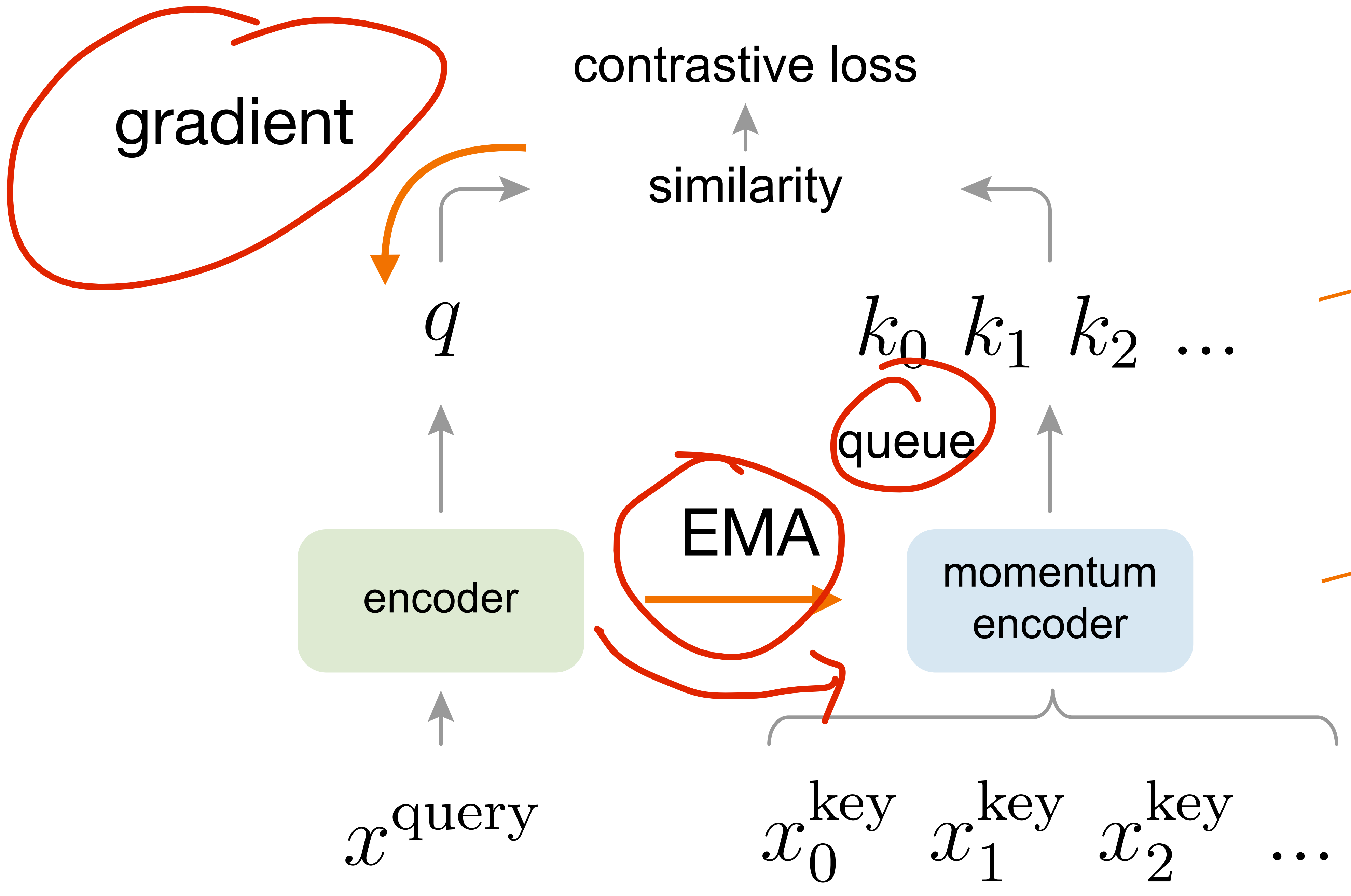
Momentum contrast

- Contrastive learning requires large sets of negative pairs.
- Can we reduce the GPU memory footprint?



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

MoCo



memory queue

slowly changing copy of the encoder (exponentially moving average, EMA)

$$\theta_k \leftarrow m\theta_k + (1 - m)\theta_q \qquad m \in (0, 1)$$

θ_q encoder parameters

θ_k momentum encoder parameters

QUIZ: How to set m? What if it is too high/low?

