The Self-Supervised Learning Paradigm in Computer Vision

Nikos Komodakis

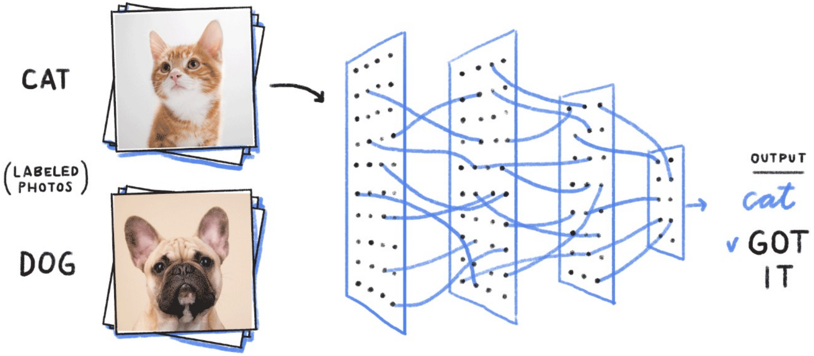
Computer Science Department, University of Crete
Archimedes, Athena RC
IACM, Forth

Outline

- Intro/motivation
- What is self-supervised learning (SSL)
- Different SSL paradigms
- A tour of SSL approaches

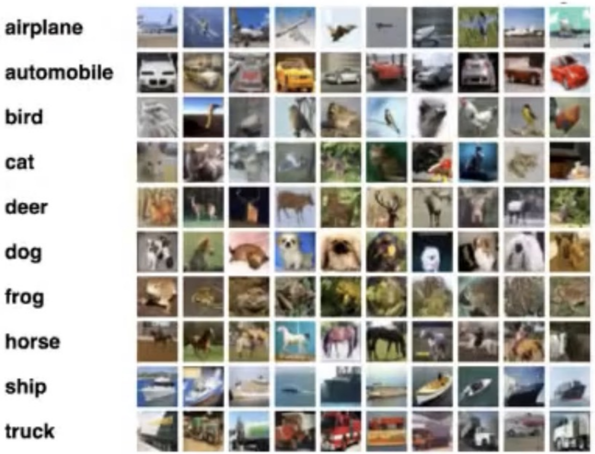
Deep learning

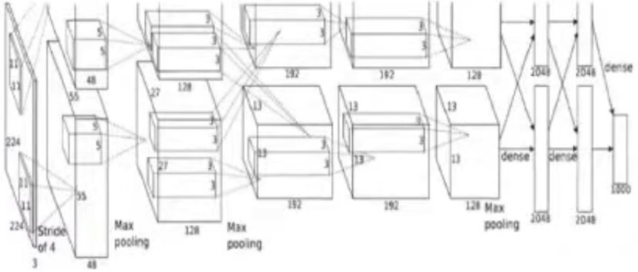
- Multi-layered neural networks

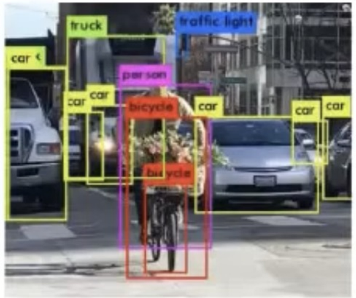


- Revolutionized many research fields

- **Software v2.0**



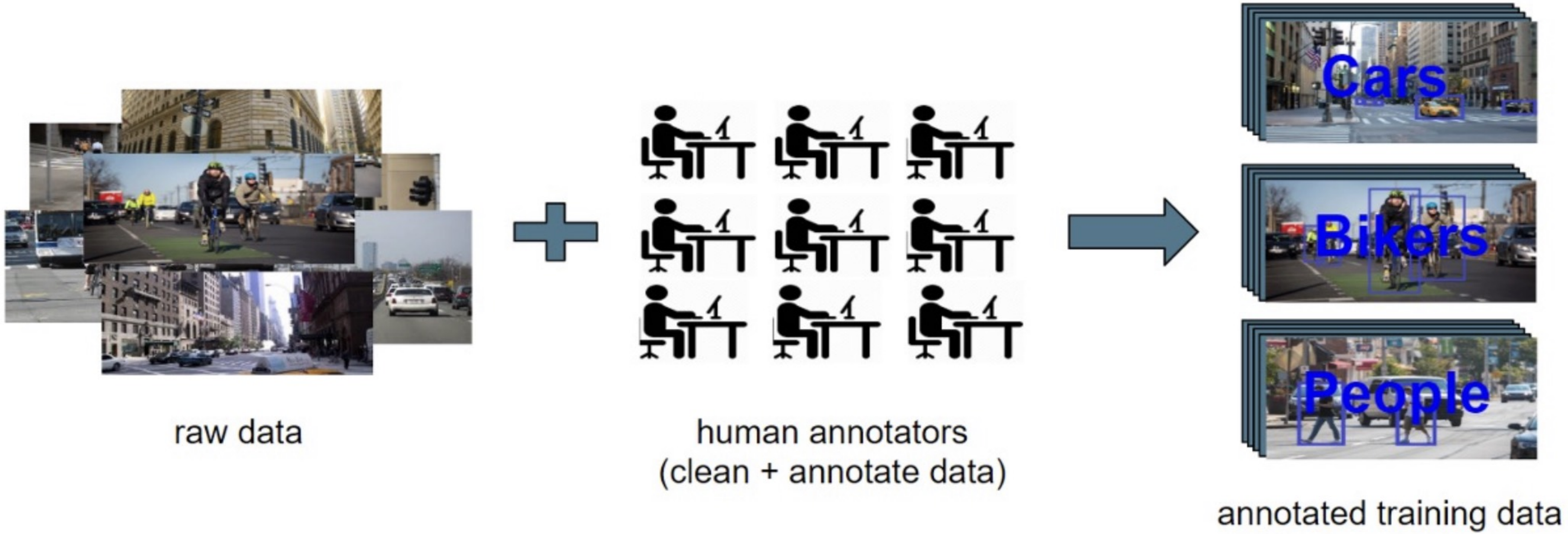




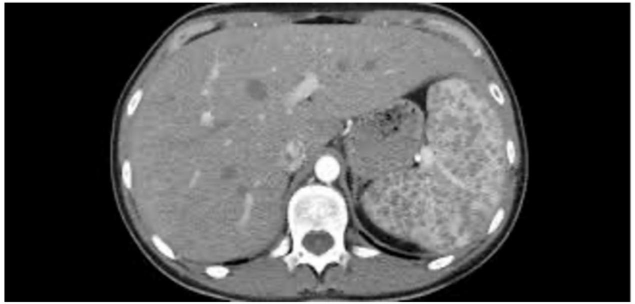
- Predefine the set of visual concepts to be learned
- Collect diverse and large number of examples for each of them

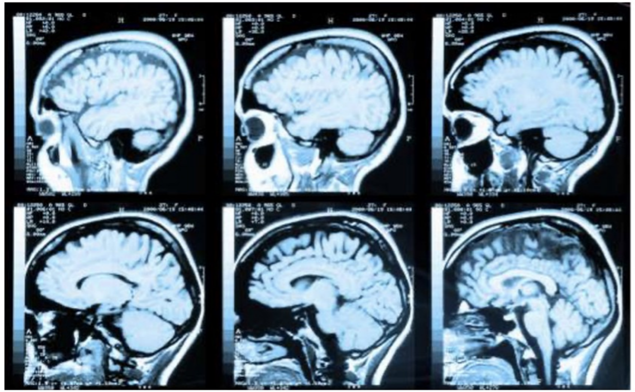
Massive amounts of manually annotated training data required

- ❑ Collecting raw data: (relatively) easy
- ❑ Annotating raw data: **very expensive & time consuming**
    - thousands of hours of tedious, error-prone human labor

