Representation Learning

Representation Learning

- Techniques that transform a form of raw data into a representation that can be effectively exploited for machine learning tasks

- Representation learning typically refers to learning such a transformation that can generalize across tasks

What are good representations?

- Representation encodes priors about data distribution(s)

  - Smoothness: close inputs map to close outputs
  - Compactness: input dimension >> output dimension
  - Robustness: features are insensitive to input noise
  - Abstraction and invariances -> problem driven

What are representations?

- A representation performs the task of converting an observation in the real world (e.g., an image, a recorded speech signal, a word in a sentence) into a mathematical form (e.g., a vector)

My heart beats as if the world is dropping, you may not feel the love but i do its a heart breaking moment of your life. enjoy the times that we have, it might not sound good but one thing it rhymes it might not be romantic but i think it is great,the best rhyme i've ever heard.

representation

$$[81, 20, 84, 64, 58, 39, 17, 54, \ldots]$$

What are representations?

- The feature vector can be used by other models to produce outputs, e.g.,
  - Classification

feature

$$[81, 20, 84, 64, \ldots] \Rightarrow \text{“CAT”}$$

What are representations?

- The feature vector can be used by other models to produce outputs, e.g.,
  - Reconstruction

$\Rightarrow$

feature

$[81, 20, 84, 64, \dots]$

$\Rightarrow$

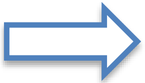
What are representations?

- The feature vector can be used by other models to produce outputs, e.g.,
  - Generation

“a photo of a cat” $\Rightarrow$ feature $[81, 20, 84, 64, \ldots]$ $\Rightarrow$ 

What are representations?

- Representation examples
  - Handcrafted attribute
    - Gender: {"female": 0, "male": 1}
    - Eye color: {"blue": 0, "brown": 1}
    - Hair color: {"black": 0, "blond": 1}



$\boxed{\text{feature}}$

$[1, 0, 1]$

What are representations?

- Representation examples
  - Binary (one-hot vector)
    - {"Paris": 0, "London": 1, "Munich": 2, …}

"Munich" $\Rightarrow$ feature $[0, 0, 1]$

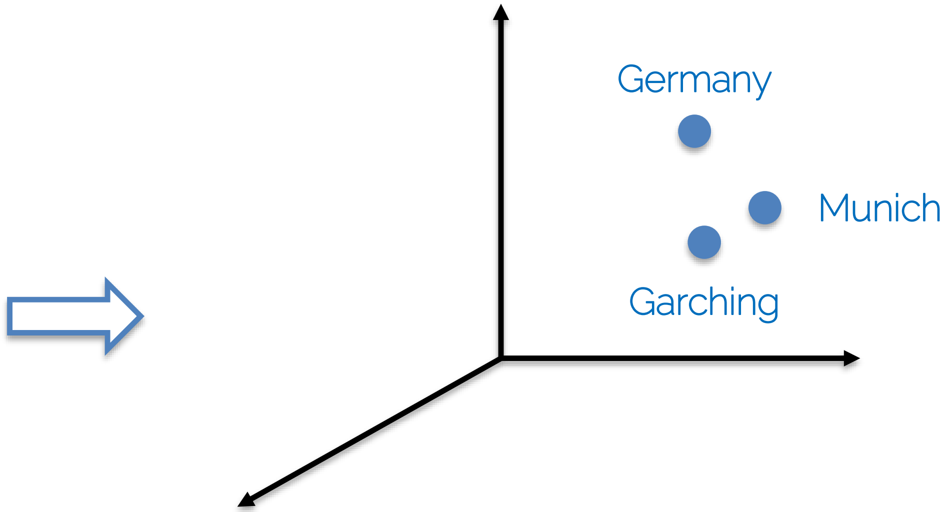
What are representations?

- Representation examples
  - Embedding vector

“Germany”

“Munich”

“Garching”



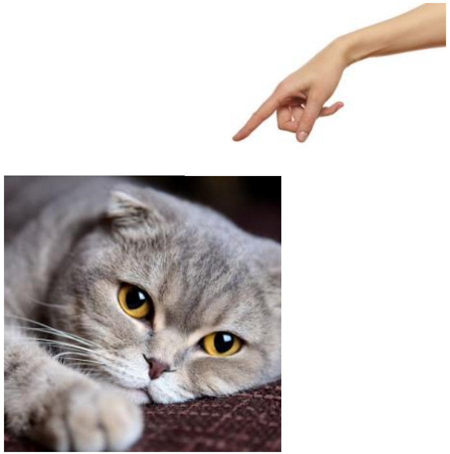
Embedding space

Representation in Computer Vision

⇒ get features vectors

Supervised

Constrained on task(s), e.g., classification



"CAT"

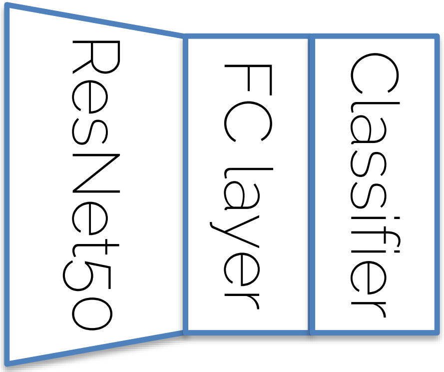
Unsupervised

Constrained on data itself, e.g., reconstruction

Supervised Approaches

- Classification
  - Train ResNet50 on ImageNet
  - Use the features in the last layer as image representations

During training:

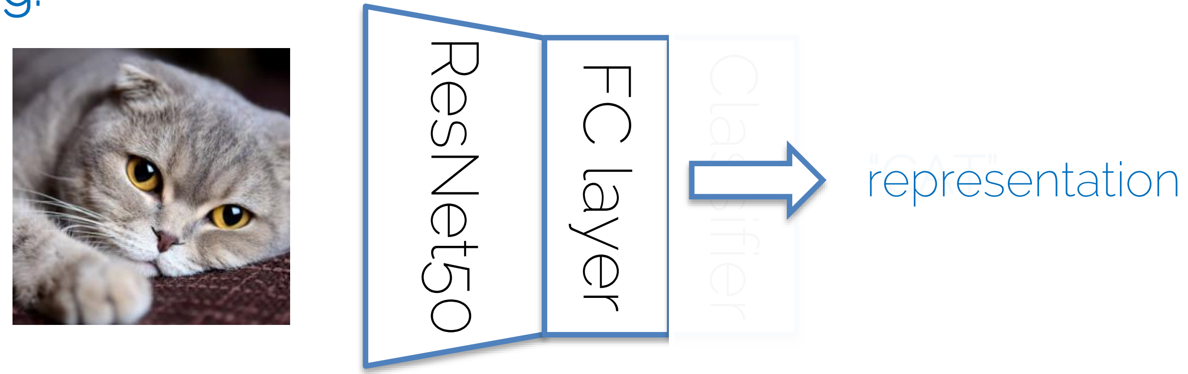


“CAT”

Supervised Approaches

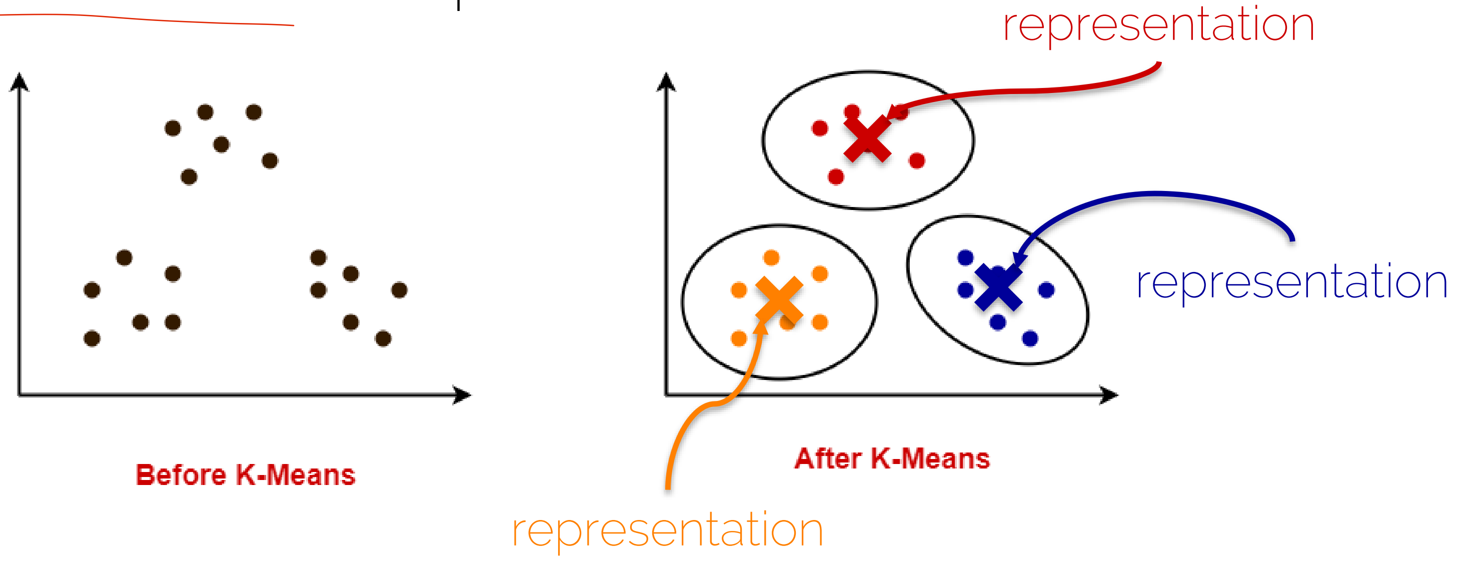
- Classification
  - Train ResNet50 on ImageNet
  - Use the features in the last layer as image representations

After training:



Unsupervised Approaches

- Clustering (K-Means)
  - ~~Mean vectors~~ as representations

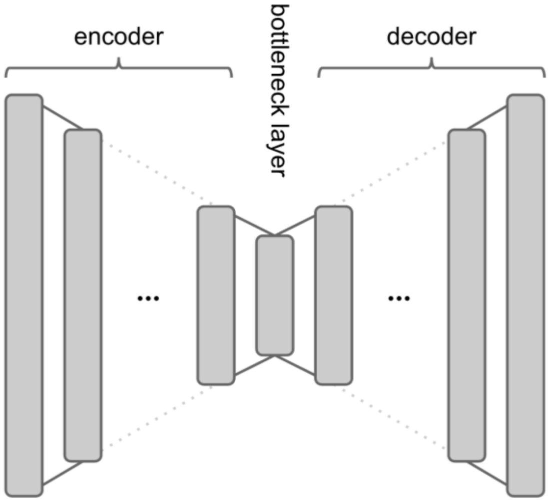


Self-supervised Approaches

- A form of unsupervised learning approaches where the data provides the supervision for itself
- With a proxy loss, e.g., reconstruction loss, the network is forced to learn the features we care about, e.g., semantic representations
- Why self-supervised?
  - Hard and expensive to obtain annotations
  - Alternative to the strong supervisions (labels)

Self-supervised Approaches

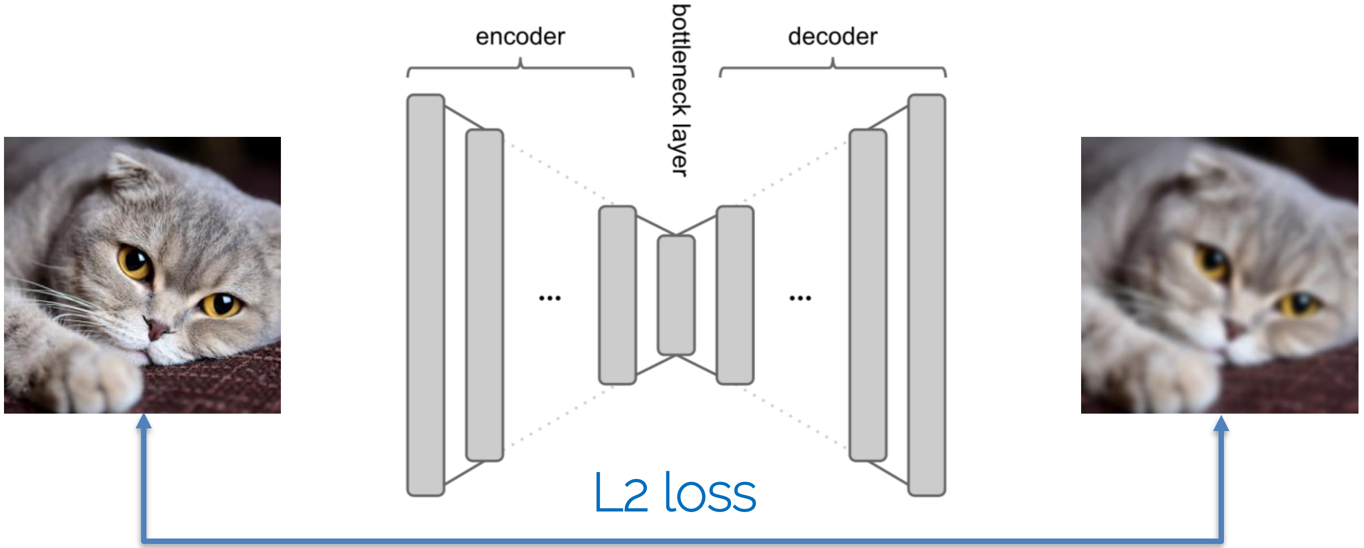
- A form of unsupervised learning approaches where the data provides the supervision for itself



Reconstruction
No label!