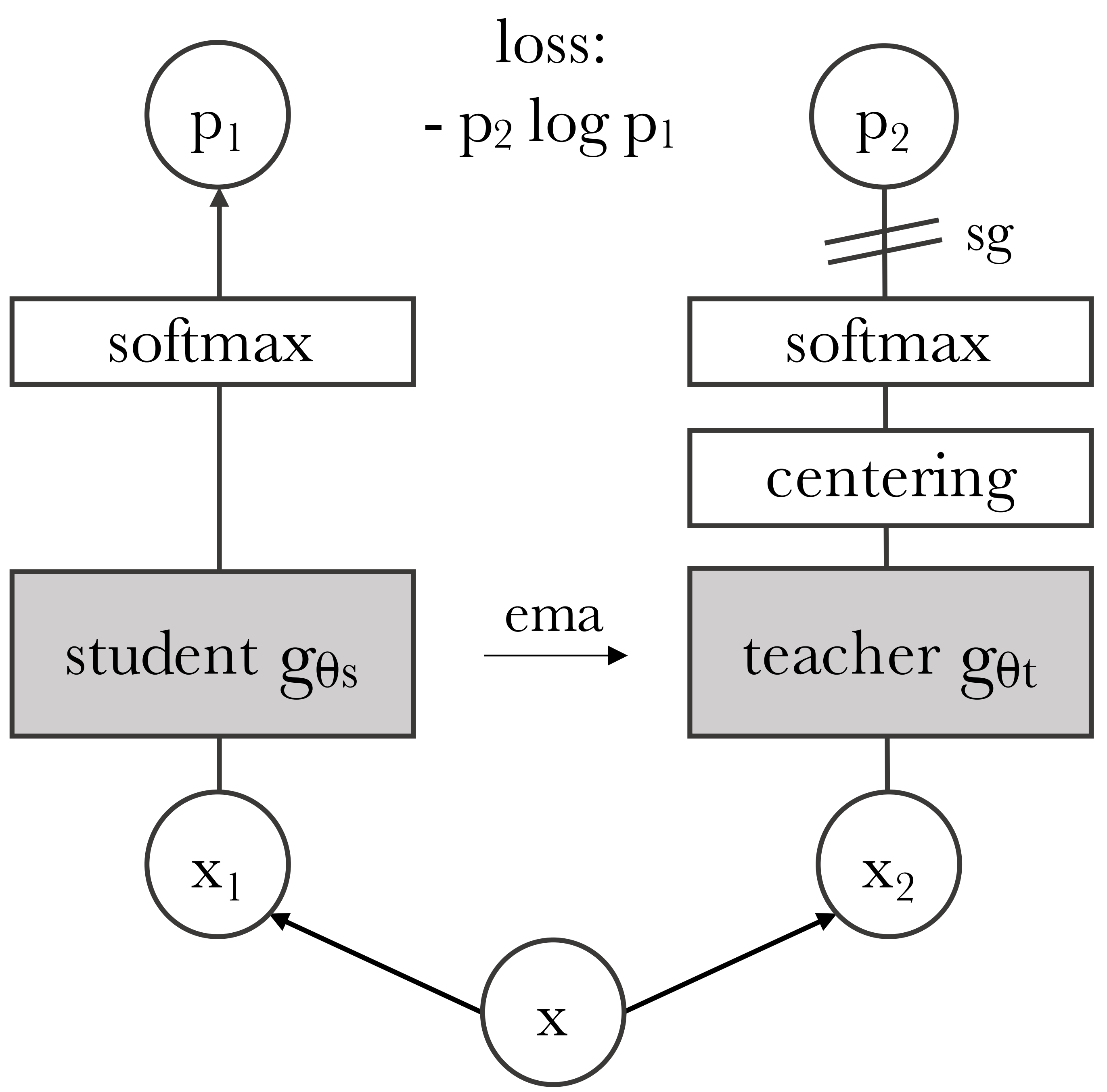
He et al., "Momentum Contrast for Unsupervised Visual Representation Learning", (2020).

Categories of self-supervision

- Pretext tasks
- Contrastive learning
- **Non-contrastive learning**

DINO*

- Simple to implement and to train.
- Broad application spectrum.



* self-distillation with no labels.

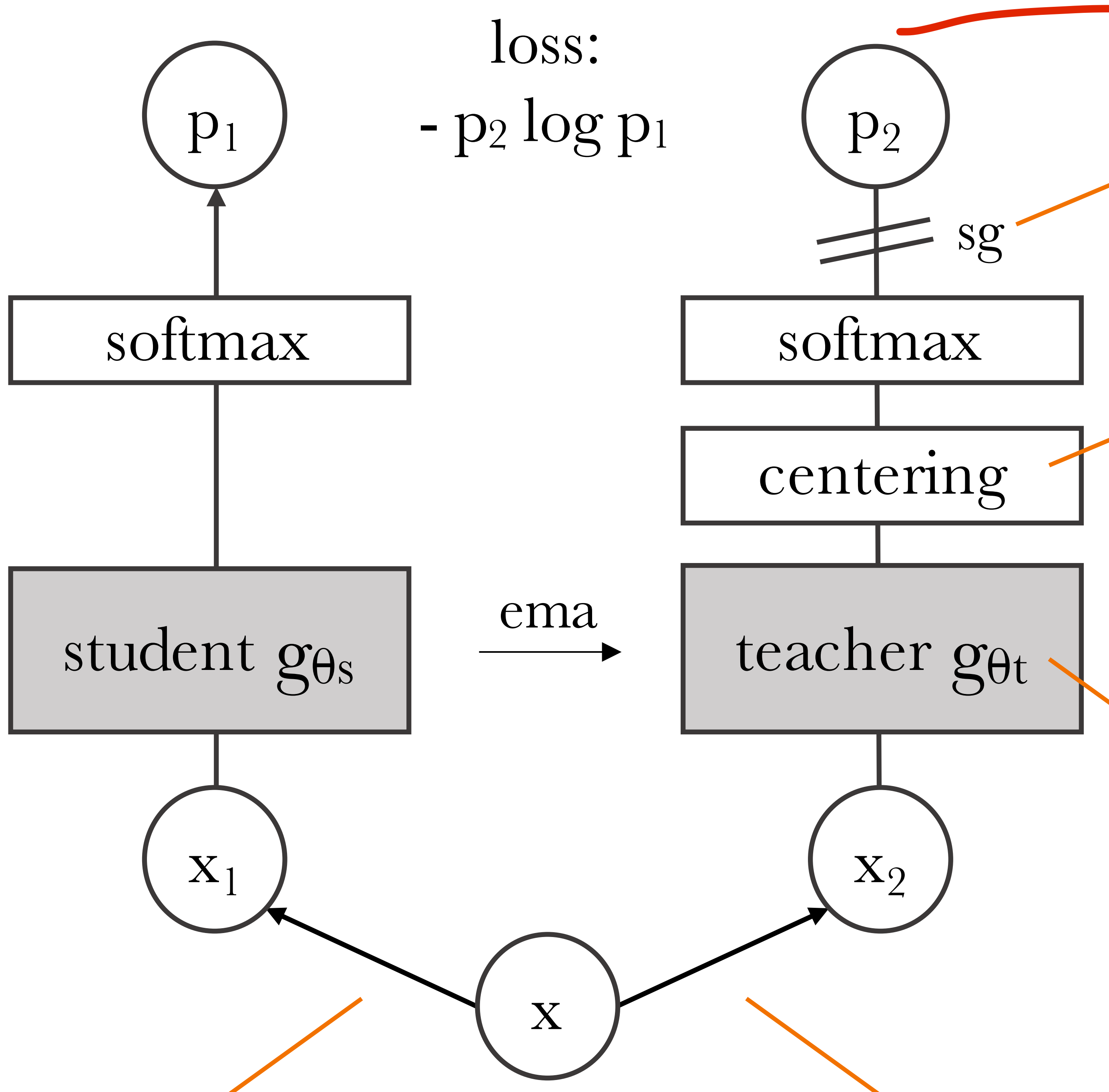
Caron et al., “Emerging Properties in Self-Supervised Vision Transformers” (2021).