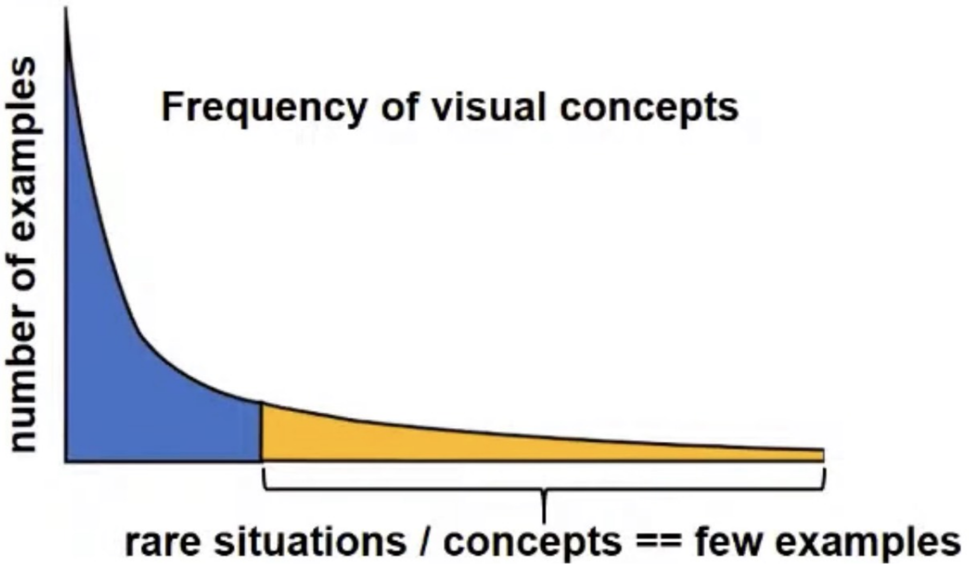


- Lack of qualified human experts to annotate data



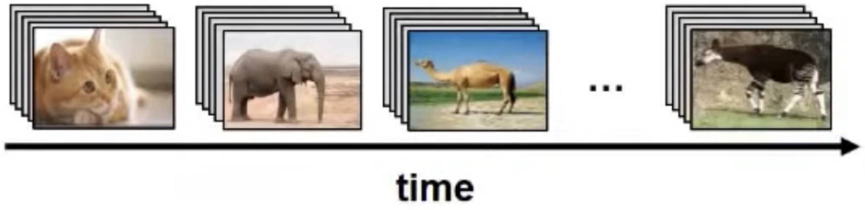


Long-tail distribution

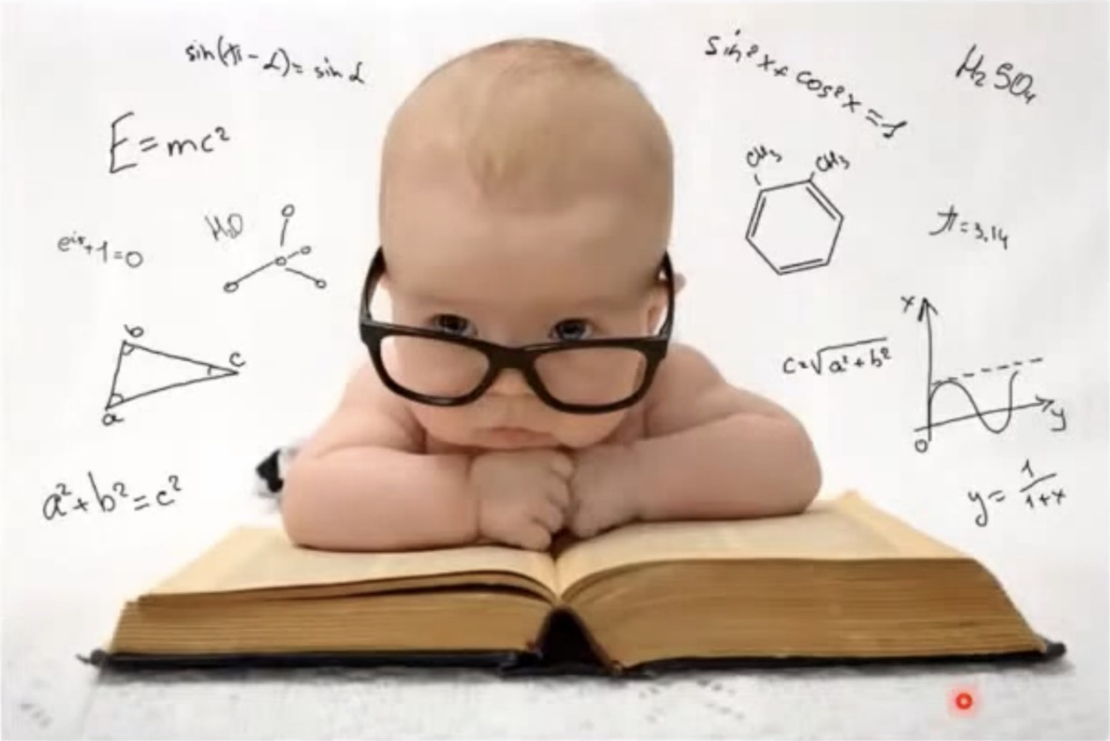


Typical supervised training: fixed dataset

In real life need to continually adapt to new data

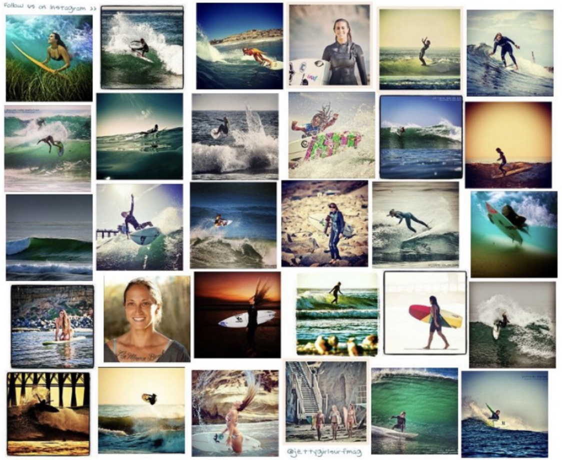


Gap with how humans learn



Exploiting unlabeled data

- Million of images uploaded (e.g. on Facebook) per day
- Hours of video uploaded on Youtube per minute
- “infinite” amount of text data available online

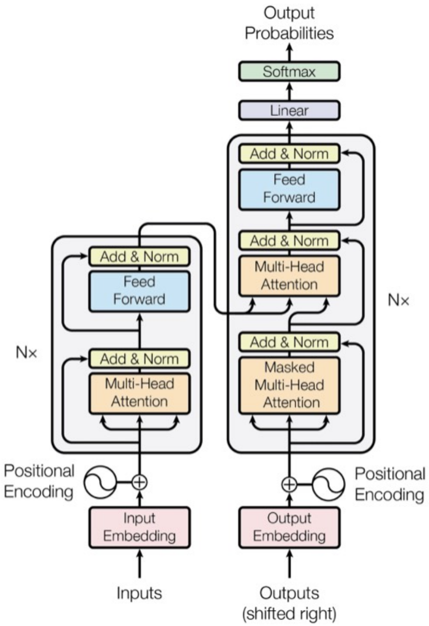


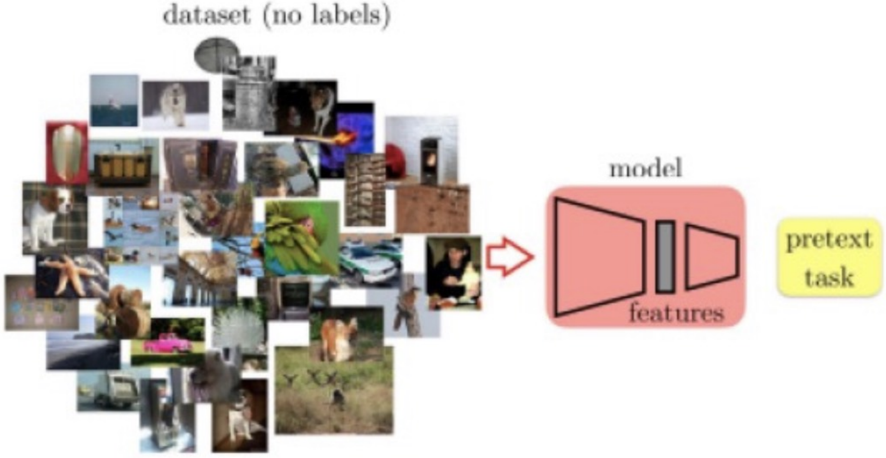
Two big recent breakthroughs

Transformers

a match made in heaven

Self-supervised learning



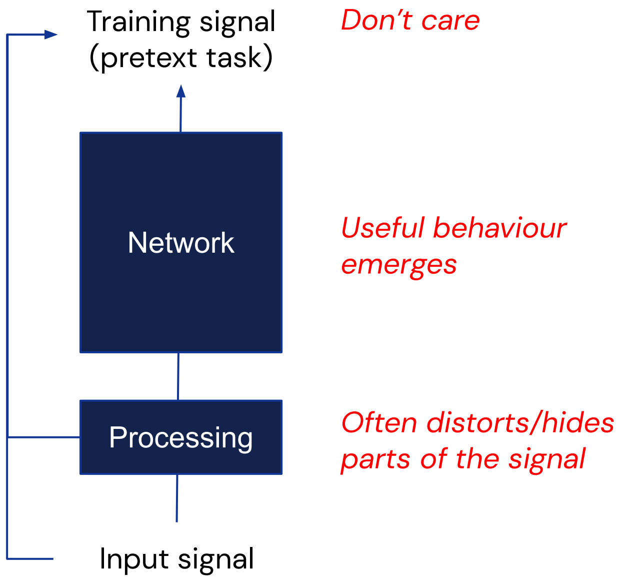


What is self-supervised image representation learning?

Self-supervised learning in a nutshell

Self-supervised learning

- **Goal:** Learn good representations
- **Means:** Construct a pretext task
  - Don't care about the pretext task itself
  - Only important it enables learning

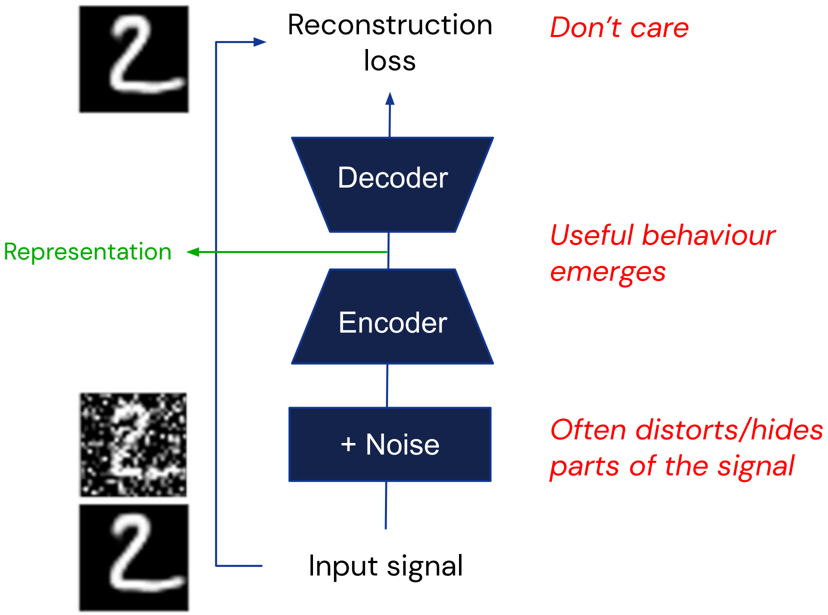


["Extracting and composing robust features with denoising autoencoders", Vincent et al. 08]

Self-supervised learning in a nutshell

Self-supervised learning

- **Goal:** Learn good representations
- **Means:** Construct a pretext task
    - Don't care about the pretext task itself
    - Only important it enables learning



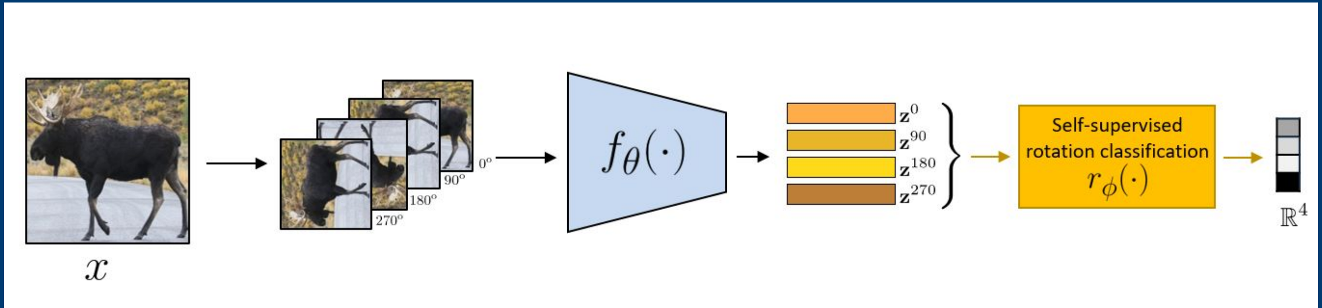
SSL: the platypus of Machine Learning

"The unusual appearance of this egg-laying, duck-billed, beaver-tailed, otter-footed mammal baffled European naturalists when they first encountered it, and the first scientists to examine a preserved platypus body (in 1799) judged it a fake, made of several animals sewn together."

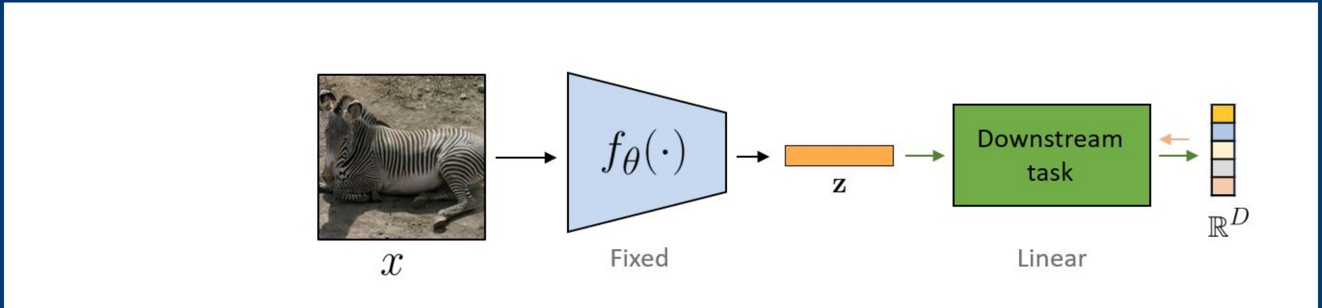
Somewhere in-between unsupervised and supervised learning

Self-supervised learning pipeline

Stage 1: Train network on pretext task (without human labels)