Self-supervised Approaches

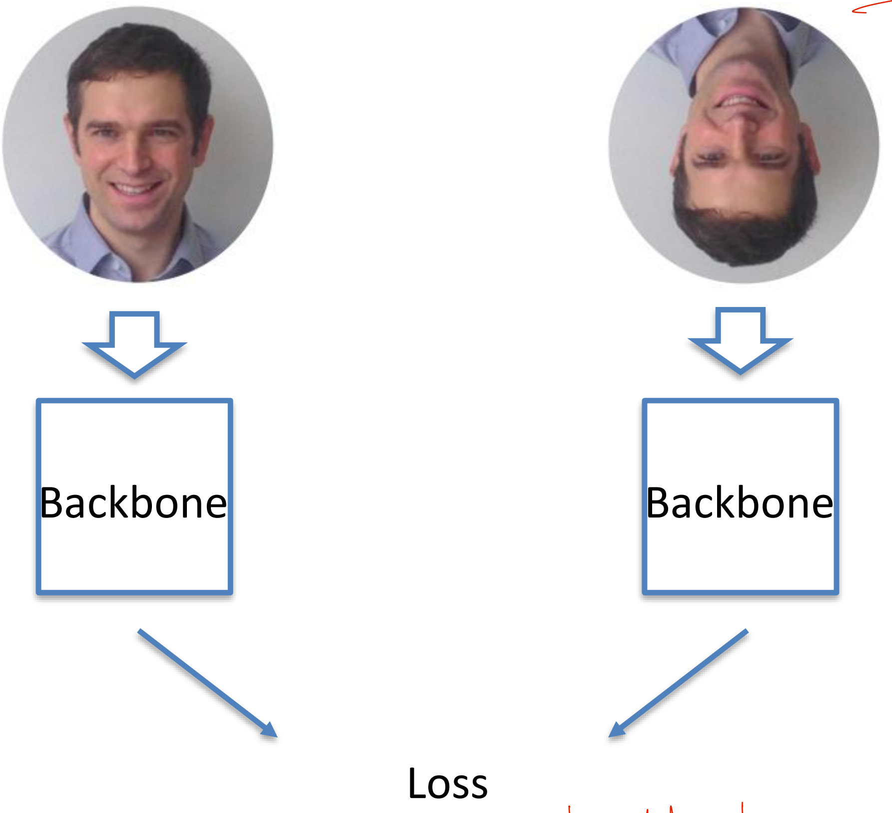
- With a proxy loss, e.g., reconstruction loss, the network is forced to learn the features we care about, e.g., semantic representations



Self-supervised Approaches

- Why self-supervised?
  - Hard and expensive to obtain annotations
  - Make the most out of the existing unlabelled data
    - Instagram: >1 billion images uploaded / day
    - YouTube: >300 hrs of vides uploaded / minute
  - Alternative to the strong supervisions (labels)

Self-supervision by Augmentation



should be same

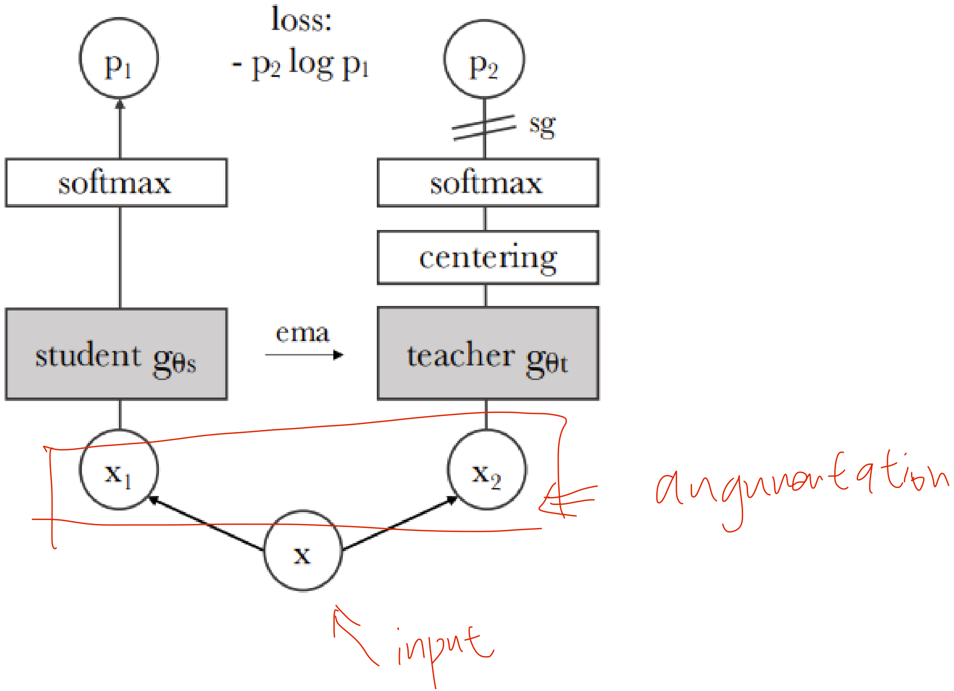
Augmentation is an Art:

- Image vs patch basis
- Color variations
- Geometric transforms
- …

Losses we have already seen some
-> contrastive learning is popular

DINO

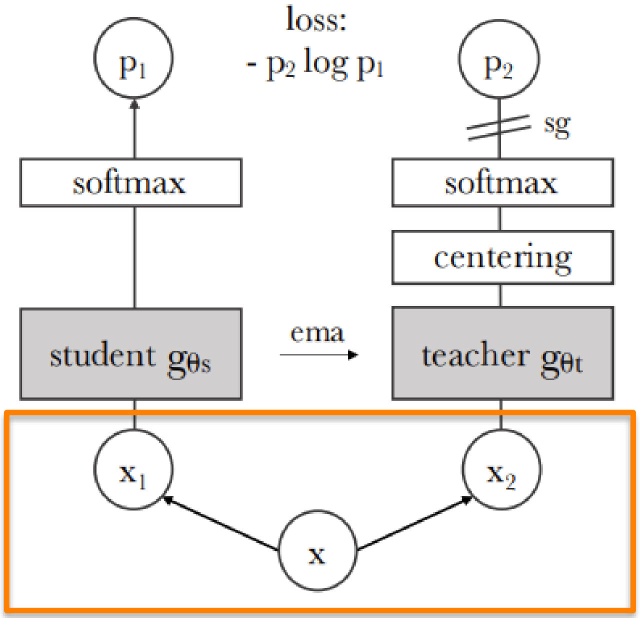
- Self-distillation with no labels



Caron et al. "Emerging Properties in Self-Supervised Vision Transformers".

DINO

- Self-distillation with no labels

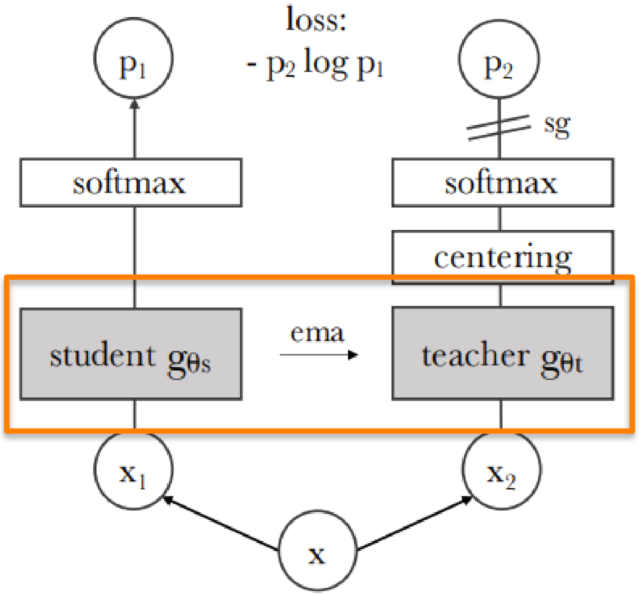


A pair of two random transformations of the image input

Caron et al. “Emerging Properties in Self-Supervised Vision Transformers”.

DINO

- Self-distillation with no labels

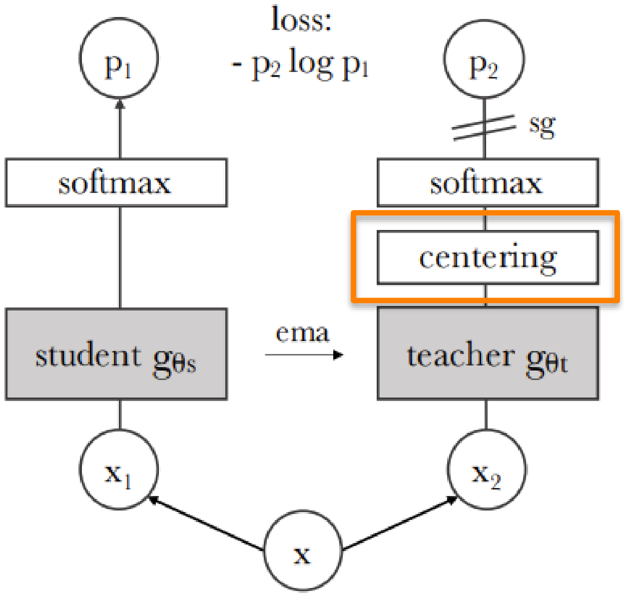


Same image encoder, e.g., ResNet50, but different parameters

Caron et al. “Emerging Properties in Self-Supervised Vision Transformers”.

DINO

- Self-distillation with no labels



Teacher's output is centered over the batch

Caron et al. "Emerging Properties in Self-Supervised Vision Transformers".

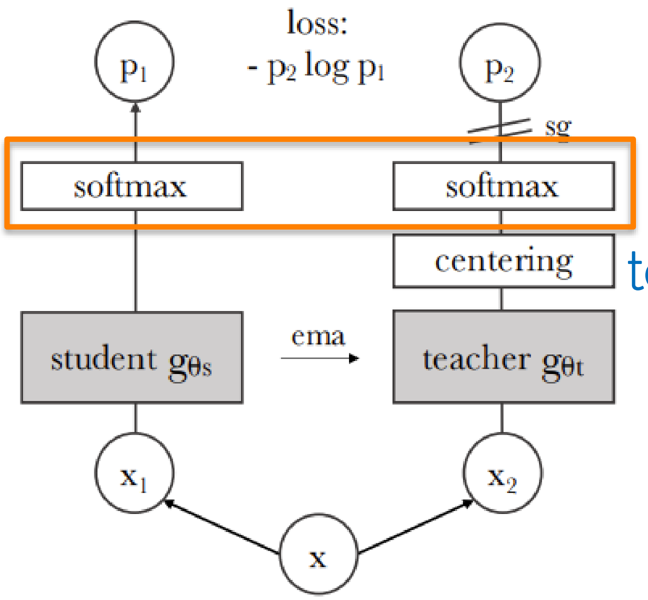
DINO

- Self-distillation with no labels

For a batch with K features

$$P_s(x)^{(i)} = \frac{\exp(\boxed{g_{\theta_s}(x)^{(i)}}/\tau_s)}{\sum_{k=1}^{K} \exp(g_{\theta_s}(x)^{(k)}/\tau_s)}$$

Network outputs



Network outputs are normalized by a temperature softmax over the feature dimension

Caron et al. "Emerging Properties in Self-Supervised Vision Transformers".