DINO

Teacher distribution is 'sharpened' with lower temperature

Gradient is not propagated through the teacher

Ensure expected value of targets is zero (avoids collapse in conjunction with sharpening teacher logits).



We've already seen this in MoCo (just different terminology)

The student observes only local views

"Global" views to the teacher

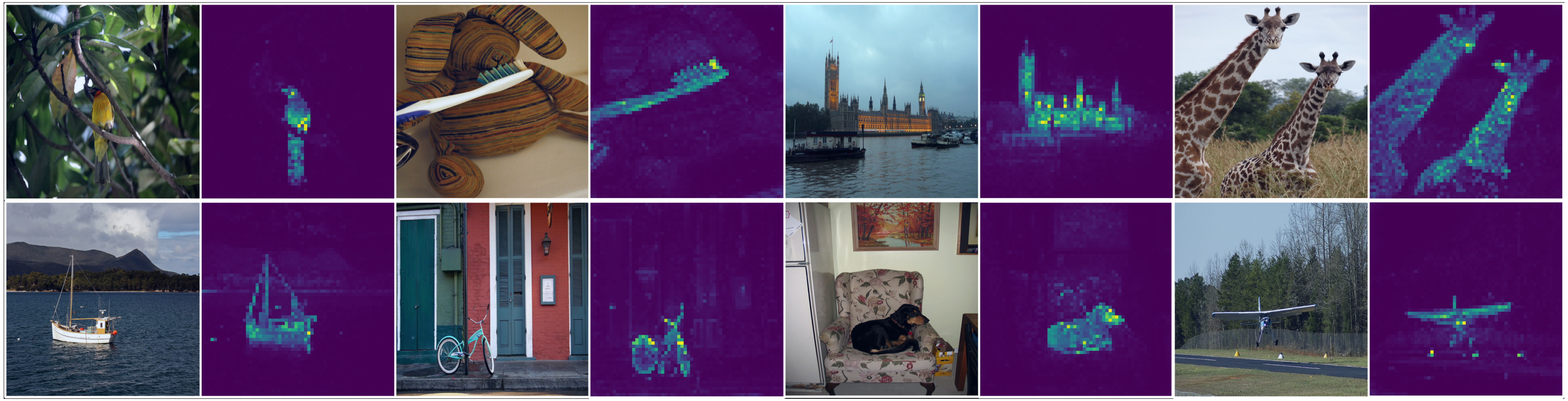
Algorithm 1 DINO PyTorch pseudocode w/o multi-crop.

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

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

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

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

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

Caron et al., "Emerging Properties in Self-Supervised Vision Transformers" (2021).

DINO: Attention maps

- Self-attention maps in the last layer of a ViT with respect to [CLS] token.



Caron et al., “Emerging Properties in Self-Supervised Vision Transformers” (2021).

DINO: Attention maps

- Self-attention maps in the last layer of a ViT with respect to [CLS] token.
- Different heads also focus on semantically distinguishable features:

Caron et al., “Emerging Properties in Self-Supervised Vision Transformers” (2021).

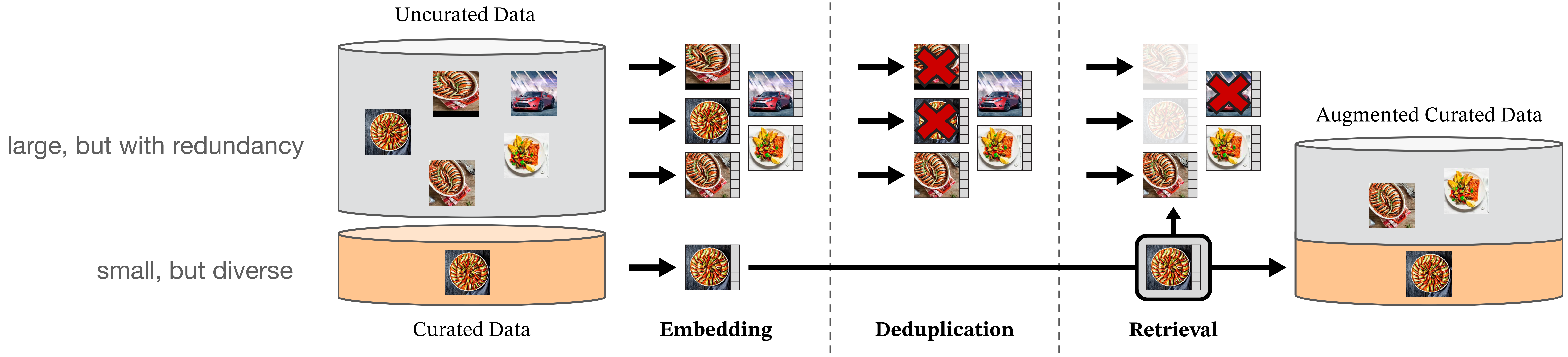
DINO Applications

- The attention maps are temporally stable:

Caron et al., “Emerging Properties in Self-Supervised Vision Transformers” (2021).