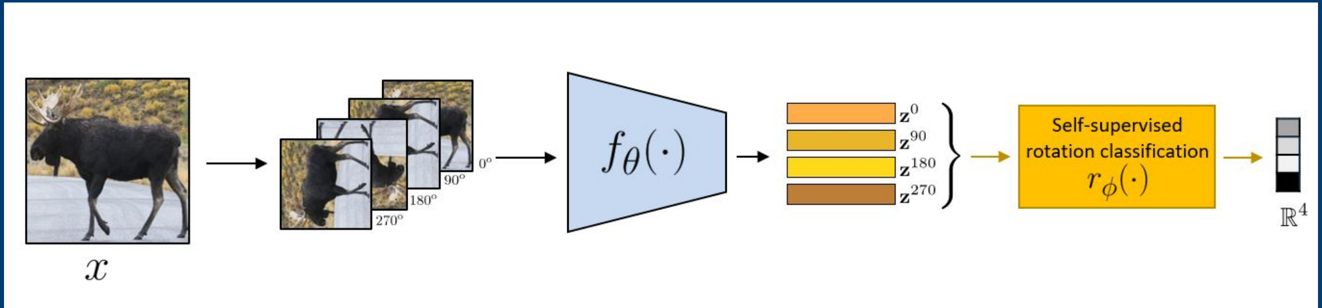


Stage 2: Train classifier on learned features for new task with fewer labels

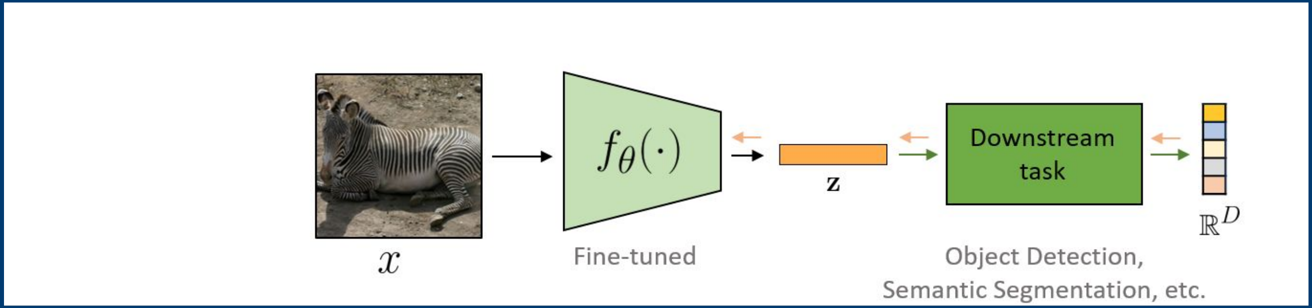


Self-supervised learning pipeline

Stage 1: Train network on pretext task (without human labels)



Stage 2: Fine-tune network for new task with fewer labels



Karate Kid and Self-Supervised Learning

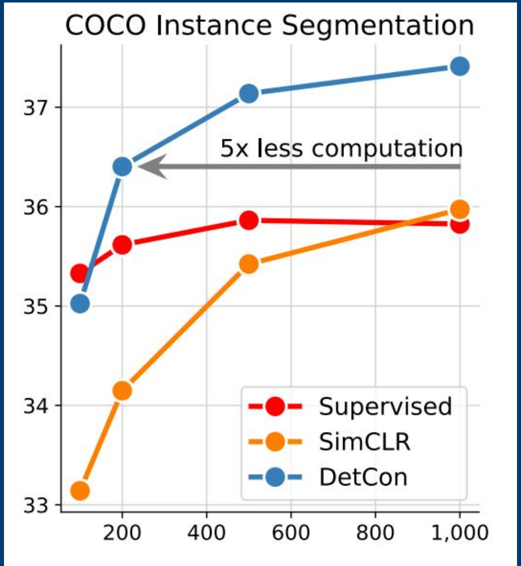
The Karate Kid (1984)

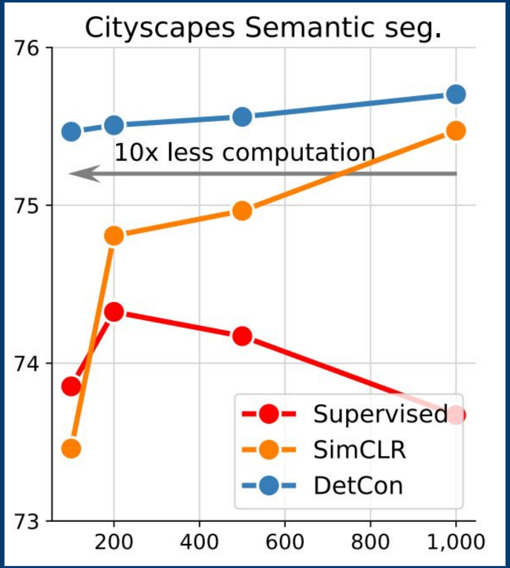
Stage 1: Train *muscle memory* on pretext tasks

Stage 2: Fine-tune skills rapidly

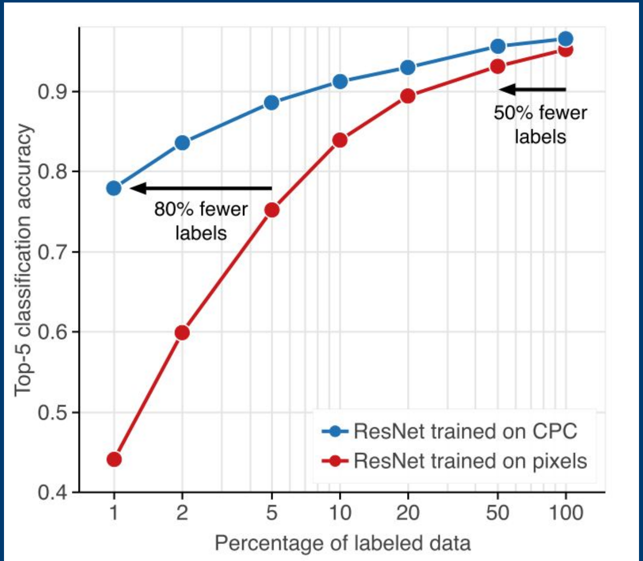
Is this actually useful in practice?

SSL methods are often more efficient than supervised methods





Efficiency in terms of number of epochs for ImageNet pretraining (SimCLR and DetCon do no use human annotated labels)

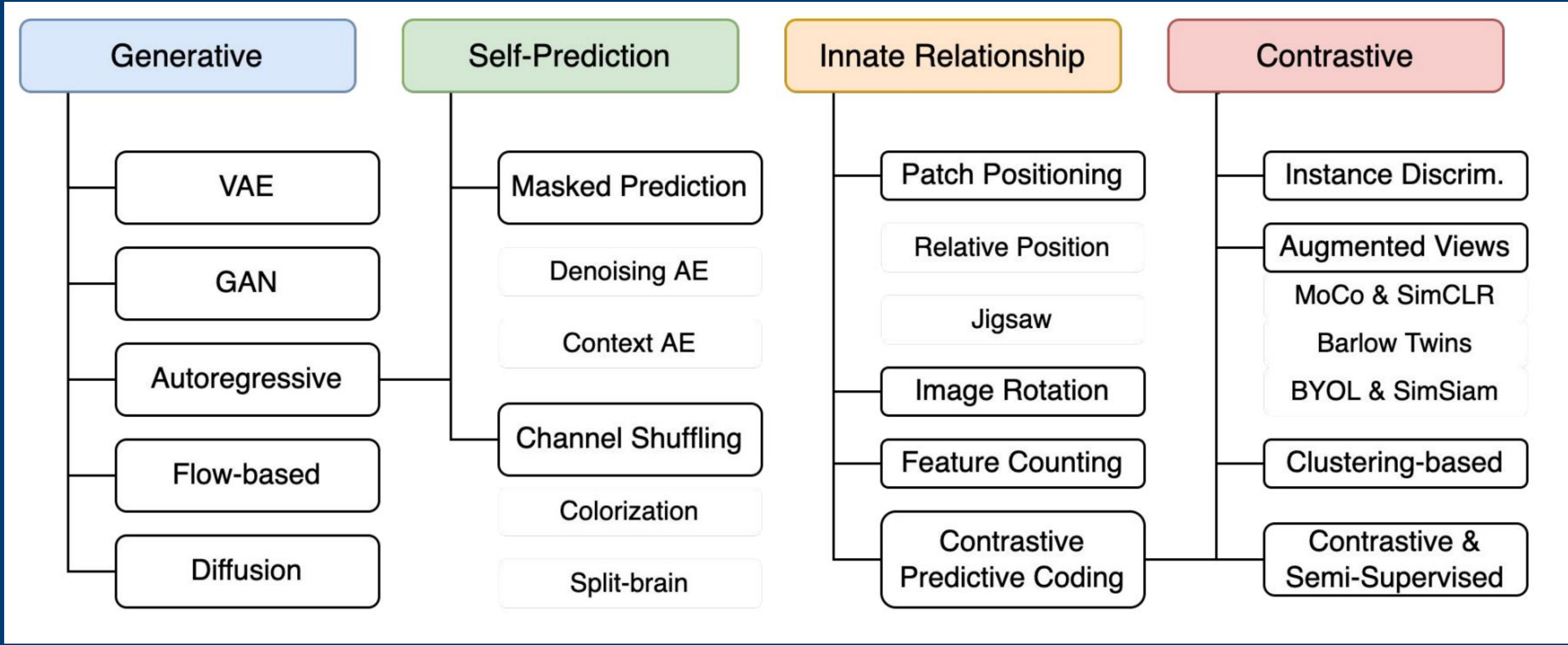


Data-efficiency of SSL and supervised learning methods

Hénaff et al., Efficient Visual Pretraining with Contrastive Detection, ICCV 2021
Hénaff et al., Data-Efficient Image Recognition with Contrastive Predictive Coding, ICML 2020

SSL paradigms for computer vision

There are many flavors of SSL



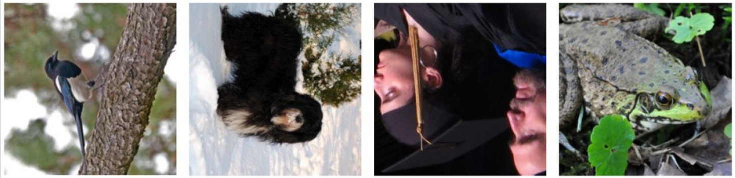
Taxonomy of pretext tasks (Weng & Kim, 2021)

Weng & Kim, Self-Supervised Learning: Self-Prediction and Contrastive Learning, NeurIPS Tutorials 2021

A tour of pretext tasks for Self-Supervised Learning

Early examples of pretext tasks

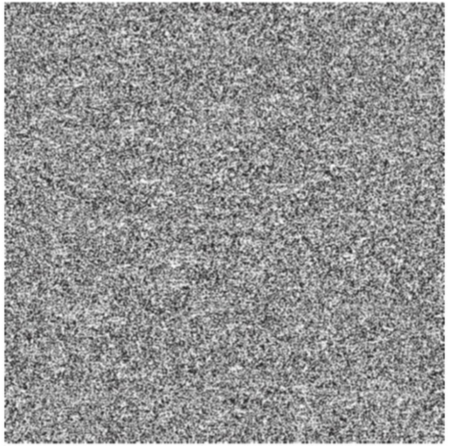
Transformation prediction


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

Self-supervised representation learning by predicting image rotations

Can you predict the 2D rotation of this image ?

Self-supervised representation learning by predicting image rotations

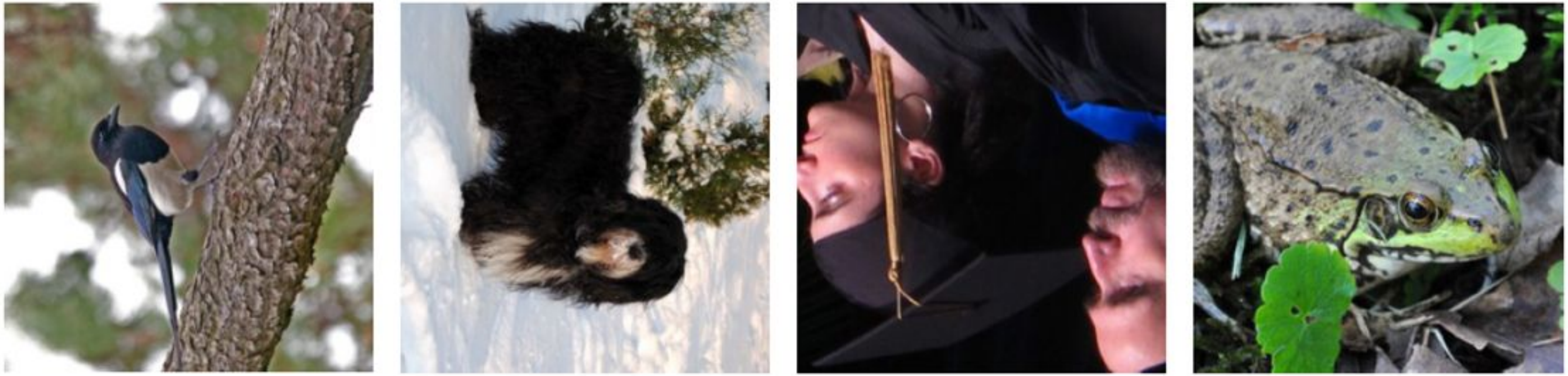


What about this image ?