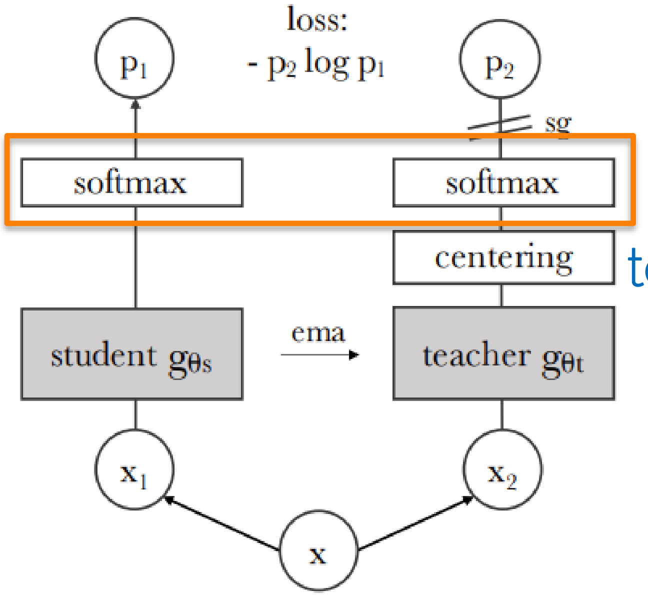
DINO

- Self-distillation with no labels

For a batch with K features

$$P_s(x)^{(i)} = \frac{\exp(g_{\theta_s}(x)^{(i)} \boxed{/\tau_s})}{\sum_{k=1}^{K} \exp(g_{\theta_s}(x)^{(k)}/\tau_s)}$$

Temperature



Network outputs are normalized by a temperature softmax over the feature dimension

Caron et al. “Emerging Properties in Self-Supervised Vision Transformers”.

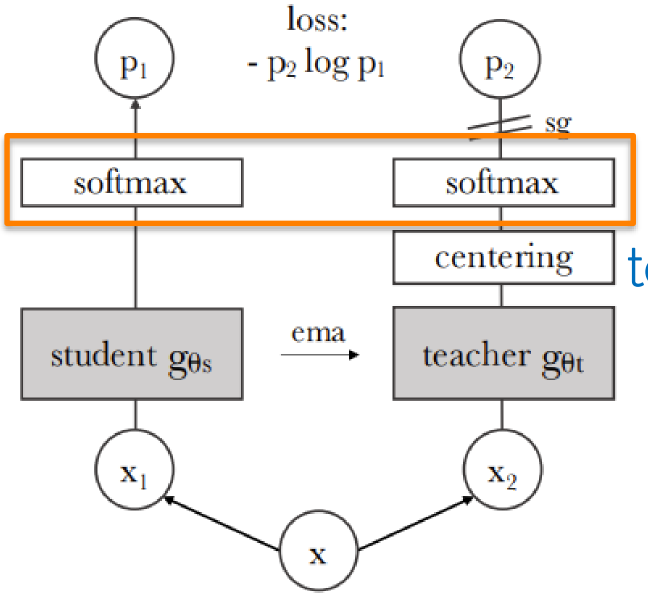
DINO

- Self-distillation with no labels

For a batch with K features

$$\boxed{P_s(x)^{(i)}} = \frac{\exp(g_{\theta_s}(x)^{(i)}/\tau_s)}{\sum_{k=1}^{K}\exp(g_{\theta_s}(x)^{(k)}/\tau_s)}$$

Normalized features

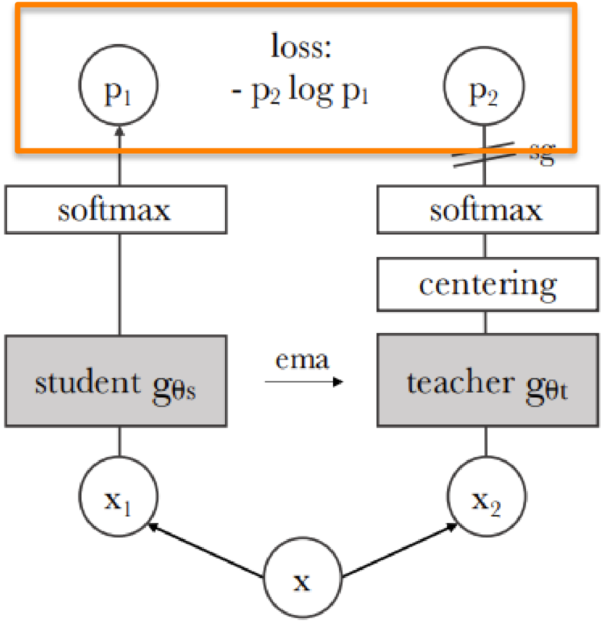


Network outputs are normalized by a temperature softmax over the feature dimension

Caron et al. "Emerging Properties in Self-Supervised Vision Transformers".

DINO

- Self-**di**stillation with **no** labels



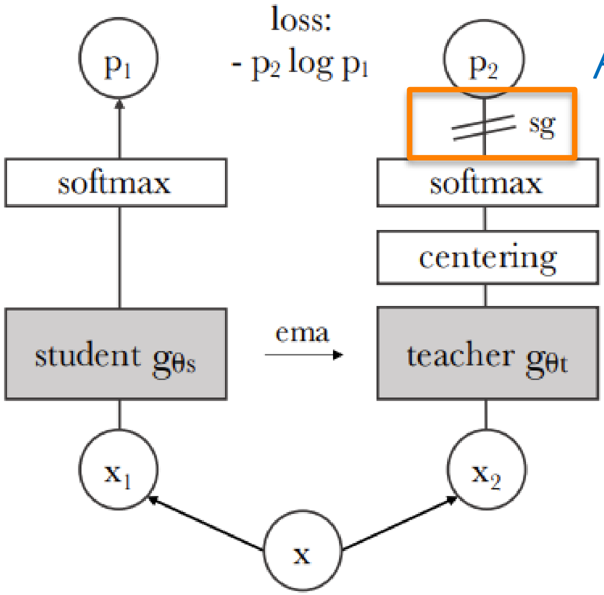
The similarity of two outputs is measured by a cross-entropy loss

Caron et al. "Emerging Properties in Self-Supervised Vision Transformers".

DINO

- Self-distillation with no labels

only train
student



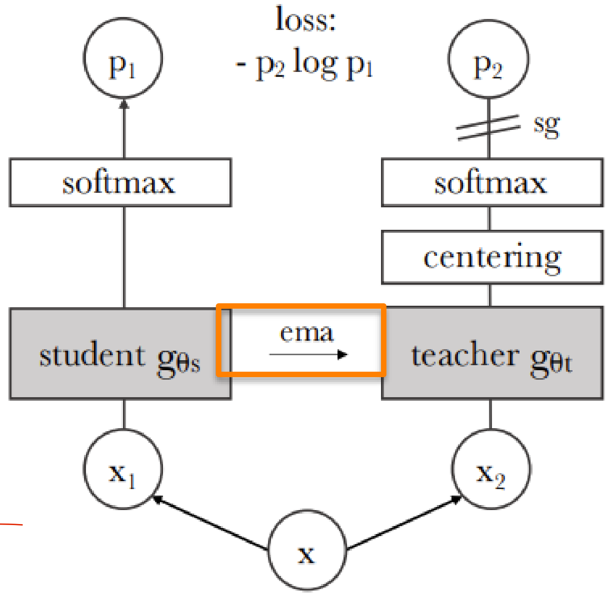
A stop-gradient operator is applied on the teacher to back-prop gradients only through the student

Caron et al. "Emerging Properties in Self-Supervised Vision Transformers".

DINO

- Self-distillation with no labels

The teacher is essentially built from past iterations of the student



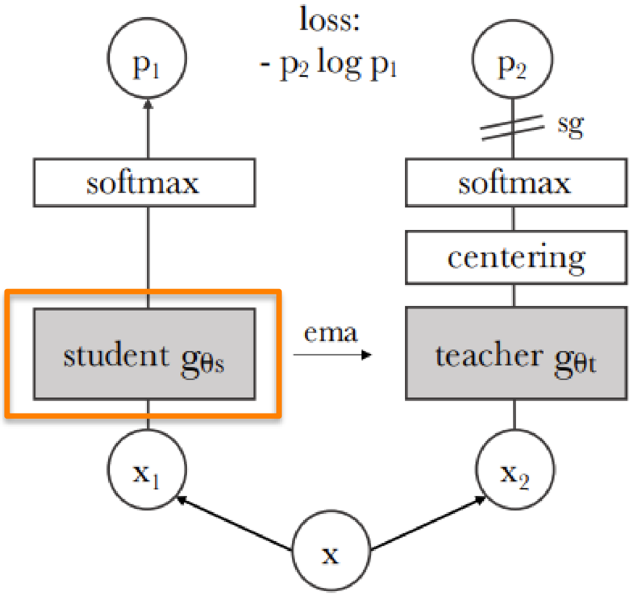
The teacher parameters are updated with an exponential moving average of the student parameters

Caron et al. "Emerging Properties in Self-Supervised Vision Transformers".

DINO

- Self-**di**stillation with **no** labels

The trained student network is used for feature extraction



Caron et al. "Emerging Properties in Self-Supervised Vision Transformers".

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”.

DINO

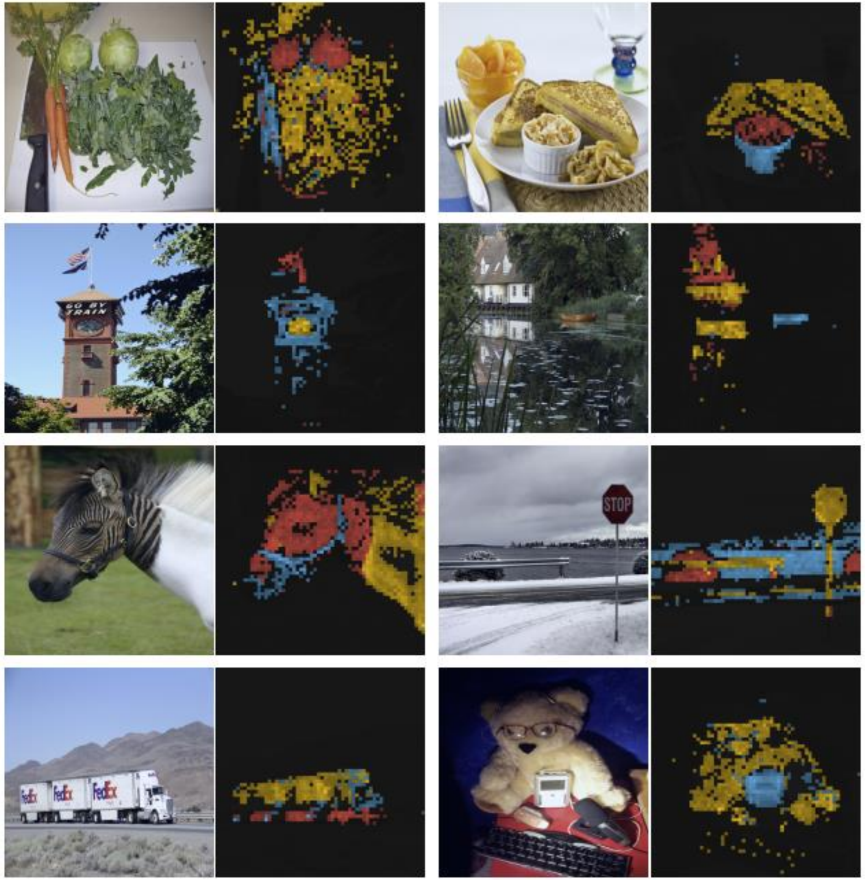
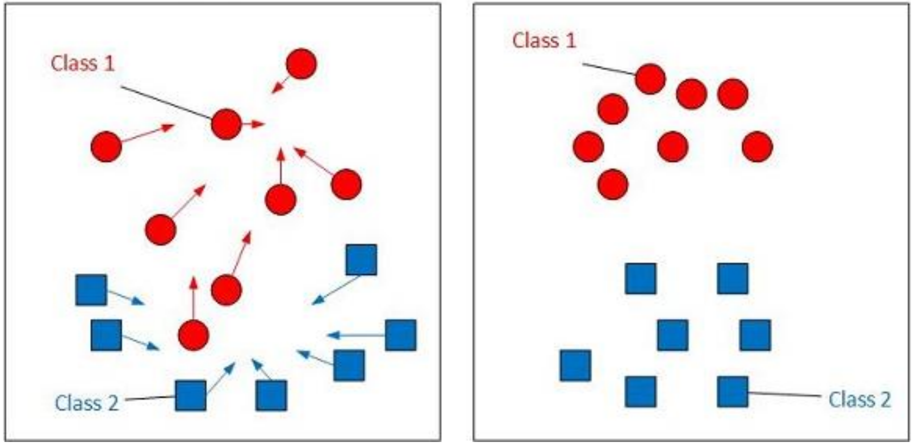


Figure 3: **Attention maps from multiple heads.** We consider the heads from the last layer of a ViT-S/8 trained with DINO and display the self-attention for `[CLS]` token query. Different heads, materialized by different colors, focus on different locations that represents different objects or parts (more examples in Appendix).