DINOv2

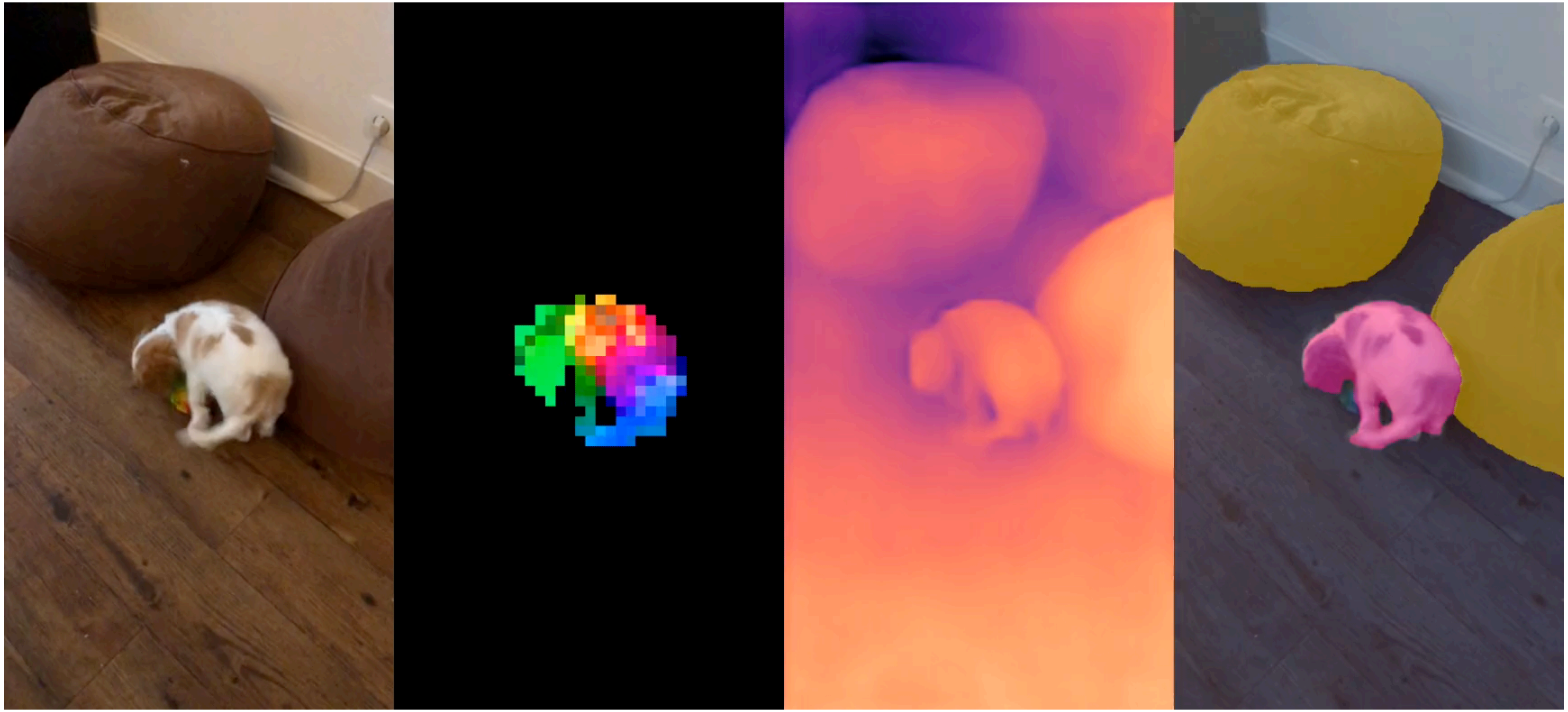
- x2 faster and x3 less memory than DINO;
- A combination of existing techniques (e.g. noisy student, adaptive resolution);
- Training one big model, distilling to the smaller networks;
- Careful data curation:



Oquab et al., "DINOv2: Learning Robust Visual Features without Supervision " (2023).