["Unsupervised representation learning by predicting image rotations", Gidaris et al. 18]

Rotation prediction

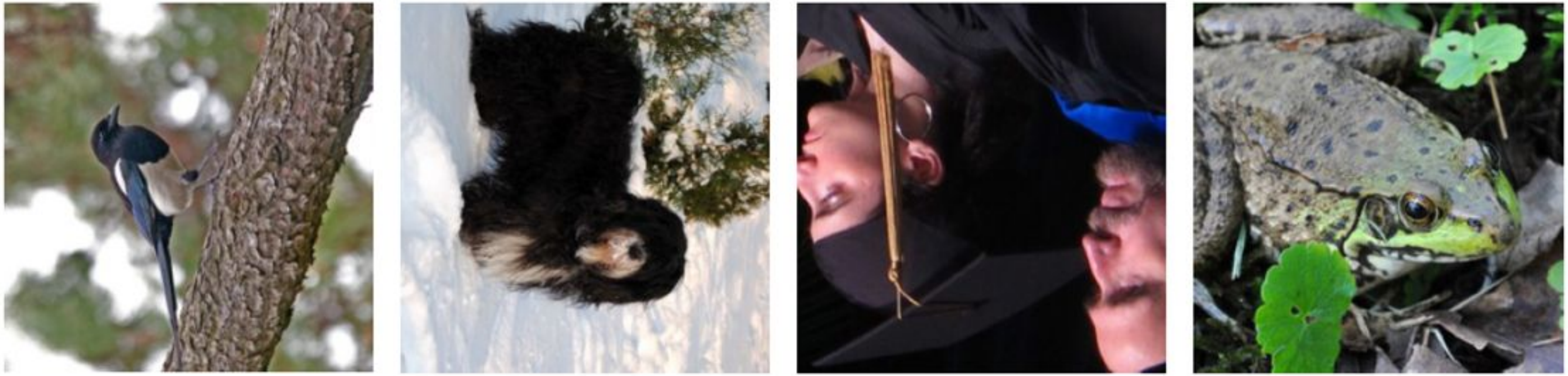
Can you guess how much rotated is applied?



[“Unsupervised representation learning by predicting image rotations”, Gidaris et al. 18]

Rotation prediction

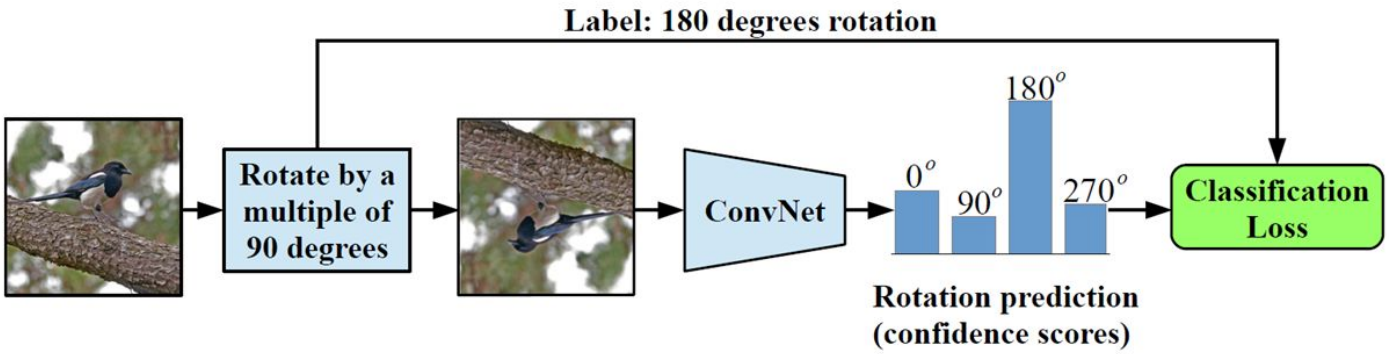
Can you guess how much rotated is applied? Much easier if you recognize the content!



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

["Unsupervised representation learning by predicting image rotations", Gidaris et al. 18]

Rotation prediction



Pros

- Very simple to implement and use, while being quite effective

Cons

- Assumes training images are photographed with canonical orientations (and canonical orientations exist)
- Train-eval gap: no rotated images at eval
- Not fine-grained enough due to no negatives from other images
  - e.g. no reason to distinguish cat from dog
- Small output space - 4 cases (rotations) to distinguish [not trivial to increase; see later]
- Some domains are trivial e.g. StreetView ⇒ just recognize sky

Transfer learned features to supervised learning

	Classification (%mAP)		Detection (%mAP)	Segmentation (%mIoU)
Trained layers	fc6-8	all	all	all
ImageNet labels	78.9	79.9	56.8	48.0
Random		53.3	43.4	19.8
Random rescaled Krähenbühl et al. (2015)	39.2	56.6	45.6	32.6
Egomotion (Agrawal et al., 2015)	31.0	54.2	43.9	
Context Encoders (Pathak et al., 2016b)	34.6	56.5	44.5	29.7
Tracking (Wang & Gupta, 2015)	55.6	63.1	47.4	
Context (Doersch et al., 2015)	55.1	65.3	51.1	
Colorization (Zhang et al., 2016a)	61.5	65.6	46.9	35.6
BIGAN (Donahue et al., 2016)	52.3	60.1	46.9	34.9
Jigsaw Puzzles (Noroozi & Favaro, 2016)	-	67.6	53.2	37.6
NAT (Bojanowski & Joulin, 2017)	56.7	65.3	49.4	
Split-Brain (Zhang et al., 2016b)	63.0	67.1	46.7	36.0
ColorProxy (Larsson et al., 2017)		65.9		38.4
Counting (Noroozi et al., 2017)	-	67.7	51.4	36.6
(Ours) RotNet	**70.87**	**72.97**	**54.4**	**39.1**

Pretrained with full ImageNet supervision

No pretraining

Self-supervised learning with rotation prediction

Self-supervised learning on **ImageNet** (entire training set) with AlexNet.

Finetune on labeled data from **Pascal VOC 2007**.

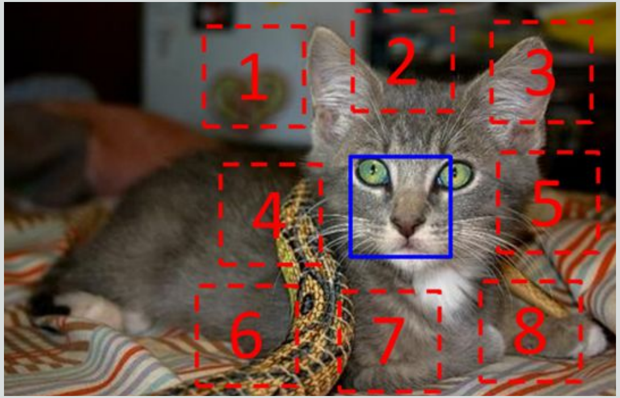
source: Gidaris et al. 2018

Visualize learned visual attentions

(a) **Attention maps of supervised model** (b) **Attention maps of our self-supervised model**

(Image source: Gidaris et al. 2018)

Inferring structure



["Unsupervised visual representation learning by context prediction", Doersh et al. 15]

Context prediction

Can you guess the spatial configuration for the two pairs of patches?

Question 1:

Question 2:

["Unsupervised visual representation learning by context prediction", Doersh et al. 15]

Context prediction

Can you guess the spatial configuration for the two pairs of patches? Much easier if you recognize the object!

Question 1:

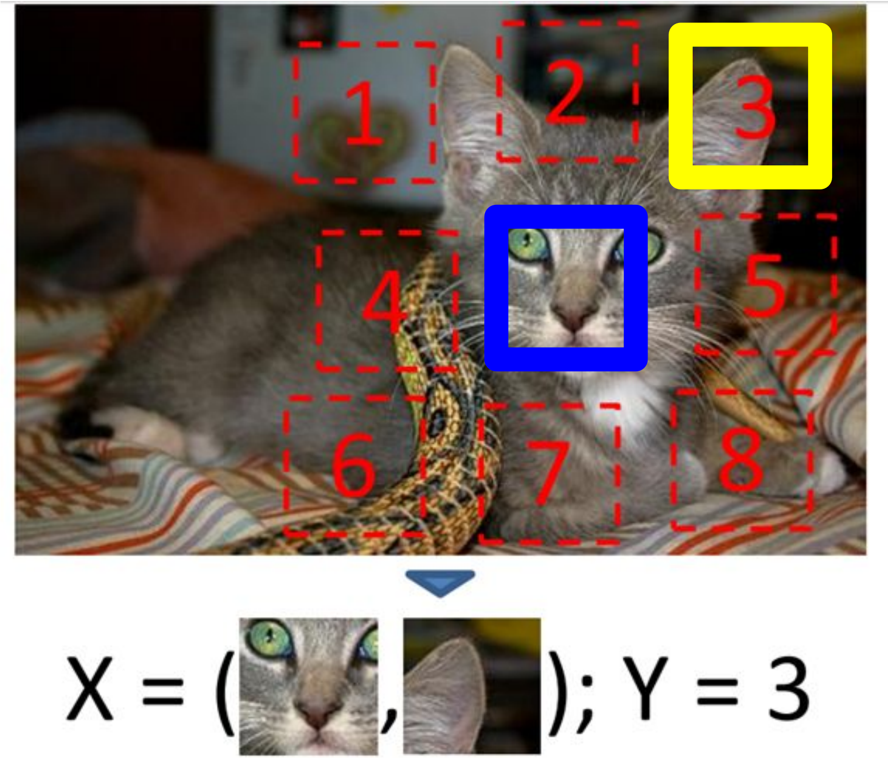
Question 2:

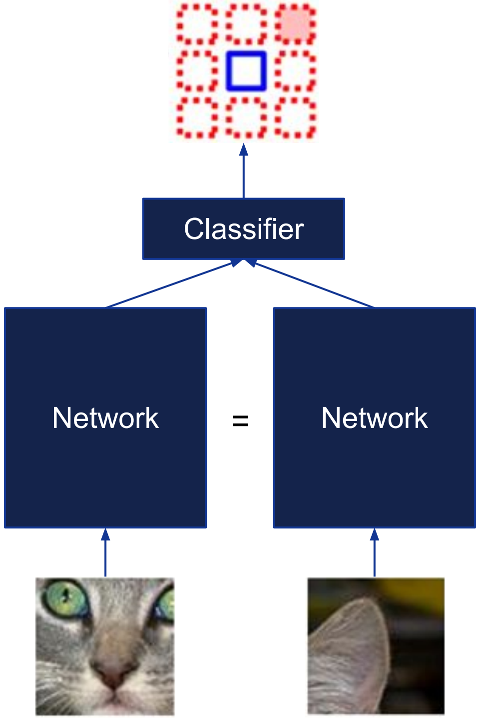
Intuition

- The network should learn to recognize object parts and their spatial relations

["Unsupervised visual representation learning by context prediction", Doersh et al. 15]