Caron et al. "Emerging Properties in Self-Supervised Vision Transformers".

Contrastive Learning Approaches

- What is contrastive learning?
  - To learn an embedding space in which similar samples pairs stay close while dissimilar ones repel



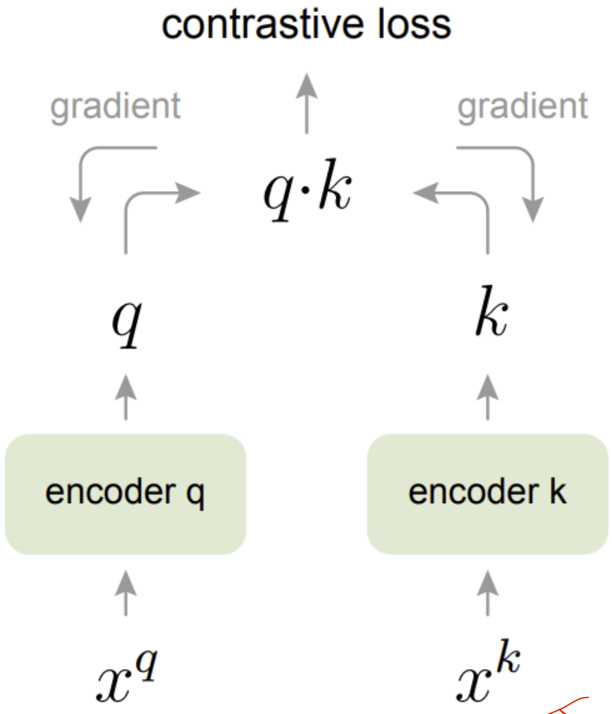
Contrastive Learning Approaches

- Can be both supervised and unsupervised
  - With labels? Without labels?

- Can be even used in semi-supervised setting -> some samples are annotated, others not

- When working with unsupervised data, it is one of the most powerful approaches in self-supervised setting

Contrastive Learning Approaches



(a) end-to-end

$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i / \tau)}$$

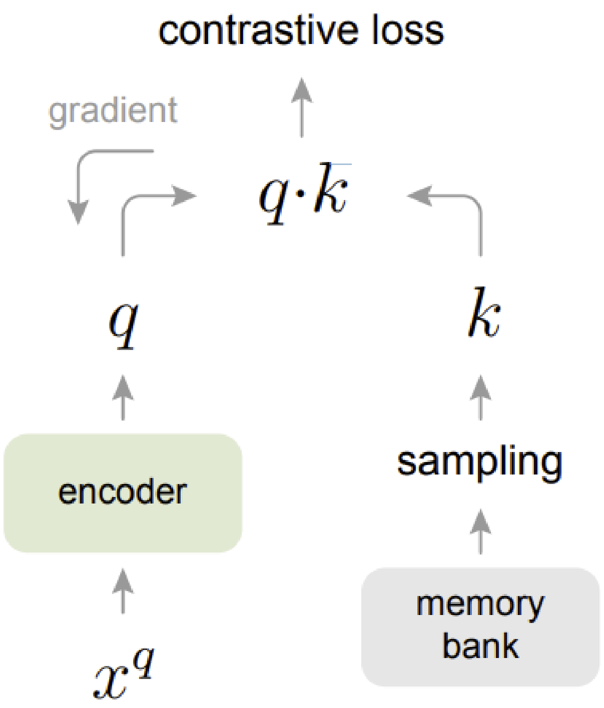
InfoNCE Loss: (k+1)-way softmax classifier

run k times

image

Issue: k is coupled to the mini-batch size which limits k by GPU memory

Contrastive Learning Approaches

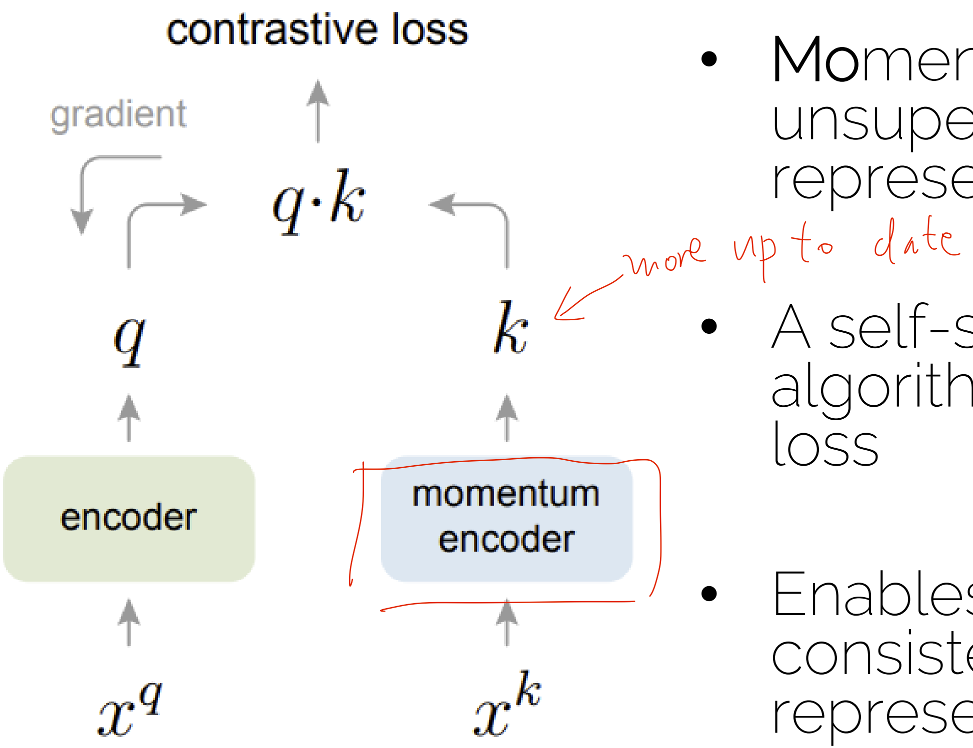


(b) memory bank

Idea: don't update keys at the same time but compare to encodings from memory bank

-> allows for large k but encodings are not up to date (typically once per epoch)

Contrastive Learning Approaches



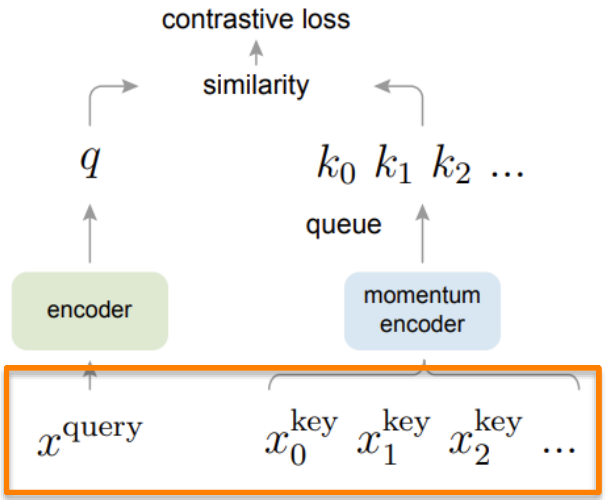
(c) MoCo

- **Mo**mentum **Co**ntrast for unsupervised visual representation learning
- A self-supervised learning algorithm with a contrastive loss
- Enables learning a large and consistent visual representation

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

MoCo

Can be thought of as building a dynamic dictionary

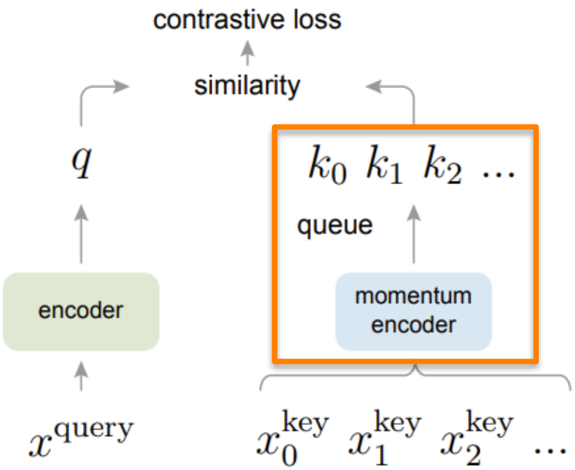


Samples from the dataset

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

MoCo

Can be thought of as building a dynamic dictionary



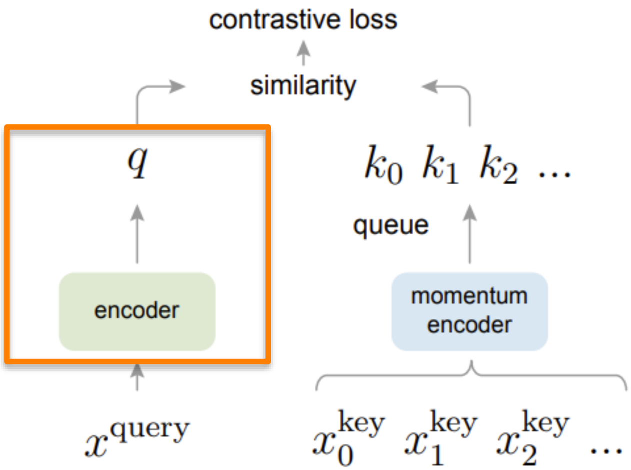
"key": samples encoded on-the-fly by a slowly updating encoder

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

MoCo

Can be thought of as building a dynamic dictionary

"query": samples encoded by another encoder to match the keys in dictionary

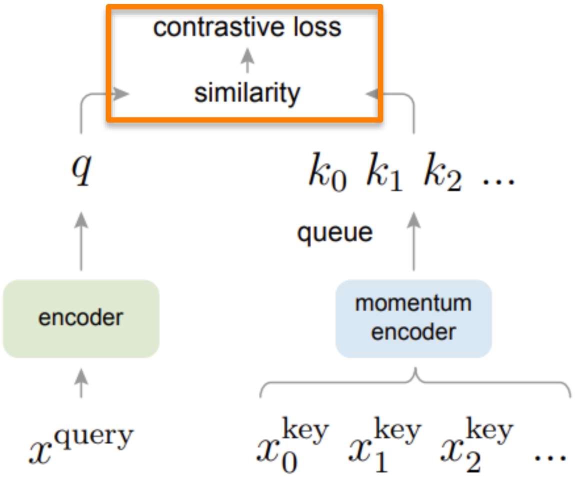


He et al. "Momentum Contrast for Unsupervised Visual Representation Learning".