DINOv2

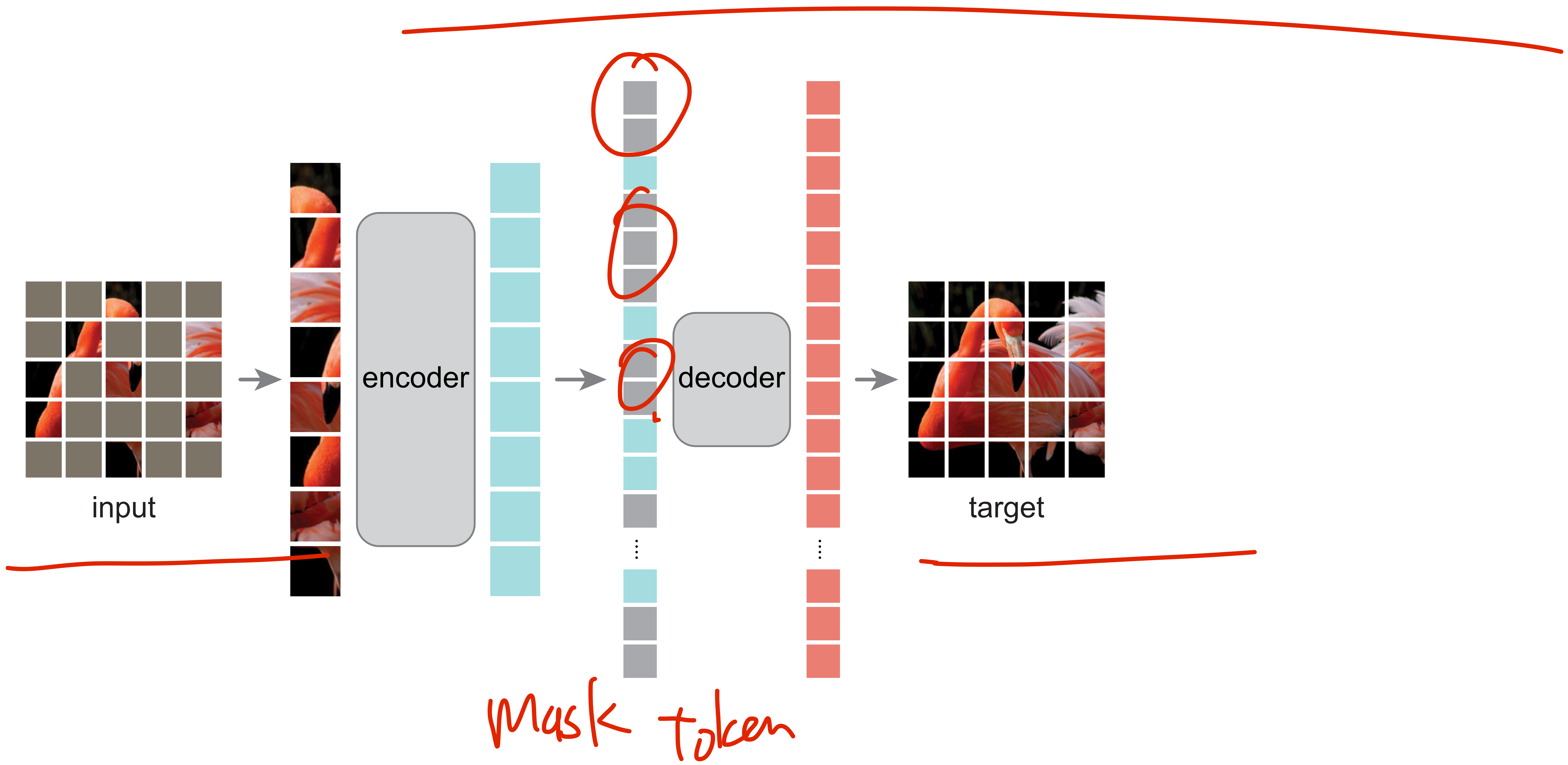
Input frames | Part segmentation | Depth | Semantic segmentation



Oquab et al., “DINOv2: Learning Robust Visual Features without Supervision ” (2023).

Masked Autoencoders

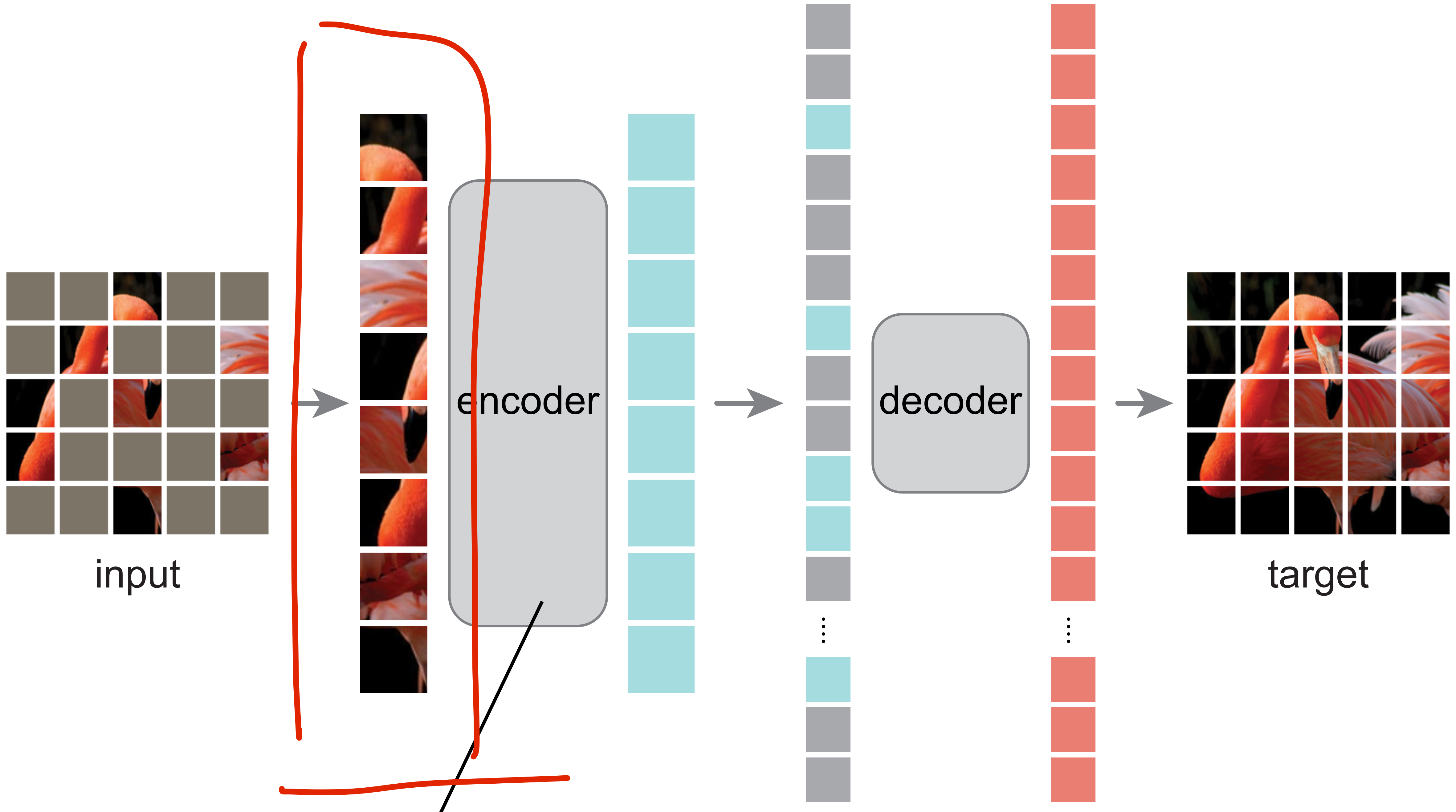
Unsupervised learning with Transformers and a reconstruction loss:



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

Masked Autoencoders

Unsupervised learning with Transformers and a reconstruction loss:

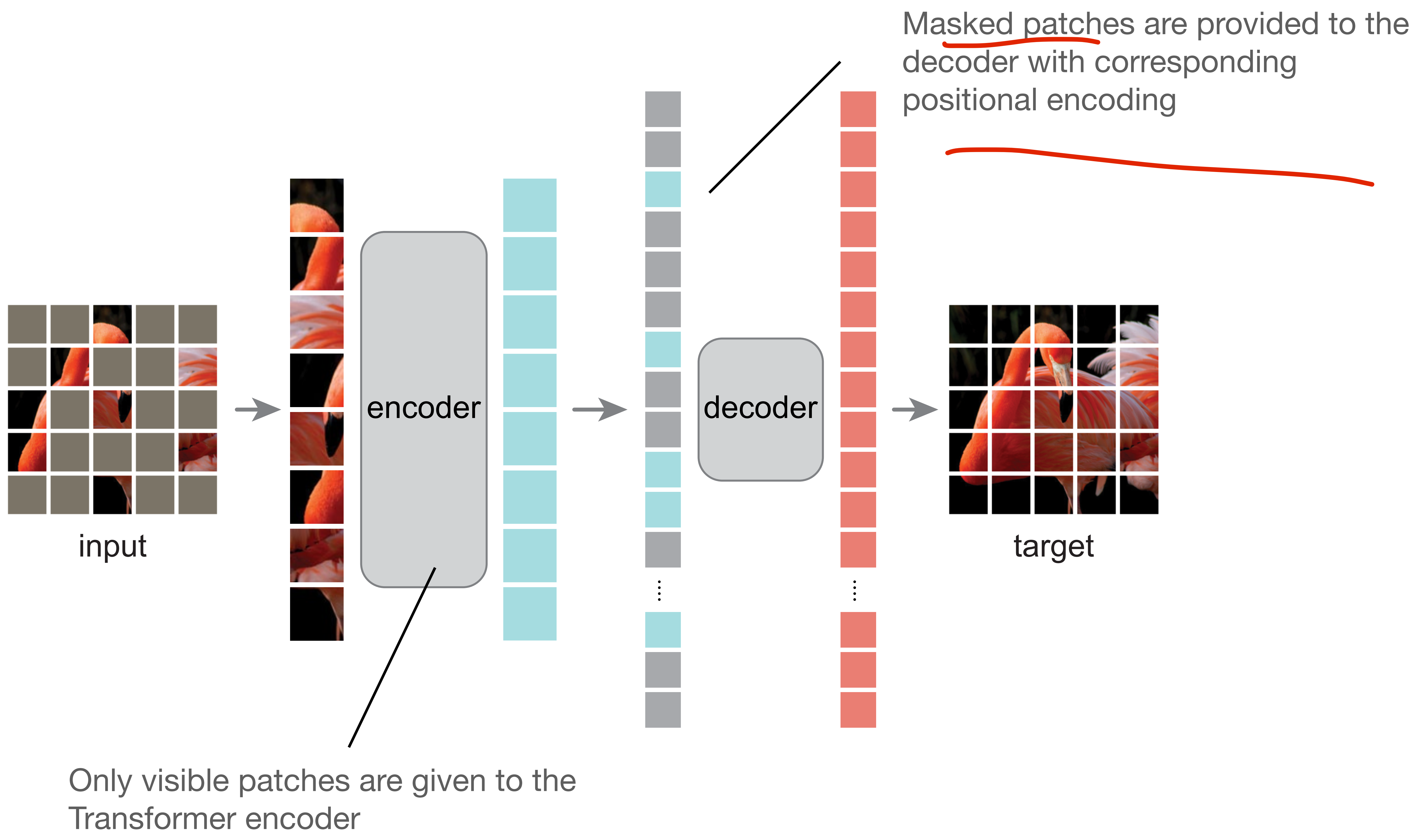


Only visible patches are given to the
Transformer encoder

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

Masked Autoencoders

Unsupervised learning with Transformers and a reconstruction loss:

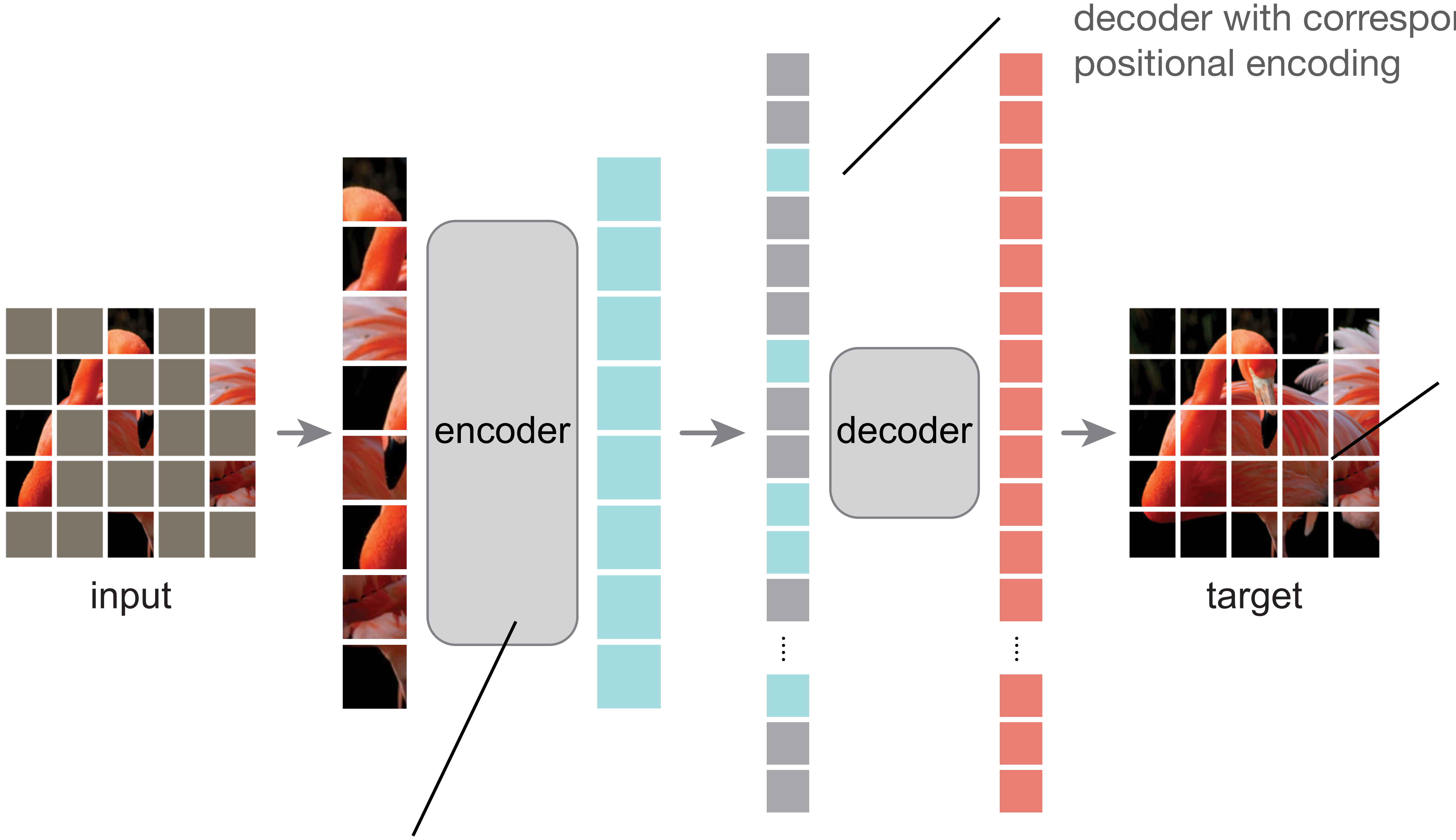


He et al., “Masked Autoencoders Are Scalable Vision Learners”. In CVPR 2022.

Masked Autoencoders

Unsupervised learning with Transformers and a reconstruction loss:

Masked patches are provided to the decoder with corresponding positional encoding