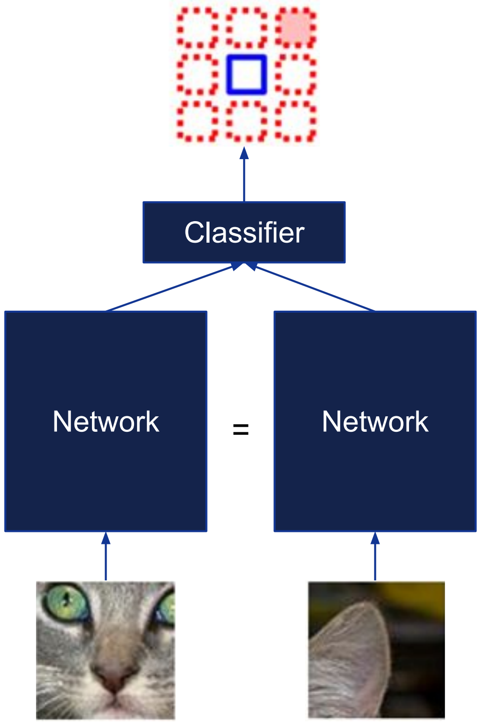
Context prediction





[“Unsupervised visual representation learning by context prediction”, Doersh et al. 15]

Context prediction



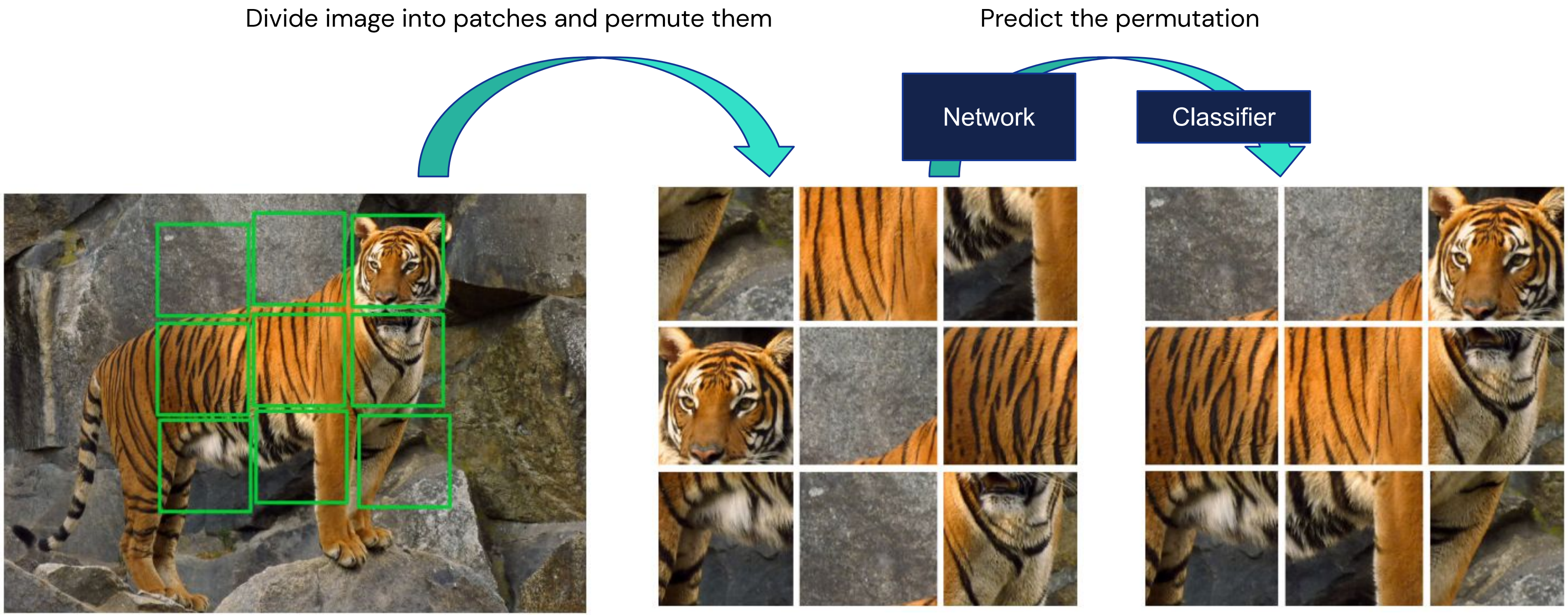
Pros

- (arguably) The first self-supervised method
- Intuitive task that should enable learning about object parts

Cons

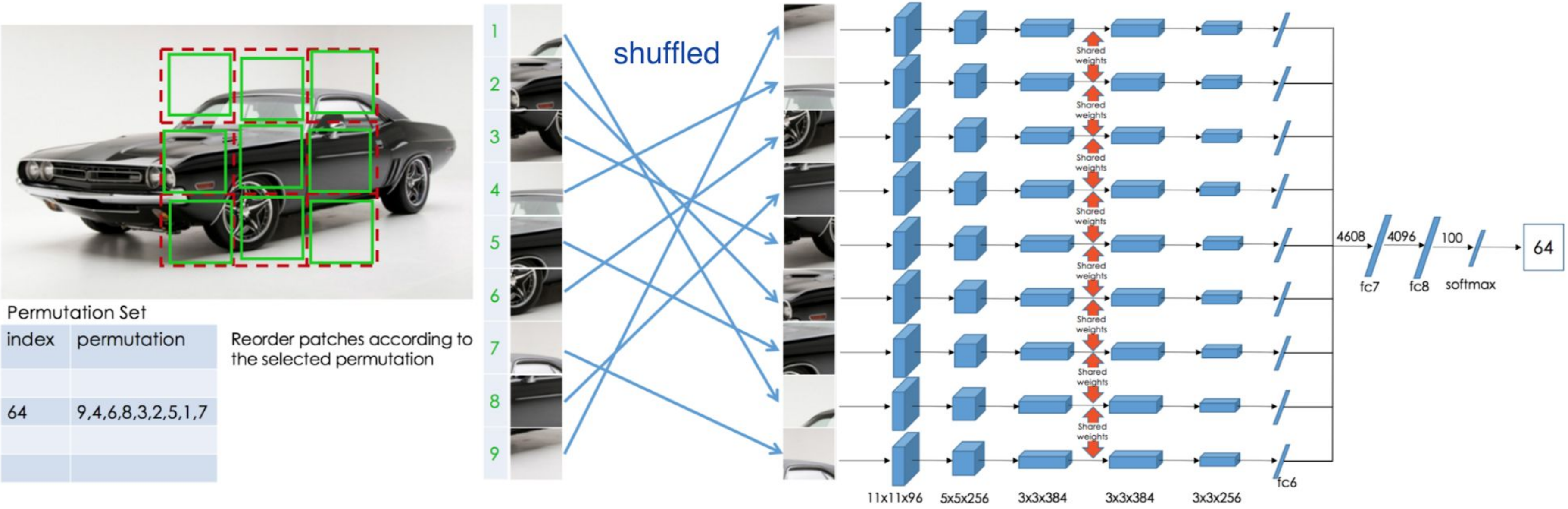
- Assumes training images are photographed with canonical orientations (and canonical orientations exist)
- Training on patches, but trying to learn image representations
- Networks can “cheat” so special care is needed [discussed later]
  - Further gap between train and eval
- Not fine-grained enough due to no negatives from other images
  - e.g. no reason to distinguish cat from dog eyes
- Small output space - 8 cases (positions) to distinguish?

Jigsaw puzzles



Pros & Cons: Same as for context prediction apart from being harder

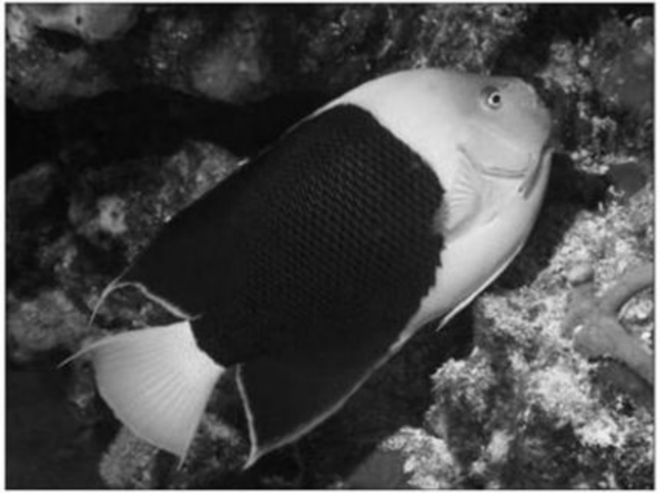
Pretext task: solving "jigsaw puzzles"



(Image source: [Noroozi & Favaro, 2016](https://arxiv.org/abs/1603.09246))

Colorization

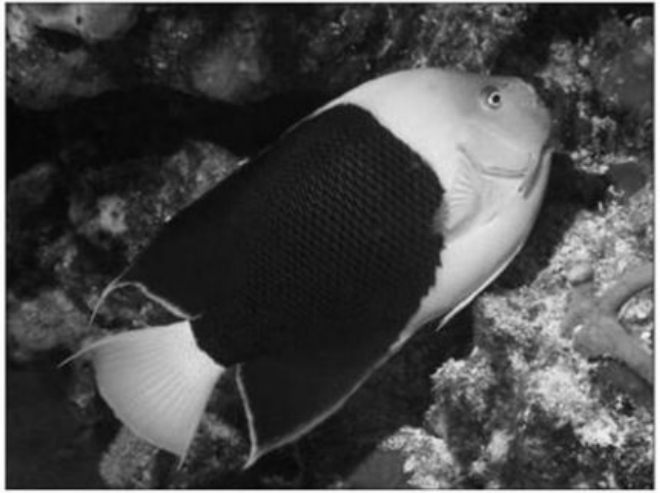
What is the colour of every pixel?



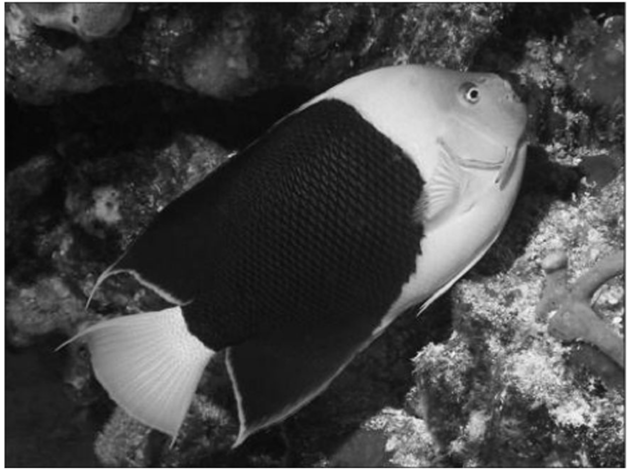
[“Colorful image colorization”, Zhang et al. 16]

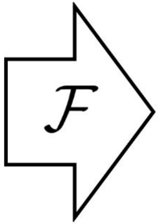
Colorization

What is the colour of every pixel? Hard if you don’t recognize the object!



Pretext task: image coloring



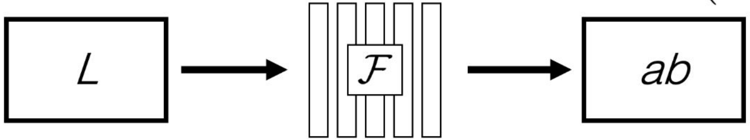


Grayscale image: *L* channel

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

Concatenate (*L*,*ab*) channels

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



Source: Richard Zhang / Phillip Isola

Pretext task: video coloring

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

reference frame

how should I color these frames?

t = 0

t = 1

t = 2

t = 3

...