MoCo

Can be thought of as building a dynamic dictionary

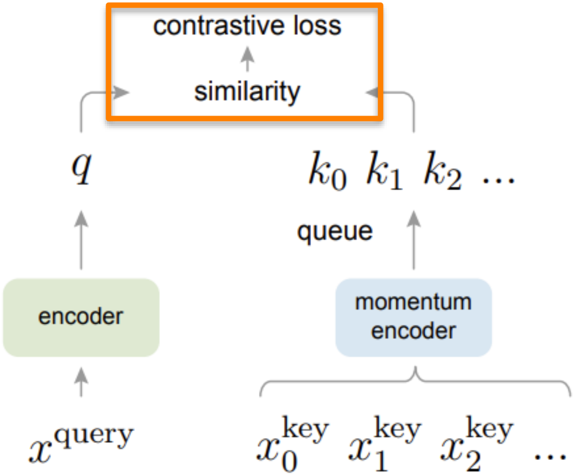
The similarities between the query and keys are supervised by a contrastive loss



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

MoCo

Can be thought of as building a dynamic dictionary



InfoNCE loss function:

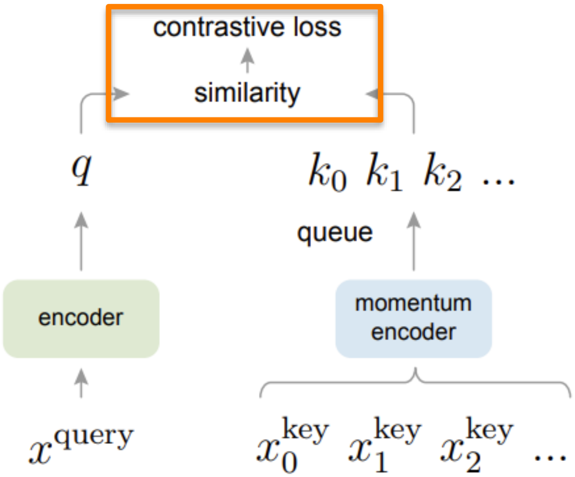
$$\mathcal{L}_q = -\log \frac{\exp(\boxed{q}\cdot k_+/\tau)}{\sum_{i=0}^{K} \exp(q\cdot k_i/\tau)}$$

Query

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

MoCo

Can be thought of as building a dynamic dictionary



InfoNCE loss function:

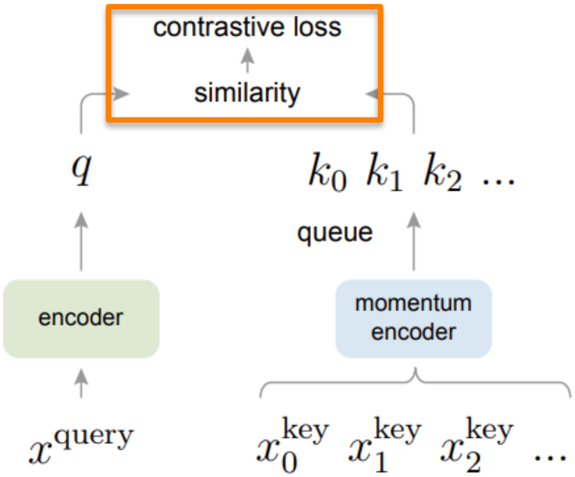
$$\mathcal{L}_q = -\log \frac{\exp(q \cdot \boxed{k_+} / \tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i / \tau)}$$

Matched key

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

MoCo

Can be thought of as building a dynamic dictionary



InfoNCE loss function:

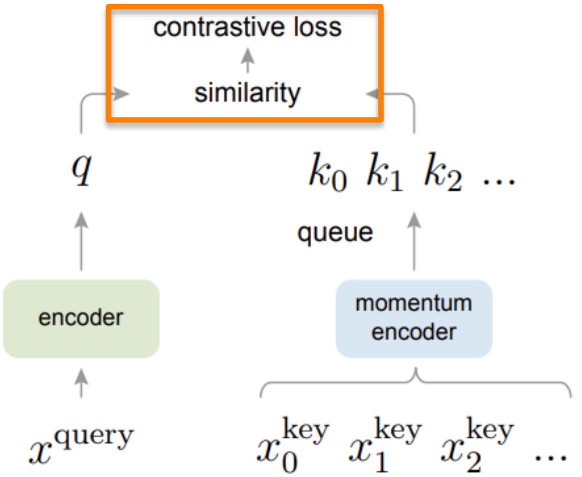
$$\mathcal{L}_q = -\log \frac{\exp(q{\cdot}k_+/\tau)}{\sum_{i=0}^{K} \exp(q{\cdot}\boxed{k_i}/\tau)}$$

All keys

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

MoCo

Can be thought of as building a dynamic dictionary



InfoNCE loss function:

temperature

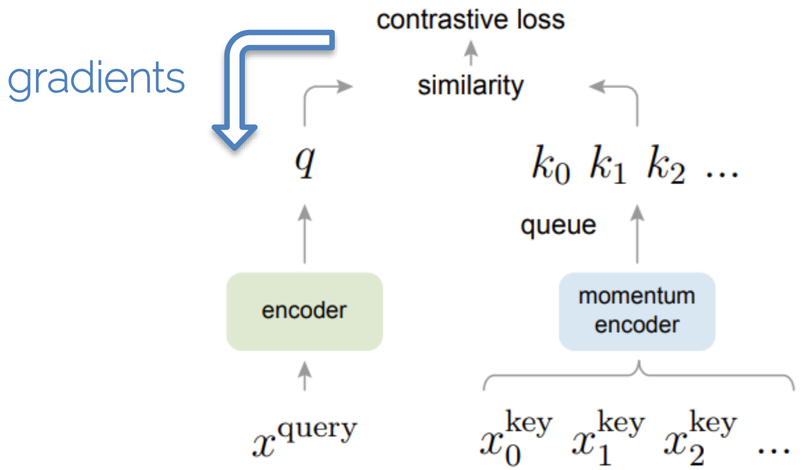
$$\mathcal{L}_q = -\log \frac{\exp(\boxed{q \cdot k_+}\boxed{/\tau})}{\sum_{i=0}^{K} \exp(\boxed{q \cdot k_i}\boxed{/\tau})}$$

Dot products as similarities

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

MoCo

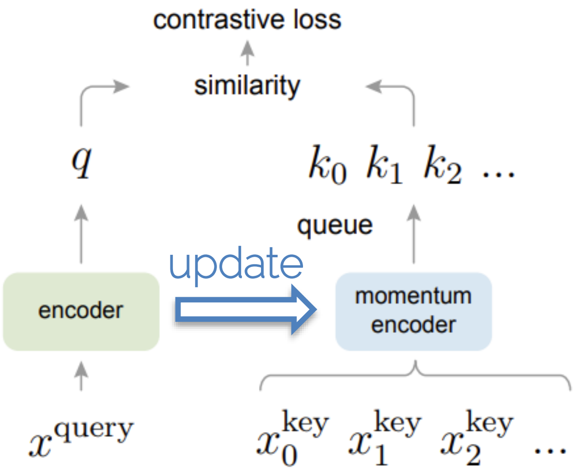
Can be thought of as building a dynamic dictionary



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

MoCo

Can be thought of as building a dynamic dictionary

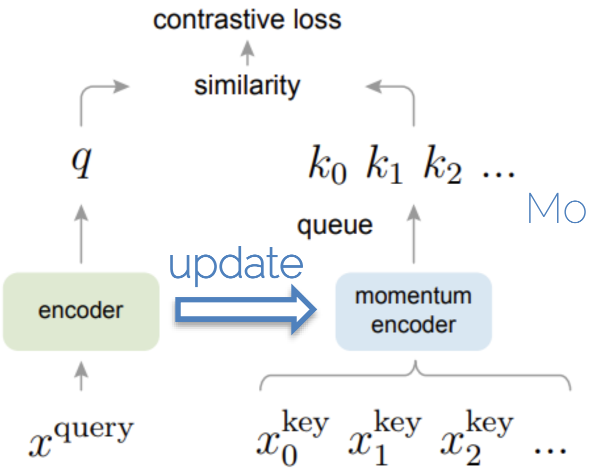


The momentum encoder is driven by a momentum update with the query encoder

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

MoCo

Can be thought of as building a dynamic dictionary



Momentum update:

Momentum encoder parameters

$$\boxed{\theta_{\mathrm{k}}} \leftarrow \boxed{m}\theta_{\mathrm{k}} + (1-m)\boxed{\theta_{\mathrm{q}}}$$

Momentum coefficient

m=0.999 by default

Target encoder parameters

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

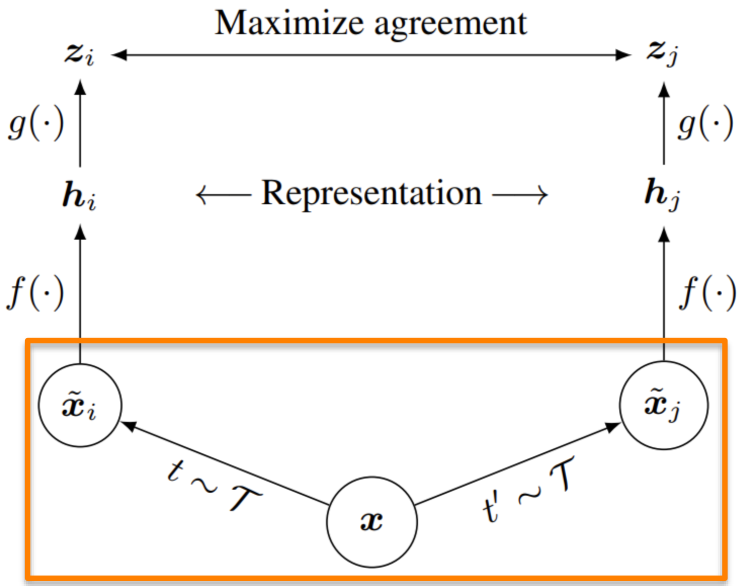
SimCLR

- A **Sim**ple Framework for **C**ontrastive **L**earning of Visual **R**epresentations
- Learns visual representations by maximizing agreement between differently augmented views of the same data samples
- Supervised via a contrastive loss in the latent space

Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations".

SimCLR

- A **Sim**ple Framework for **C**ontrastive **L**earning of Visual **R**epresentations

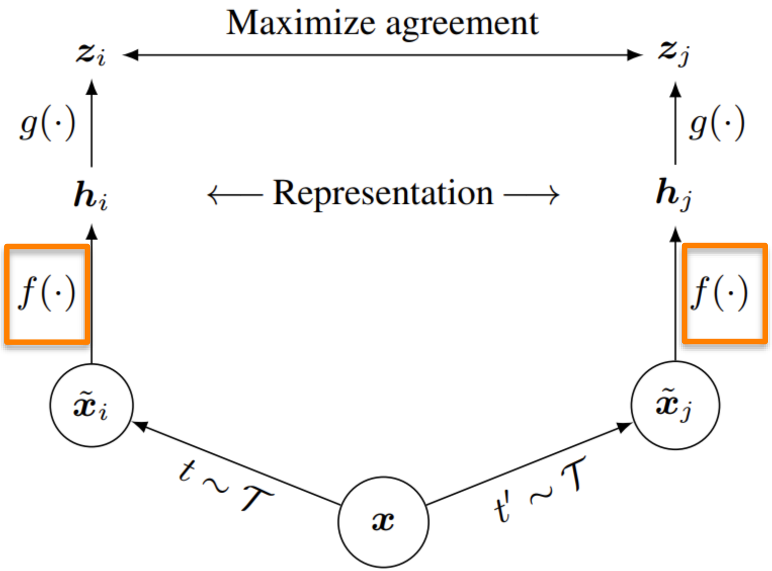


One input sample is augmented to two views by two different operators from the same family, e.g., different rotations

Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations".

SimCLR

- A Simple Framework for Contrastive Learning of Visual Representations

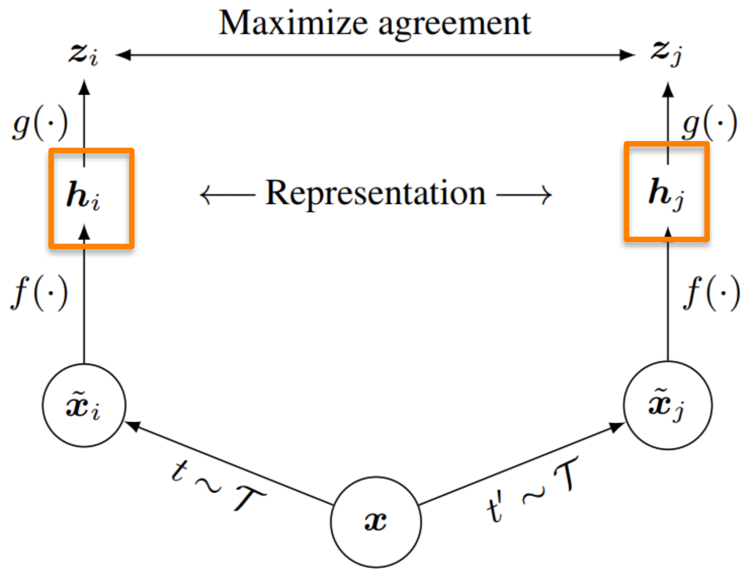


Encoder network, e.g., ResNet50

Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations".

SimCLR

- A **Sim**ple Framework for **C**ontrastive **L**earning of Visual **R**epresentations

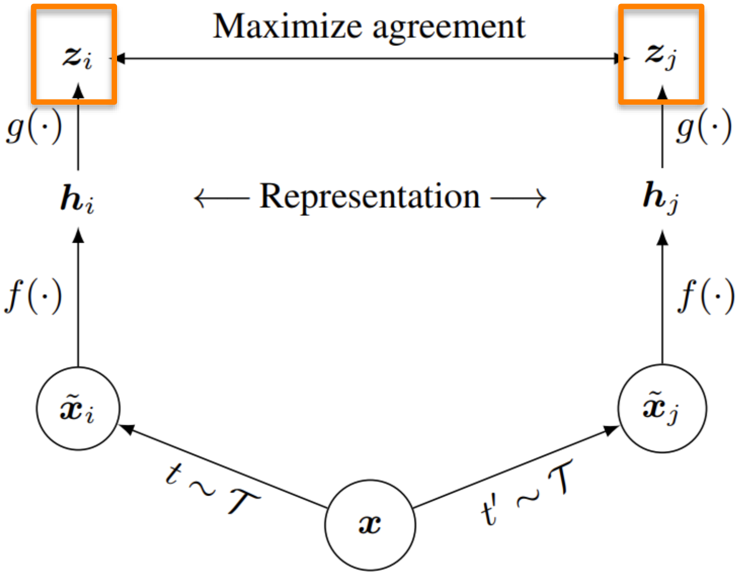


Visual representations

Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations".