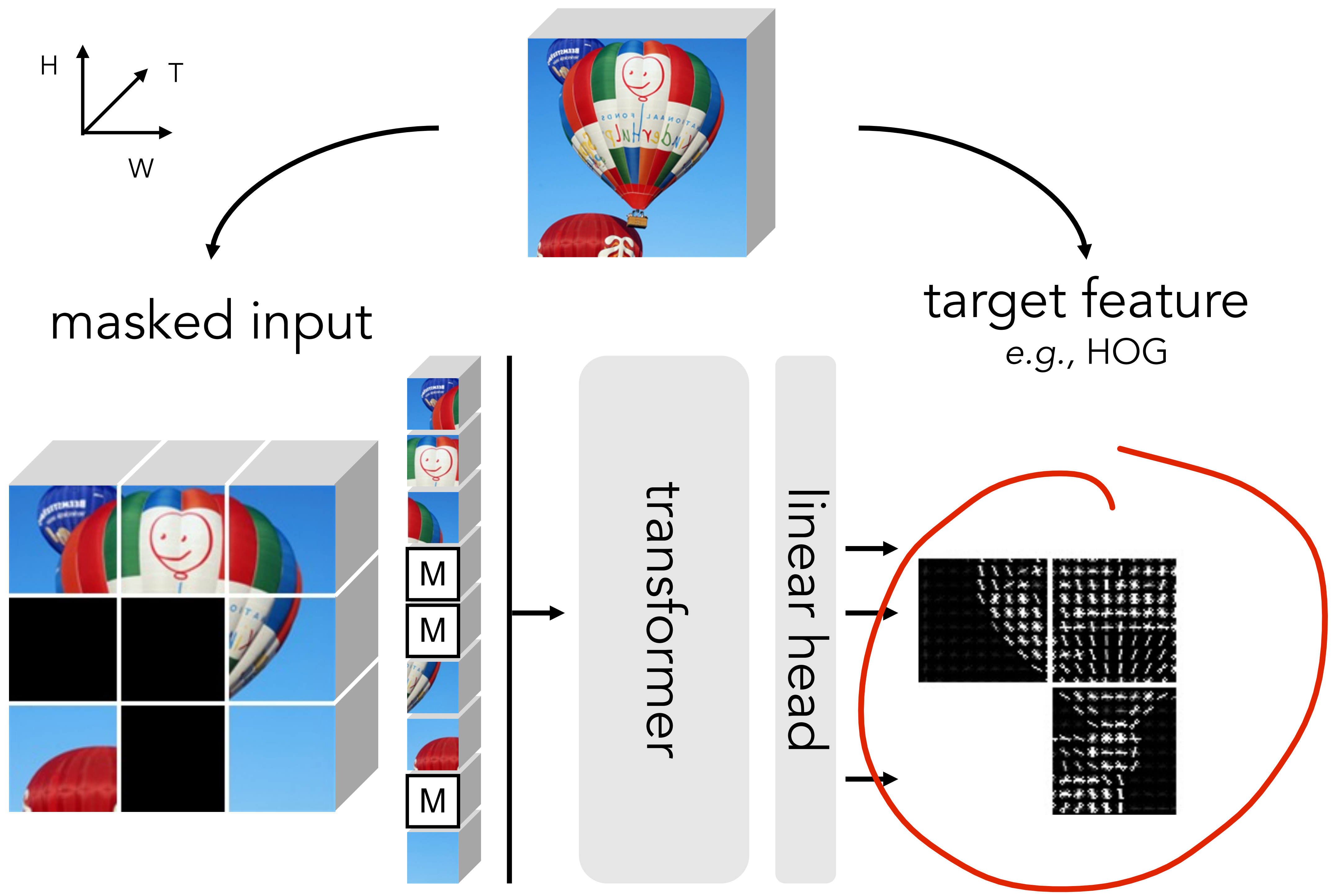
Simple L2 reconstruction loss is already sufficient

Only visible patches are given to the Transformer encoder

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

Masked Feature Prediction

Reconstructing HoG features, instead of pixel values:



Wei et al., “Masked Feature Prediction for Self-Supervised Visual Pre-Training”, (2022).

Remarks

- High masking ratio is necessary (75% and more).
- Compute the loss only on the patches masked out in the input.
  - this is different from denoising autoencoders.
- Reconstruction w.r.t. normalised pixel values:
  - compute the mean and deviation of pixels within each patch.
- Why does this work?
  - “this behavior occurs by way of a rich hidden representation inside the MAE”

Remarks

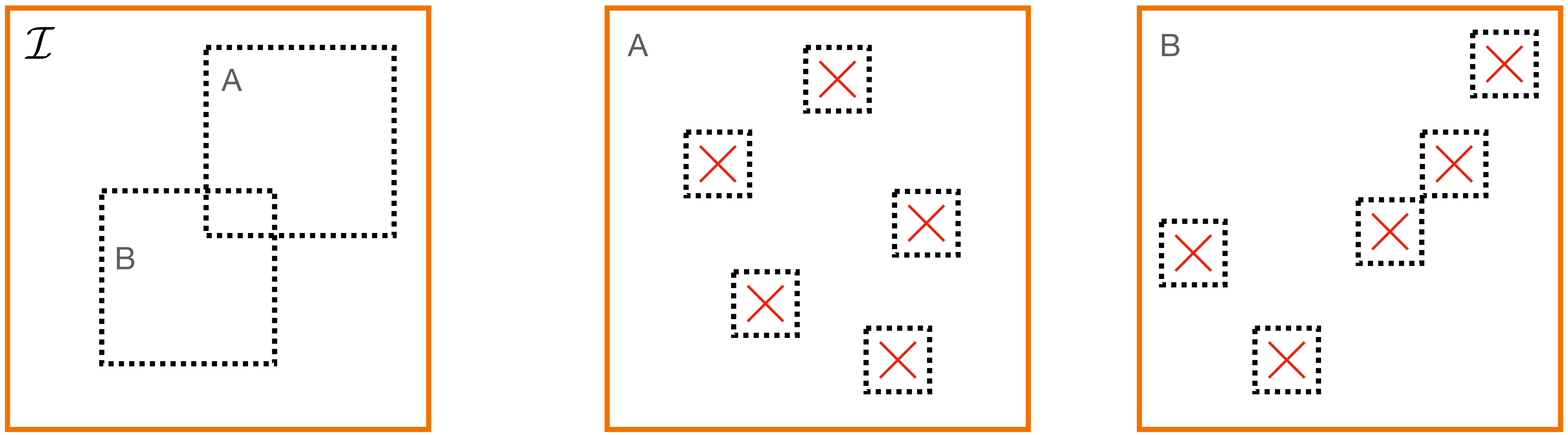
- Why does this work?
- Intuition: The goal is not much different from contrastive learning!

Creating views with crops
(e.g. in contrastive learning)

Masking

*Multiview assumption**:
Either view (crop) provides
enough information for
a downstream task

The assumption is the same
in contrastive learning
and MAEs!



$f(A) \approx f(B) \approx \mathcal{I}$

$f(A) \approx \mathcal{I}$

$f(B) \approx \mathcal{I}$

So, $f(A) \approx f(B) \approx \mathcal{I}$

*Shwartz-Ziv and LeCun, "To Compress or Not to Compress - Self-Supervised Learning and Information Theory: A Review" (2023)

Unsupervised Learning: Downstream Applications

Computer Vision III:
Unsupervised learning

Nikita Araslanov
09.01.2024