Source: [Vondrick et al., 2018]

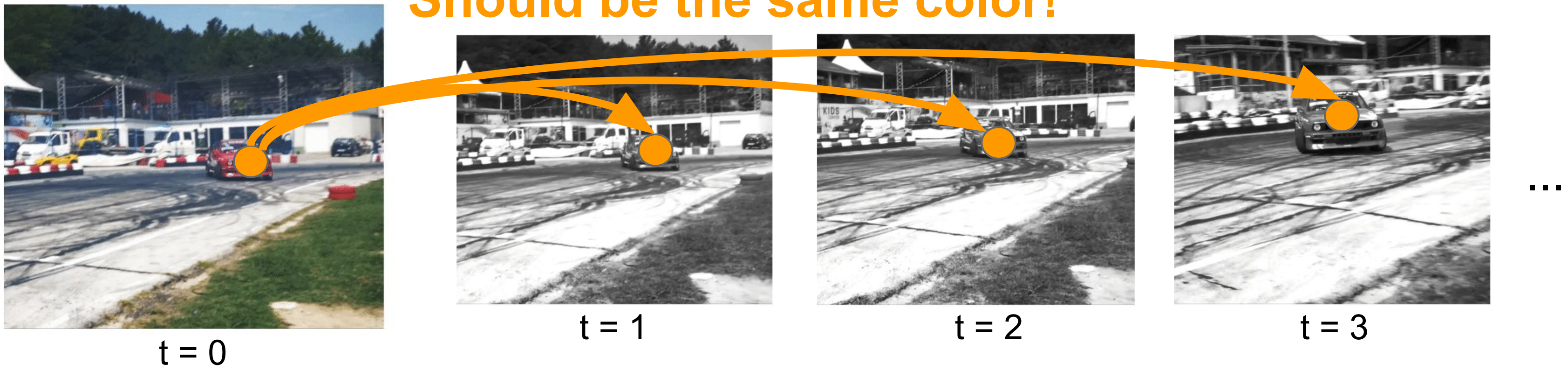
Pretext task: video coloring

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

reference frame

how should I color these frames?

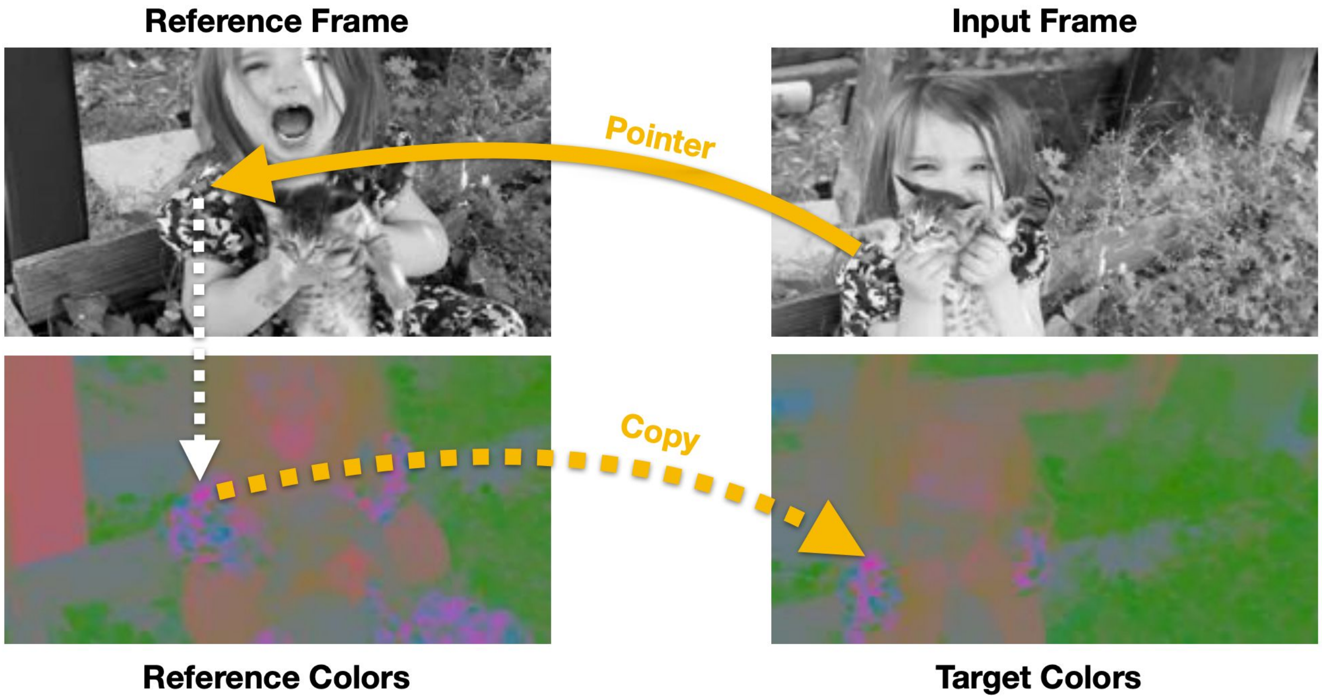
Should be the same color!



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

Source: [Vondrick et al., 2018]

Learning to color videos



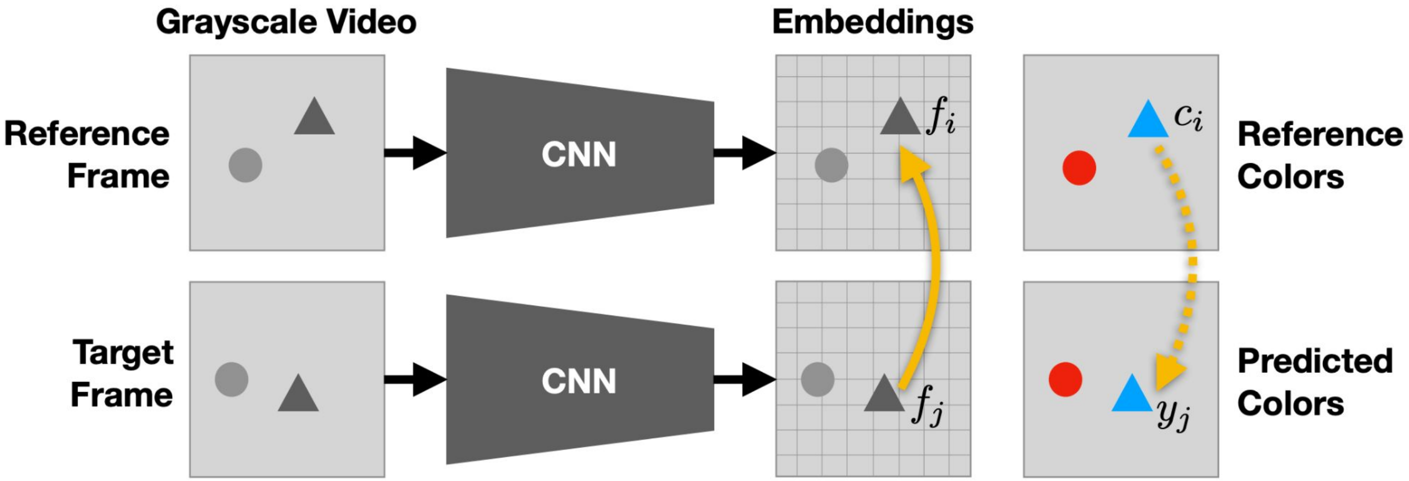
Learning objective:

Establish mappings between reference and target frames in a learned feature space.

Use the mapping as "pointers" to copy the correct color (LAB).

Source: Vondrick et al., 2018

Learning to color videos

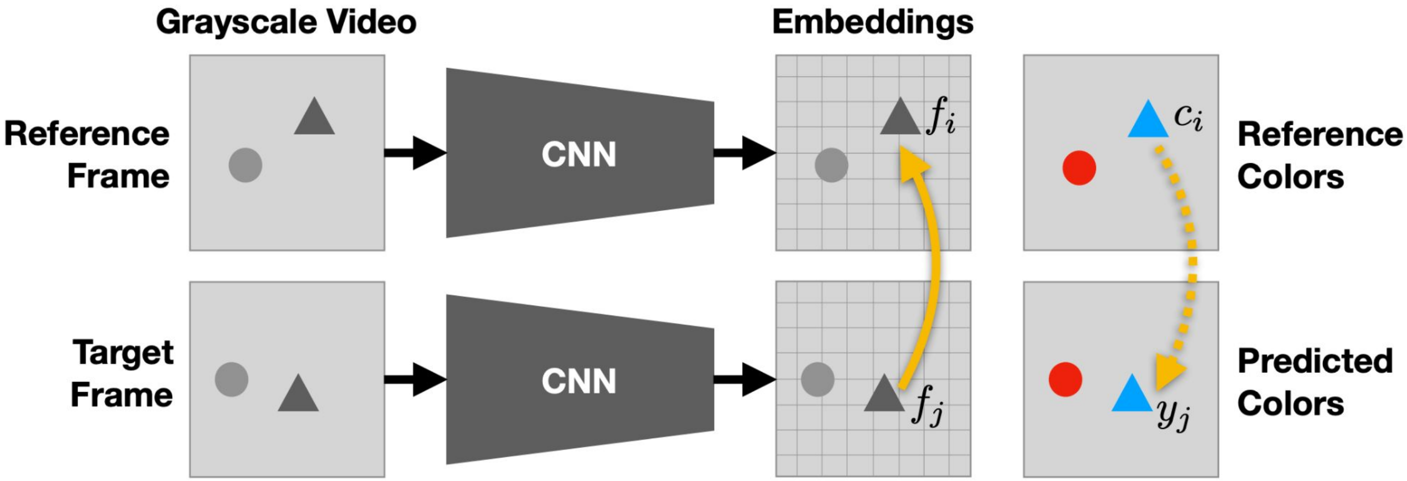


attention map on the reference frame

$$A_{ij} = \frac{\exp\left(f_i^T f_j\right)}{\sum_k \exp\left(f_k^T f_j\right)}$$

Source: [Vondrick et al., 2018]

Learning to color videos



attention map on the reference frame

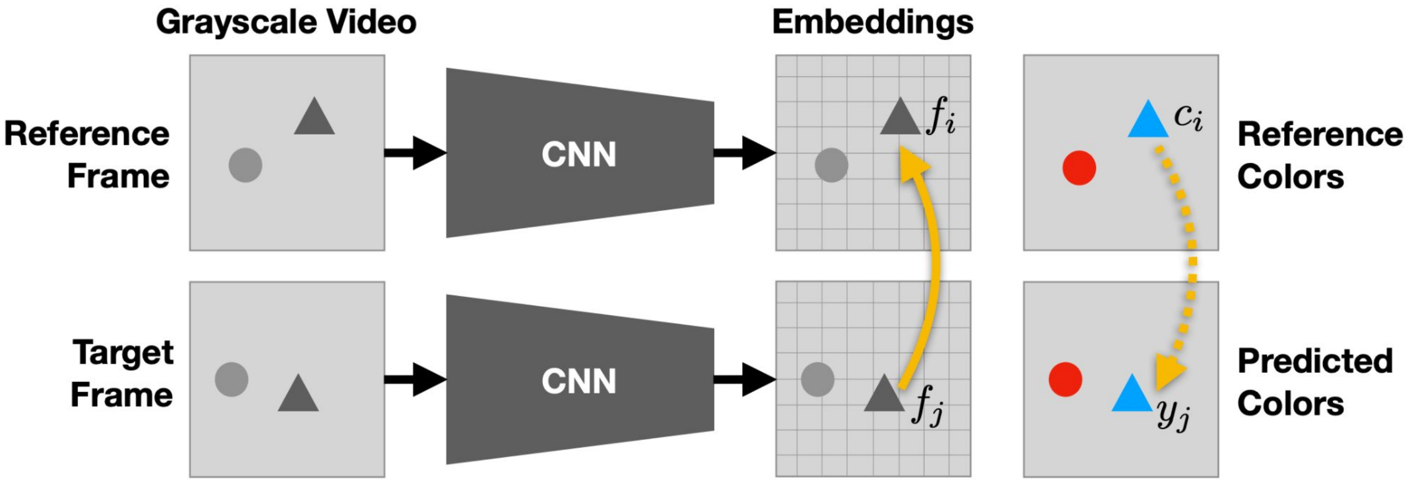
$$A_{ij} = \frac{\exp\left(f_i^T f_j\right)}{\sum_k \exp\left(f_k^T f_j\right)}$$

predicted color = weighted sum of the reference color

$$y_j = \sum_i A_{ij} c_i$$

Source: [Vondrick et al., 2018]