SimCLR

- A Simple Framework for Contrastive Learning of Visual Representations



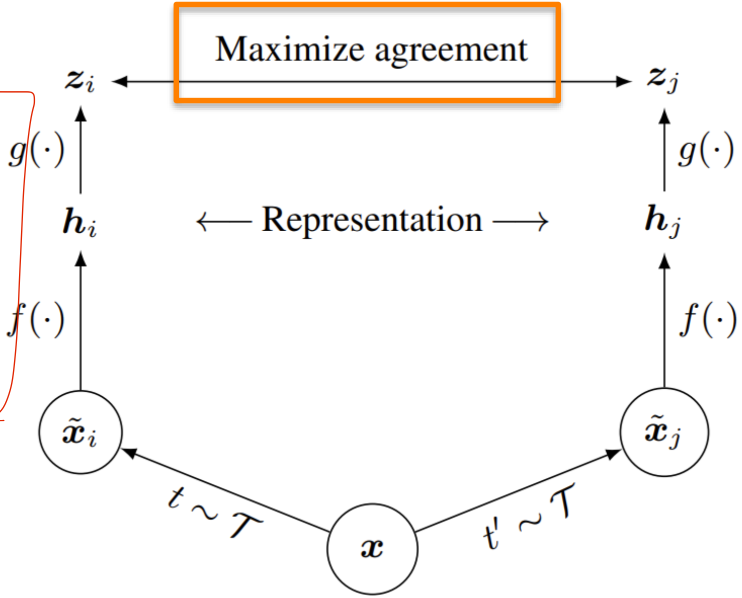
Projected representations

Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations".

SimCLR

- A **Sim**ple Framework for **C**ontrastive **L**earning of Visual **R**epresentations

For a batch with N samples, 2N augmented samples are produced



$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

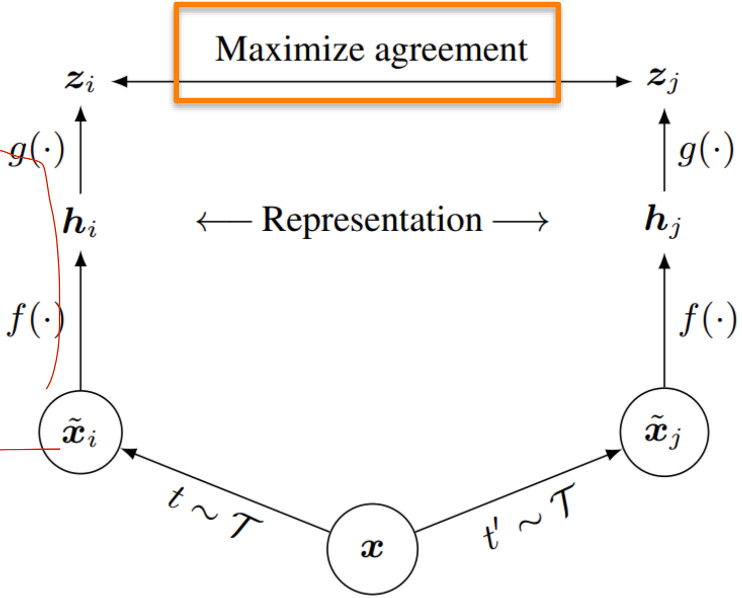
Normalized temperature-scaled cross entropy loss

Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations".

SimCLR

- A **Sim**ple Framework for **C**ontrastive **L**earning of Visual **R**epresentations

Given a positive pair, all other 2(N-1) pairs are treated as negative pairs



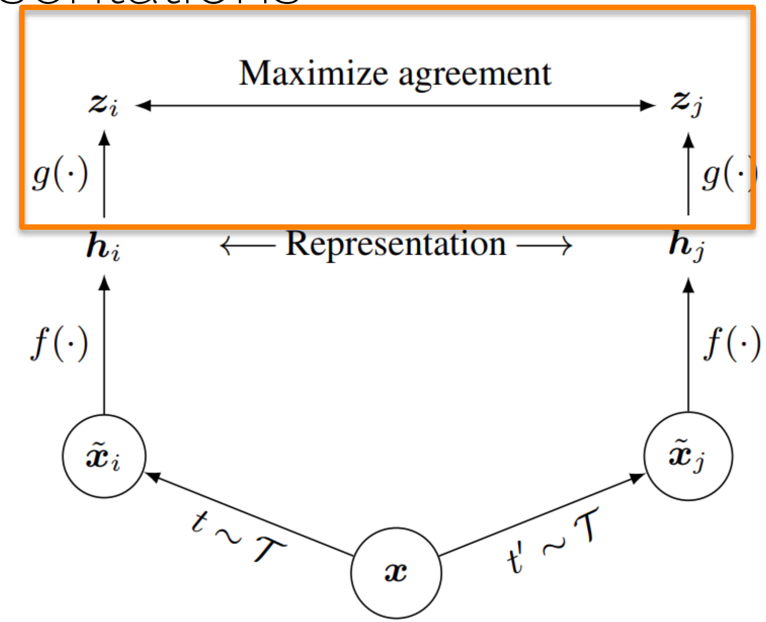
$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \boxed{\mathbb{1}_{[k \neq i]}} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

Set to 1 iff it is a negative pair

Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations".

SimCLR

- A **Sim**ple Framework for **C**ontrastive **L**earning of Visual **R**epresentations



The projection head is removed for downstream tasks

Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations".

SimCLR

- Key takeaways from SimCLR
  - Larger batch (4k or 8k) to provide more negative samples
  - Apply a MLP on the ResNet outputs to encode the final features during training, use the ResNet outputs directly during inference
  - Stronger data augmentations help

Augmentation is an Art

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

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

Augmentations in SimCLR

Multi-Model Representation Learning

- Augmentations are key for contrastive learning: why not use matching samples from different modes?

- Image <-> text is a prime example since there are millions of training pairs on the web

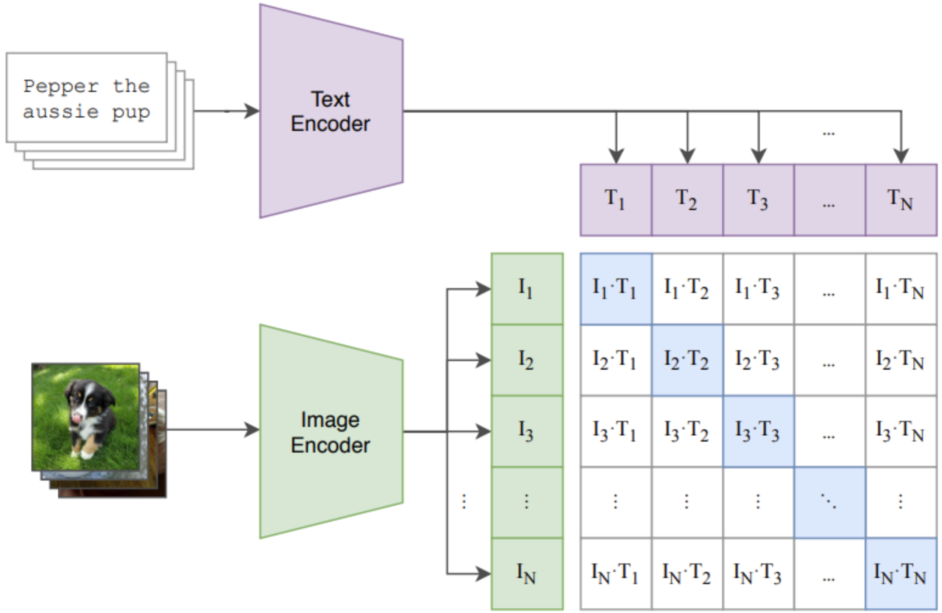
CLIP

- Contrastive Language-Image Pre-training
- Trained on a new dataset of 400 million image-text pairs
- Use a very large batch size of 32,678

CLIP

- Contrastive Language-Image Pre-training

Contrastive pre-training

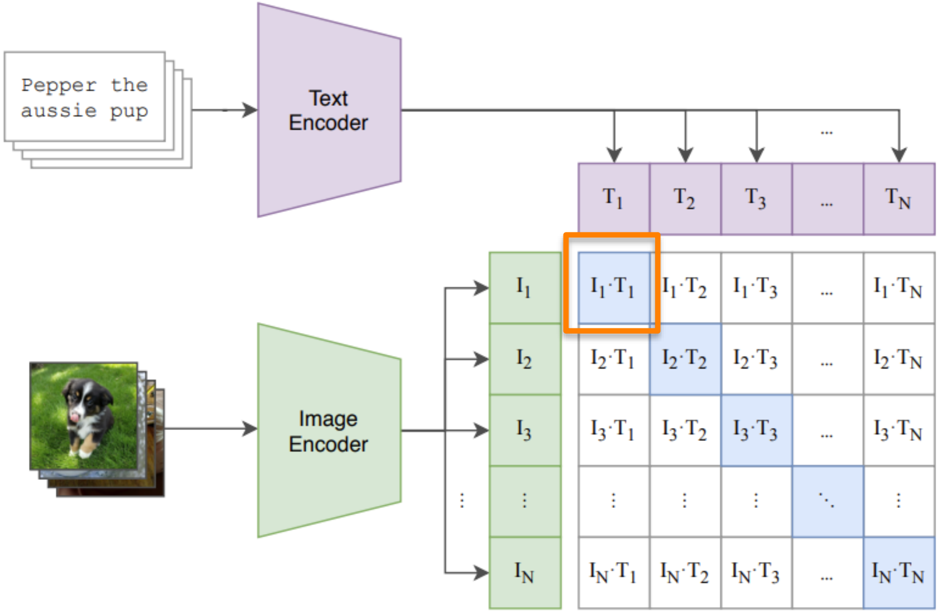


Radford et al. "Learning Transferable Visual Models From Natural Language Supervision".

CLIP

- Contrastive Language-Image Pre-training

Contrastive pre-training



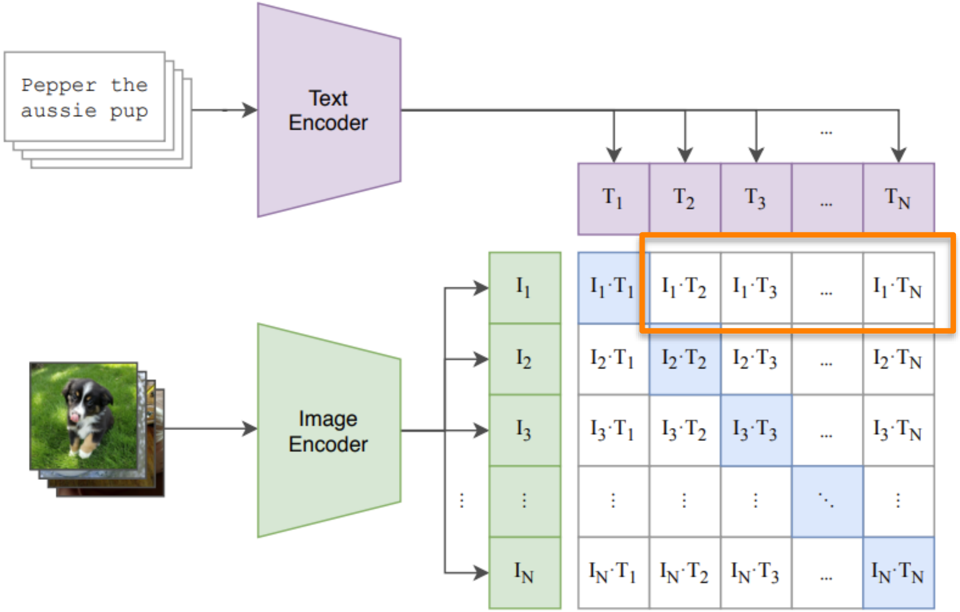
Positive pairs

Radford et al. "Learning Transferable Visual Models From Natural Language Supervision".

CLIP

- Contrastive Language-Image Pre-training

Contrastive pre-training



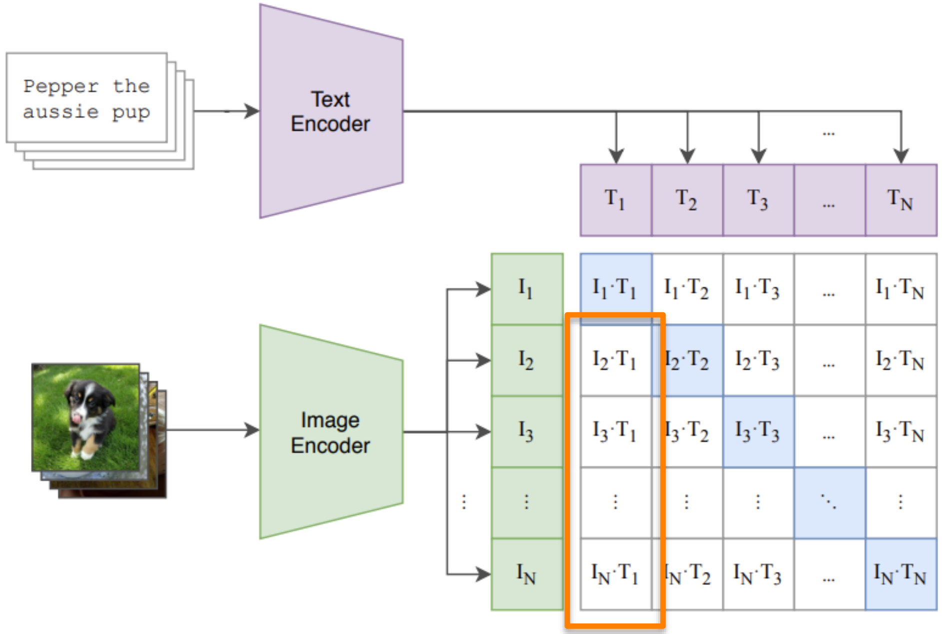
Negative image-text pairs (image as anchor)

Radford et al. "Learning Transferable Visual Models From Natural Language Supervision".

CLIP

- Contrastive Language-Image Pre-training

Contrastive pre-training



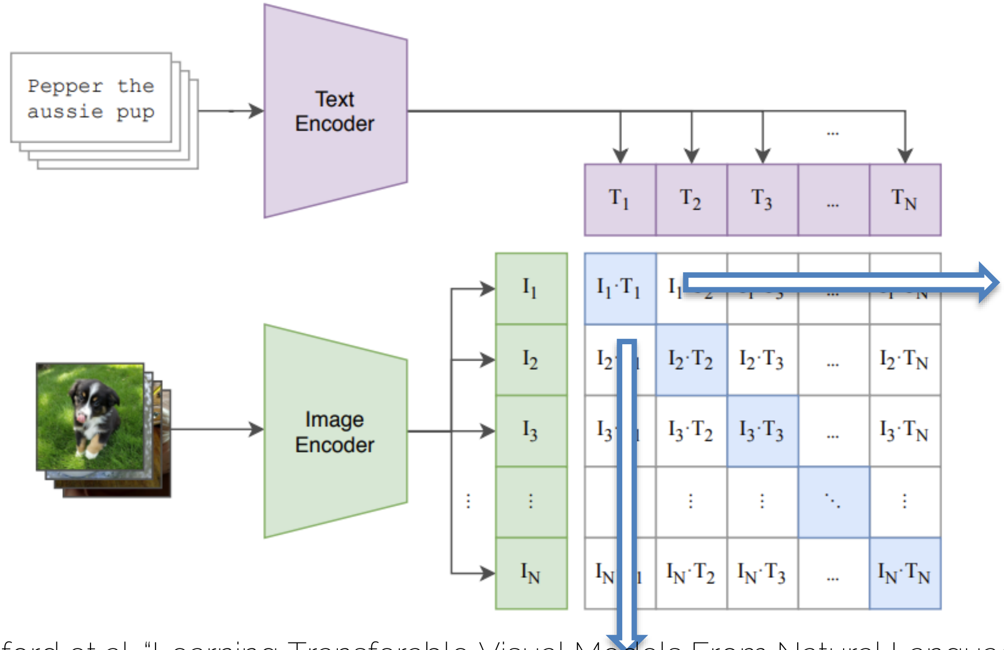
Negative text-image pairs (text as anchor)

Radford et al. "Learning Transferable Visual Models From Natural Language Supervision".

CLIP

- Contrastive Language-Image Pre-training

Contrastive pre-training



Apply cross-entropy loss on both directions

Radford et al. "Learning Transferable Visual Models From Natural Language Supervision".