Learning to color videos



attention map on the reference frame

$$A_{ij} = \frac{\exp\left(f_i^T f_j\right)}{\sum_k \exp\left(f_k^T f_j\right)}$$

predicted color = weighted sum of the reference color

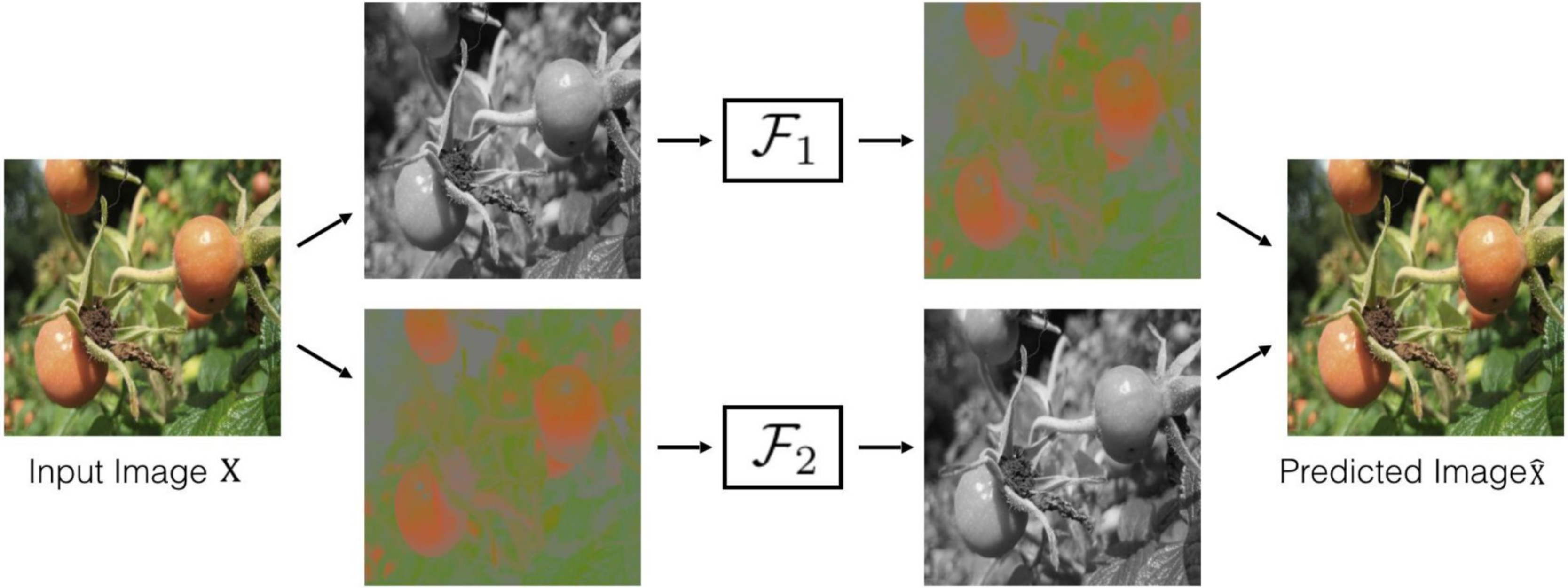
$$y_j = \sum_i A_{ij} c_i$$

loss between predicted color and ground truth color

$$\min_\theta \sum_j \mathcal{L}\left(y_j, c_j\right)$$

Source: [Vondrick et al., 2018]

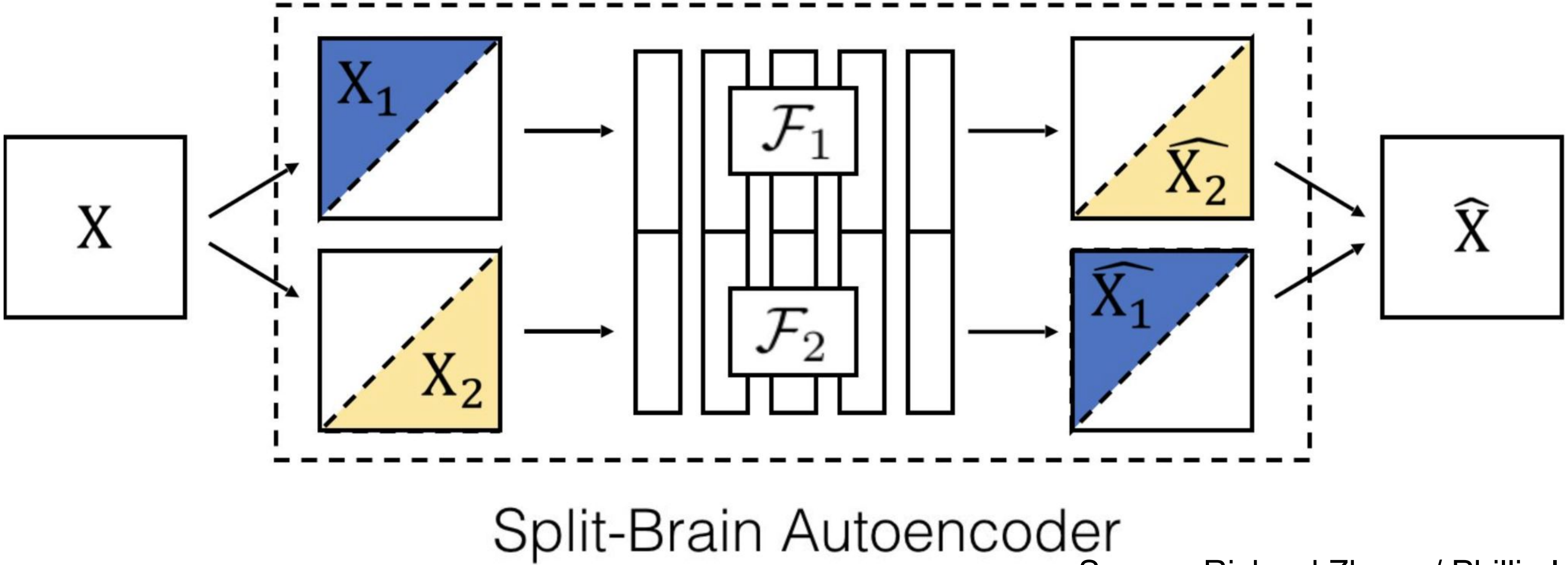
Learning features from colorization: Split-brain Autoencoder



Source: Richard Zhang / Phillip Isola

Learning features from colorization: Split-brain Autoencoder

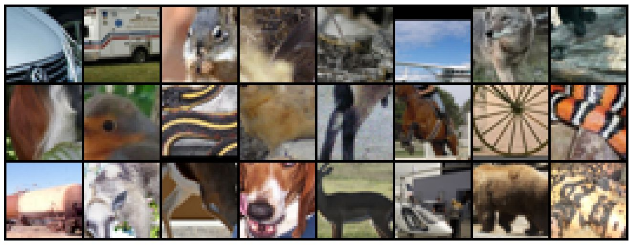
Idea: cross-channel predictions



Source: Richard Zhang / Phillip Isola

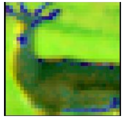
The contrastive-based SSL paradigm

Instance classification



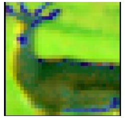
["Discriminative unsupervised feature learning with exemplar convolutional neural networks", Dosovitskiy et al. 14]

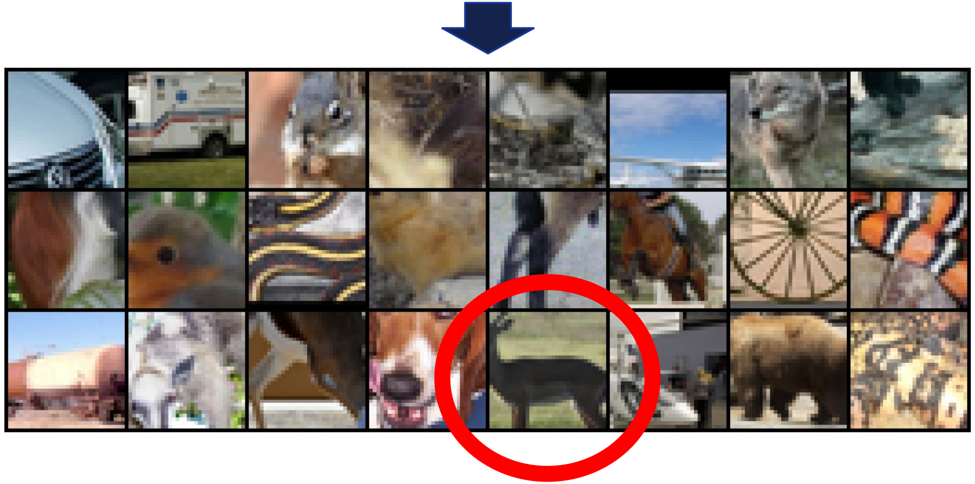
Exemplar ConvNets

This  is a distorted crop extracted from an image, which of these crops has the same source image?

["Discriminative unsupervised feature learning with exemplar convolutional neural networks", Dosovitskiy et al. 14]

Exemplar ConvNets

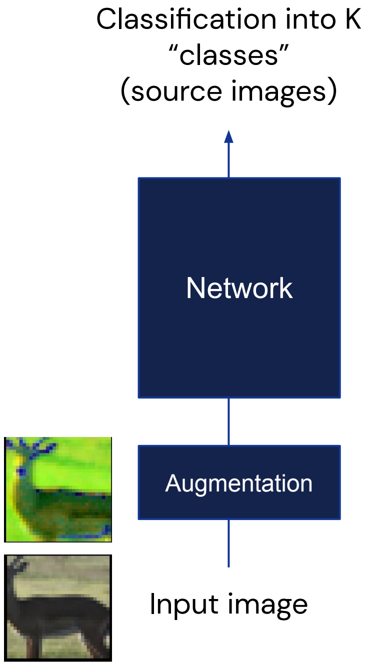
This  is a distorted crop extracted from an image, which of these crops has the same source image?