CLIP

- **C**ontrastive **L**anguage-Image **P**re-training

Contrastive pre-training

```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

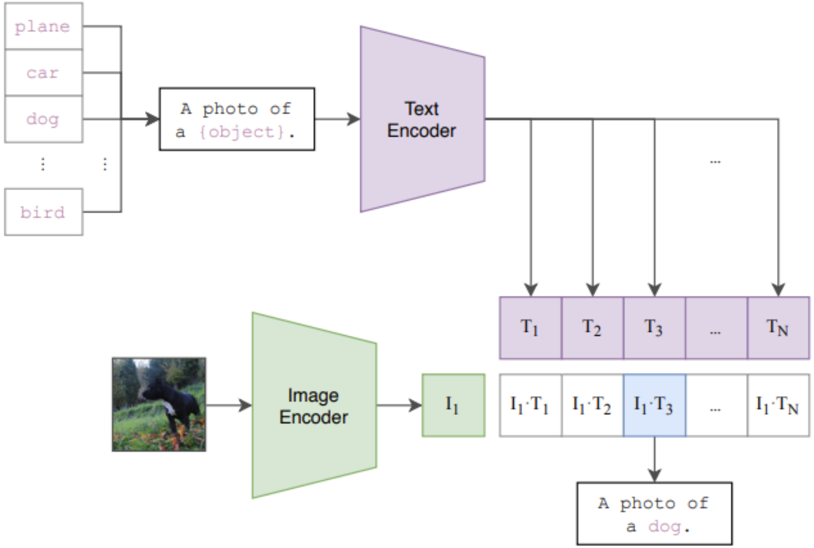
Simple implementation

Radford et al. “Learning Transferable Visual Models From Natural Language Supervision”.

CLIP

- **C**ontrastive **L**anguage-Image **P**re-training

  Inference: zero-shot classification

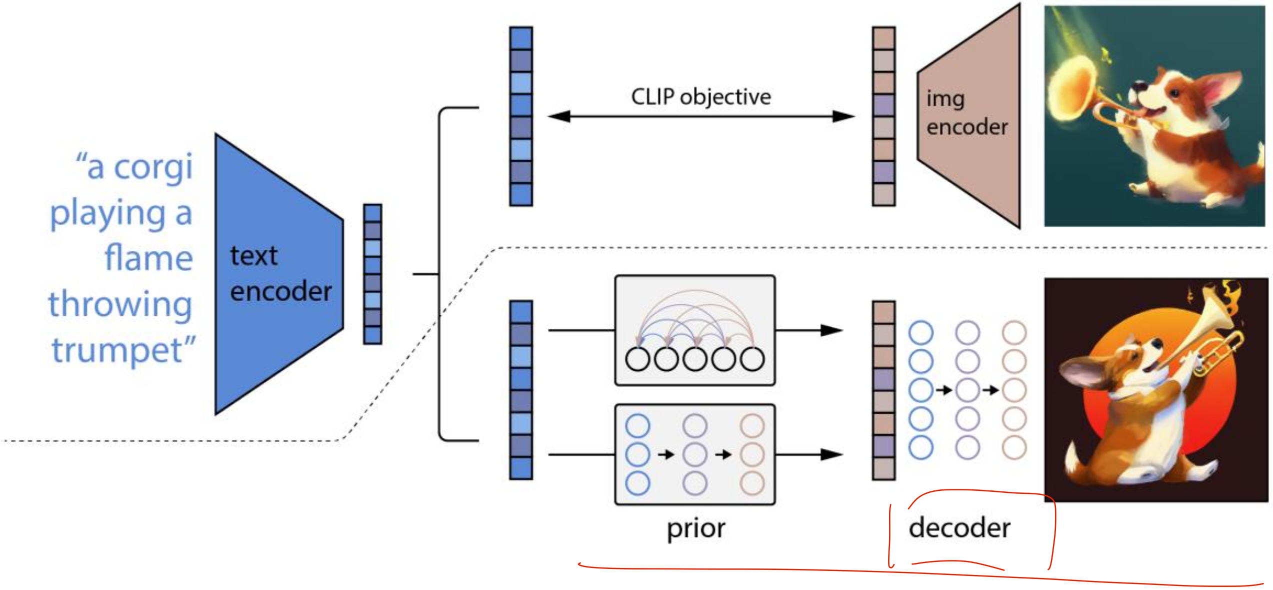


Radford et al. "Learning Transferable Visual Models From Natural Language Supervision".

CLIP

- Contrastive Language-Image Pre-training

  Use case: text-to-image generation (DALL-E 2)



Radford et al. "Hierarchical Text-Conditional Image Generation with CLIP Latents".

CLIP

- Contrastive Language-Image Pre-training

  Use case: text-to-image generation (DALL-E 2)

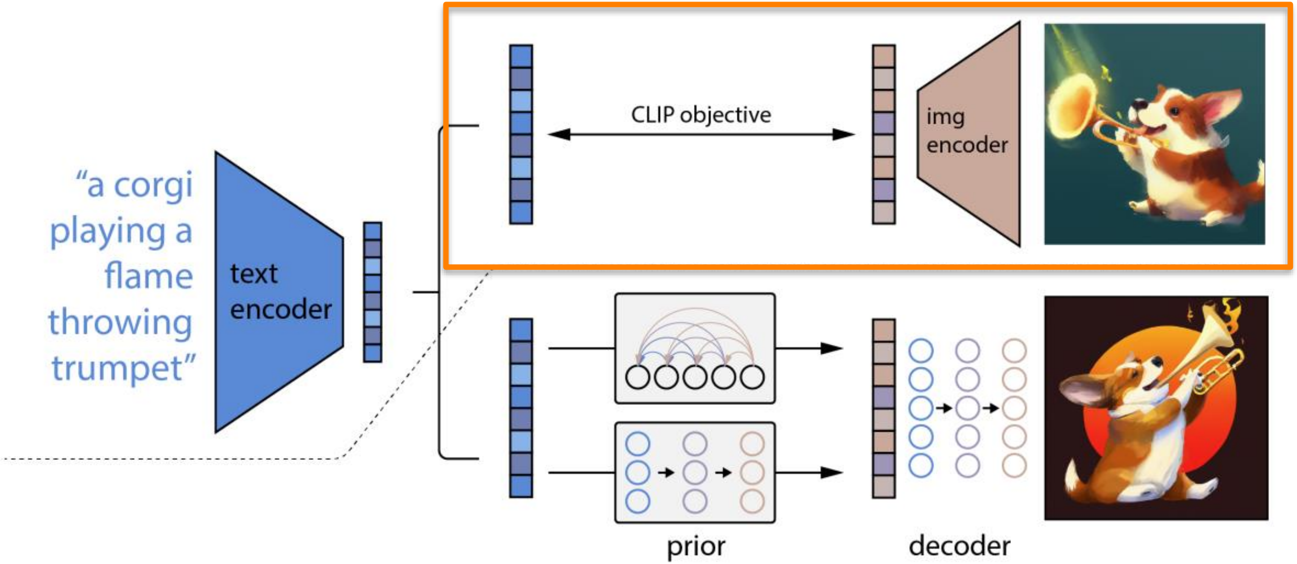
"a shiba inu wearing a beret and black turtleneck"

Radford et al. "Hierarchical Text-Conditional Image Generation with CLIP Latents".

CLIP

- Contrastive Language-Image Pre-training

Use case: text-to-image generation (DALL-E 2)



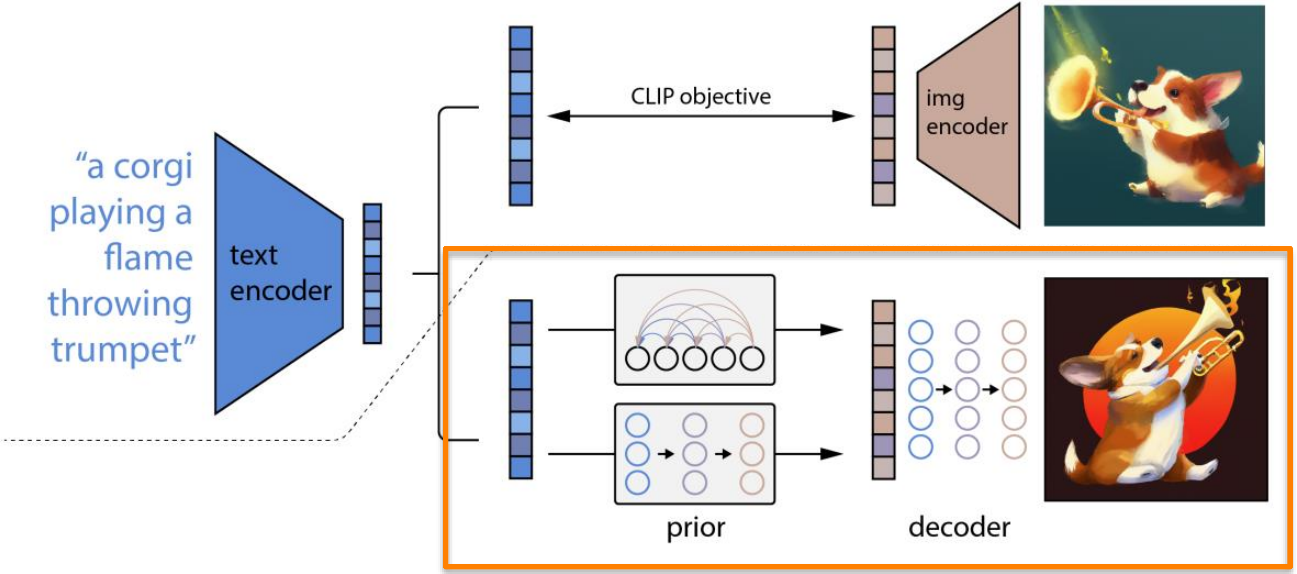
CLIP training process

Radford et al. "Hierarchical Text-Conditional Image Generation with CLIP Latents".

CLIP

- **C**ontrastive **L**anguage-Image **P**re-training

Use case: text-to-image generation (DALL-E 2)



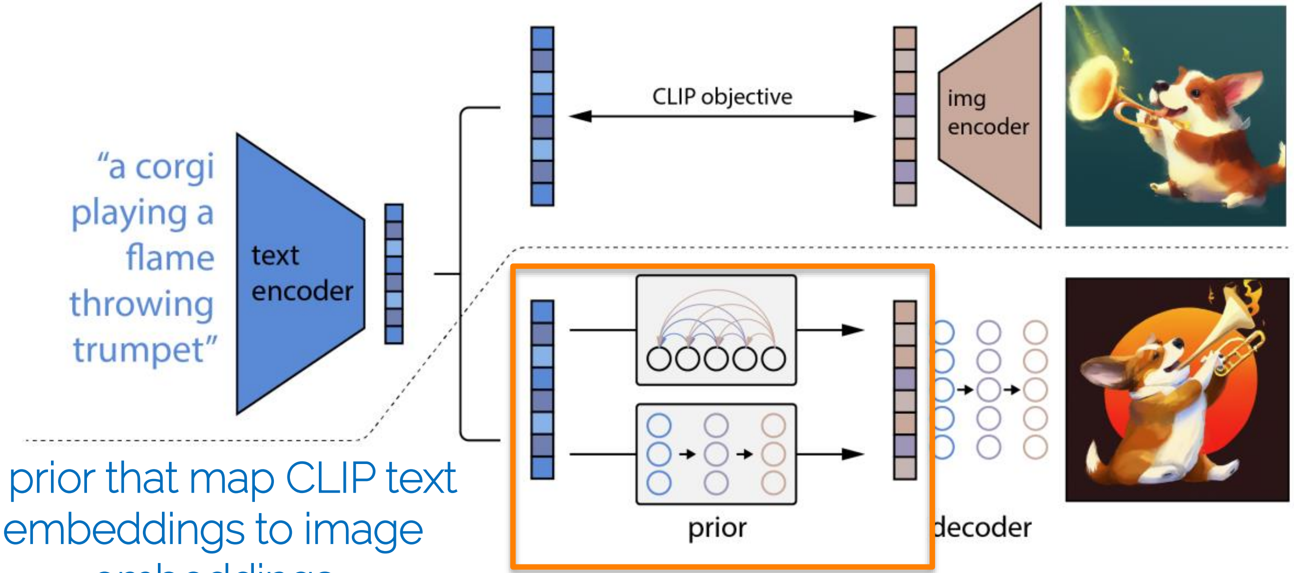
Generation process

Radford et al. "Hierarchical Text-Conditional Image Generation with CLIP Latents".

CLIP

- **C**ontrastive **L**anguage-**I**mage **P**re-training

  Use case: text-to-image generation (DALL-E 2)



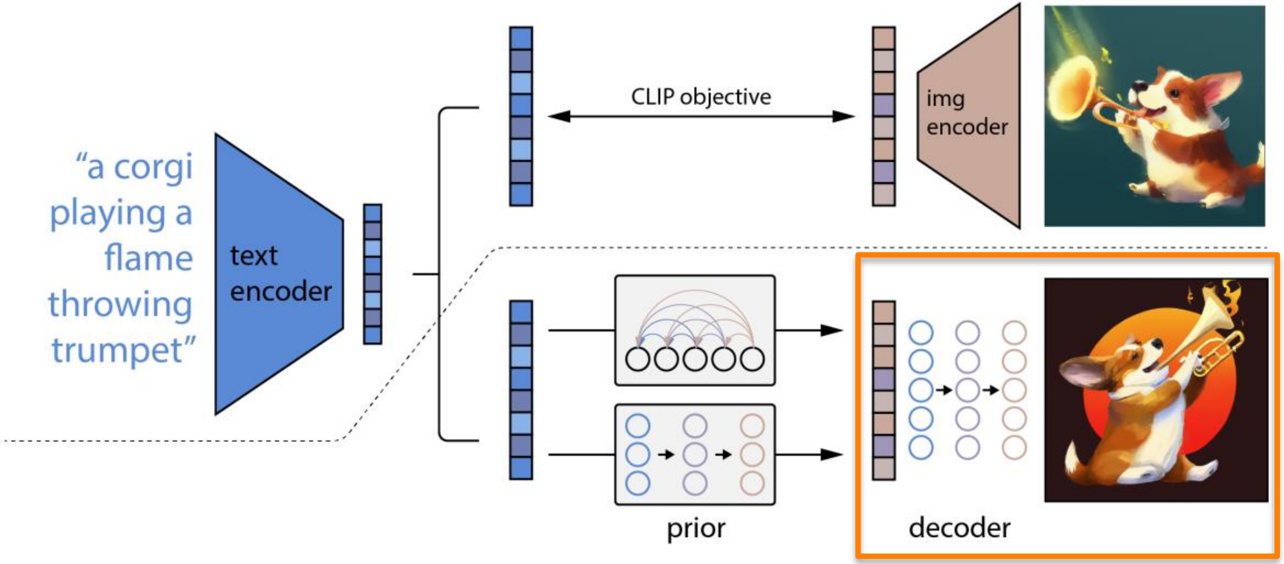
A prior that map CLIP text embeddings to image embeddings

Radford et al. "Hierarchical Text-Conditional Image Generation with CLIP Latents".

CLIP

- Contrastive Language-Image Pre-training

Use case: text-to-image generation (DALL-E 2)

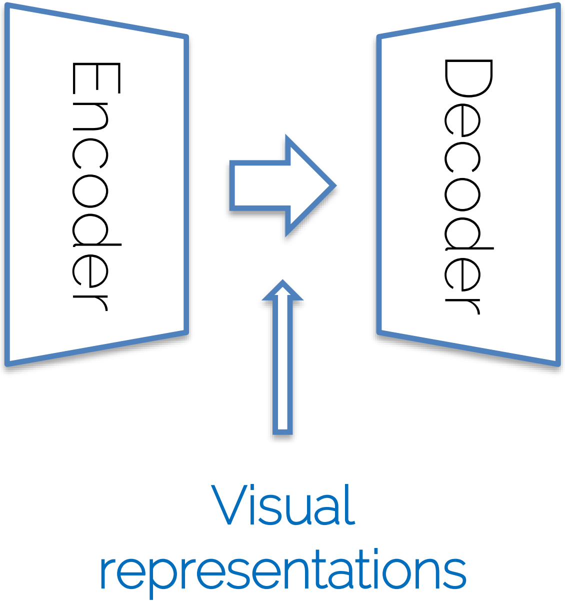


A decoder produces images conditioned on CLIP image embeddings

Radford et al. "Hierarchical Text-Conditional Image Generation with CLIP Latents".

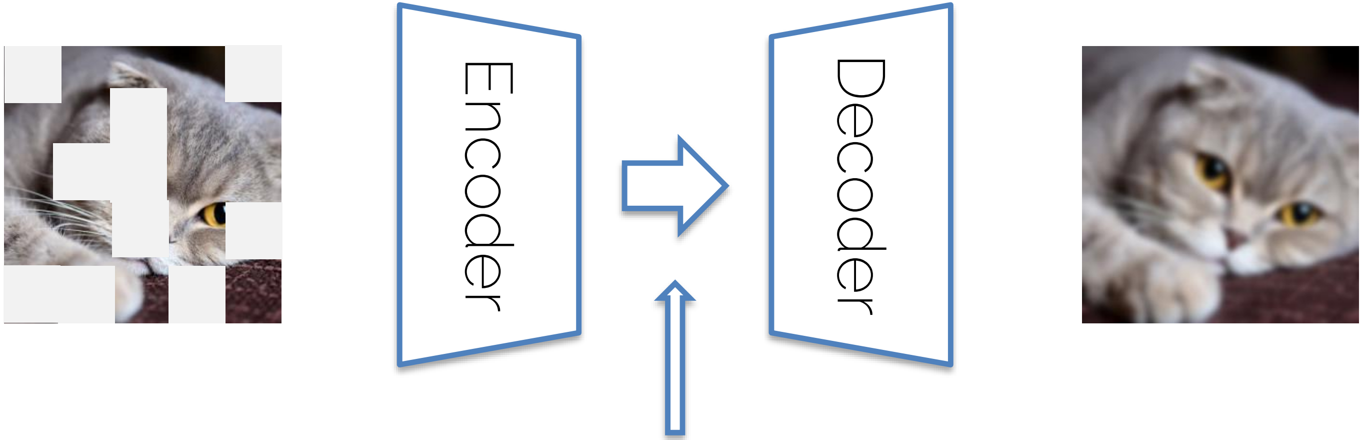
Other Self-supervised Approaches

- Autoencoder to Masked autoencoder



Other Self-supervised Approaches

- Autoencoder to Masked autoencoder



Visual representations

More in the next lecture!

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

Representation Learning Caveats

- Lots of hyperparameters make difficult to asses what made improvements possible:
  - Better engineering vs method idea

- Long training cycles make things difficult to reproduce, in particular, for class projects

- Improvements can be small but require lots of effort to produce (training + hyperparam finding)

Reading Homework

- MoCo v2: [Chen et al. 2020] Improved Baselines with Momentum Contrastive Learning
  - https://arxiv.org/pdf/2003.04297v1.pdf

- MoCo v3: [Chen et al. 2020] An Empirical Study of Training Self-Supervised Vision Transformers
  - https://arxiv.org/pdf/2104.02057v4.pdf

- Masked autoencoder: [He et al. 2021] Masked Autoencoders Are Scalable Vision Learners
  - https://openaccess.thecvf.com/content/CVPR2022/papers/He_Masked_Autoencoders_Are_Scalable_Vision_Learners_CVPR_2022_paper.pdf

Literature

- DINO: [Caron. 2021] Emerging Properties in Self-Supervised Vision Transformers
  - https://arxiv.org/pdf/2104.14294.pdf
- MoCo: [He et al. 2019] Momentum Contrast for Unsupervised Visual Representation Learning
  - https://arxiv.org/pdf/1911.05722.pdf
- SimCLR: [Chen et al. 2020] A Simple Framework for Contrastive Learning of Visual Representations
  - https://arxiv.org/pdf/2002.05709.pdf
- CLIP: [Radford et al. 2021] Learning Transferable Visual Models From Natural Language Supervision
  - https://arxiv.org/pdf/2103.00020.pdf
- DALL-E 2: [Ramesh et al. 2022] Hierarchical Text-Conditional Image Generation with CLIP Latents
  - https://arxiv.org/pdf/2204.06125.pdf

Thanks for watching!