Easy if robust to the desired transformations (geometry and colour)

["Discriminative unsupervised feature learning with exemplar convolutional neural networks", Dosovitskiy et al. 14]

Exemplar ConvNets



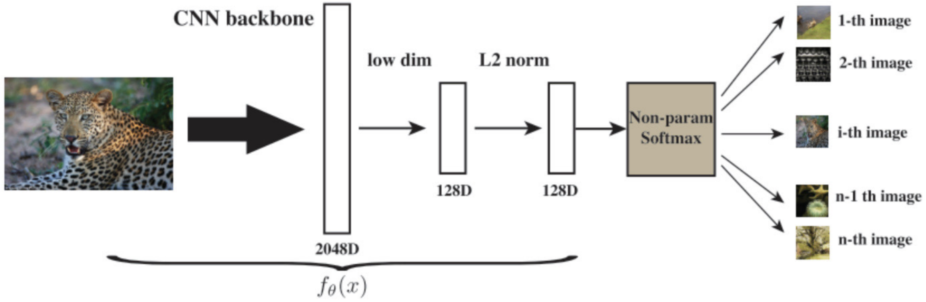
Pros

- Representations are invariant to desired transformations
- Requires preservation of fine-grained information

Cons

- Choosing the augmentations is important
- Exemplar based: images of the same class or instance are negatives
  - Nothing prevents it from focusing on the background
- Original formulation is not scalable (number of "classes" = dataset size)

Non-Parametric Classifier



Self-supervised learning as image instance-level discrimination

$$\mathcal{L}_{\text{non-param-softmax}}(q) = -\log \frac{\exp(q^\top k_q)}{\sum_{i \in N} \exp(q^\top k_i)}$$

The learning objective focuses now entirely on feature representation, instead of class-specific representations

---

Z. Wu et al., Unsupervised Feature Learning via Non-Parametric Instance Discrimination, CVPR 2018

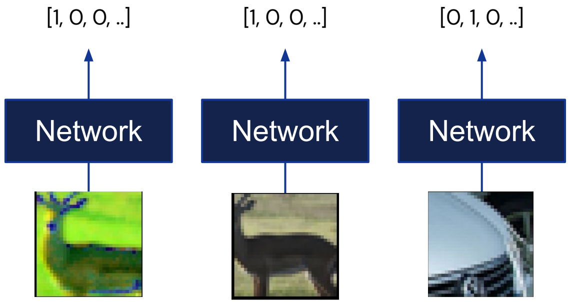
["Representation Learning with Contrastive Predictive Coding", van den Oord et al. 18]

Exemplar ConvNets via metric learning

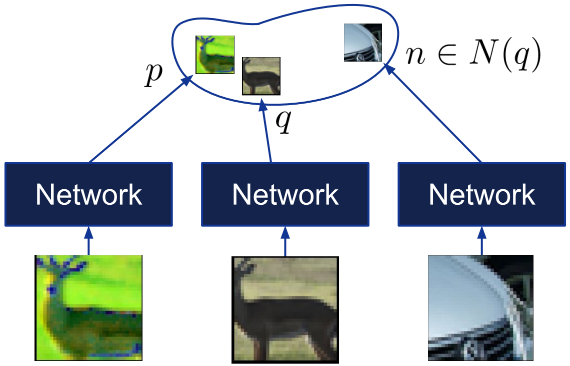
Exemplar ConvNets are not scalable (number of "classes" = number of training images)

- Reformulate in terms of metric learning
- Traditional losses such as contrastive or triplet ["Multi-task self-supervised visual learning", Doersch and Zisserman 17], ["HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips", Miech et al. 19]
- Recently popular: InfoNCE ["Representation Learning with Contrastive Predictive Coding", van den Oord et al. 18]
  - Used by many recent methods: CPC, AMDIM, SimCLR, MoCo, ..

Classification



Metric learning



SimCLR

[Chen et al, A simple framework for contrastive learning of visual representations, ICML'20]

Maximizing the agreement of representations under data transformation, using a contrastive loss in the latent/feature space.

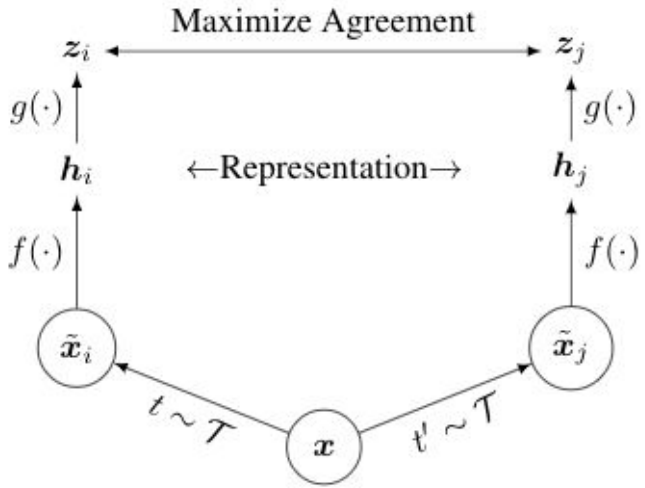
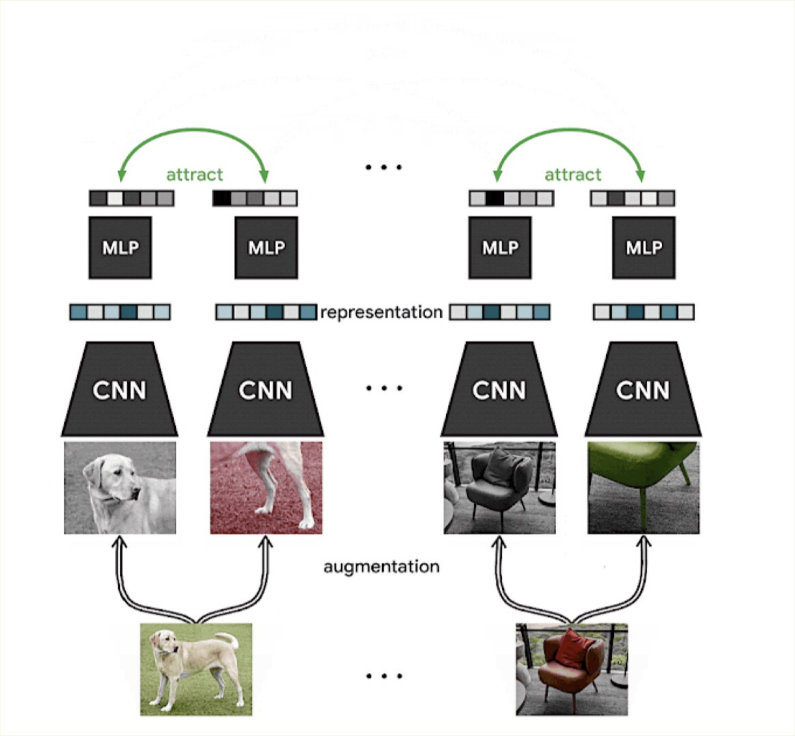
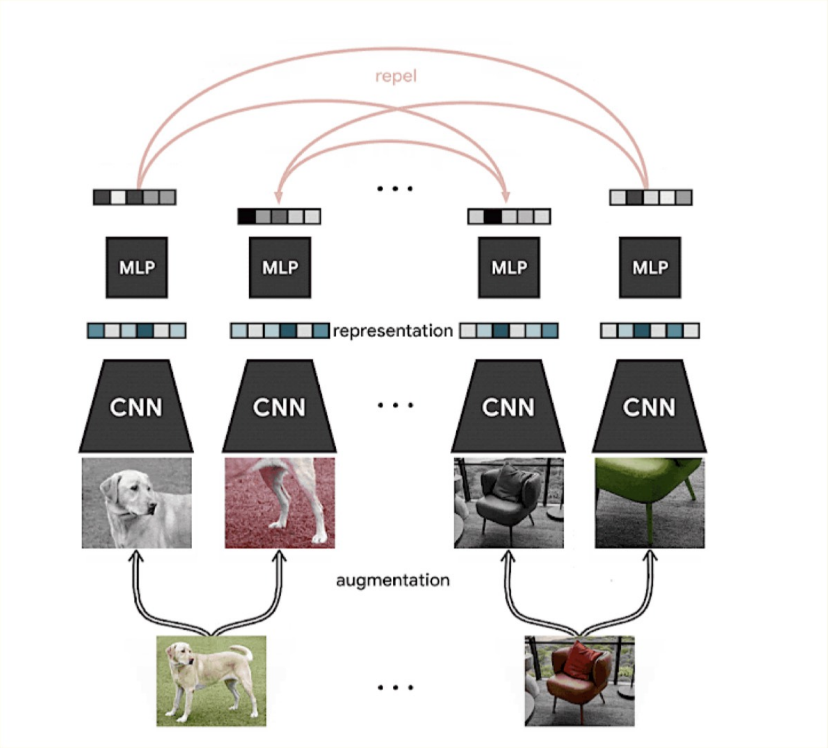


Figure 2. A framework for contrastive representation learning. Two separate stochastic data augmentations $t, t' \sim \mathcal{T}$ are applied to each example to obtain two correlated views. A base encoder network $f(\cdot)$ with a projection head $g(\cdot)$ is trained to maximize agreement in *latent representations* via a contrastive loss.

SimCLR - Modern framework for Contrastive Learning



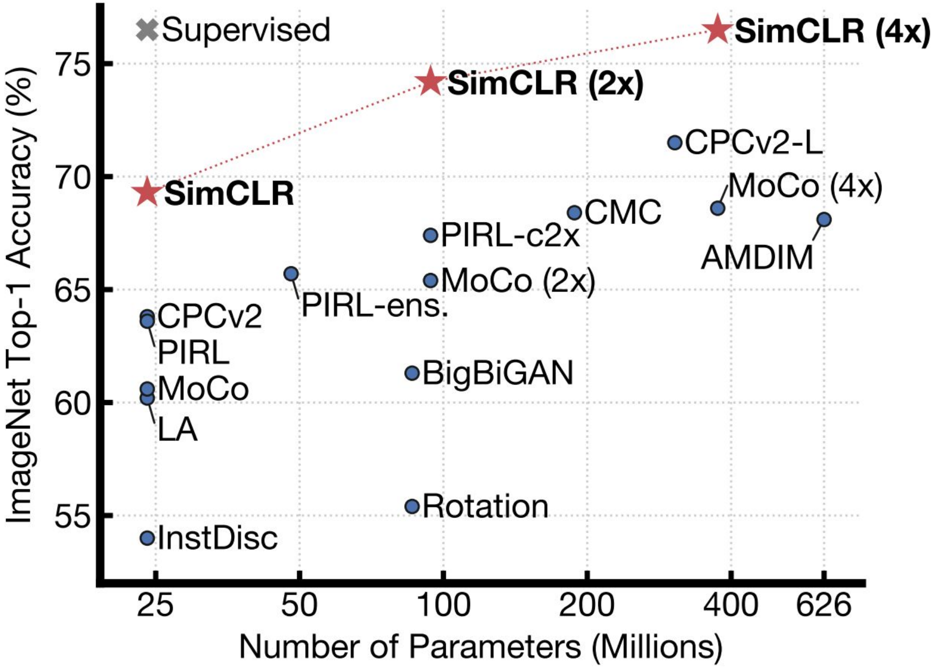
SimCLR - Modern framework for Contrastive Learning



SimCLR - Modern framework for Contrastive Learning

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

Training linear classifier on SimCLR features



Train feature encoder on **ImageNet** (entire training set) using SimCLR.

Freeze feature encoder, train a linear classifier on top with labeled data.

Source: [Chen et al., 2020]()

Semi-supervised learning on SimCLR features

Method	Architecture	Label fraction 1%	10%
		Top 5	
Supervised baseline	ResNet-50	48.4	80.4
Methods using other label-propagation:			
Pseudo-label	ResNet-50	51.6	82.4
VAT+Entropy Min.	ResNet-50	47.0	83.4
UDA (w. RandAug)	ResNet-50	-	88.5
FixMatch (w. RandAug)	ResNet-50	-	89.1
S4L (Rot+VAT+En. M.)	ResNet-50 (4×)	-	91.2
Methods using representation learning only:			
InstDisc	ResNet-50	39.2	77.4
BigBiGAN	RevNet-50 (4×)	55.2	78.8
PIRL	ResNet-50	57.2	83.8
CPC v2	ResNet-161(∗)	77.9	91.2
SimCLR (ours)	ResNet-50	75.5	87.8
SimCLR (ours)	ResNet-50 (2×)	83.0	91.2
SimCLR (ours)	ResNet-50 (4×)	**85.8**	**92.6**

Table 7. ImageNet accuracy of models trained with few labels.

Train feature encoder on **ImageNet** (entire training set) using SimCLR.

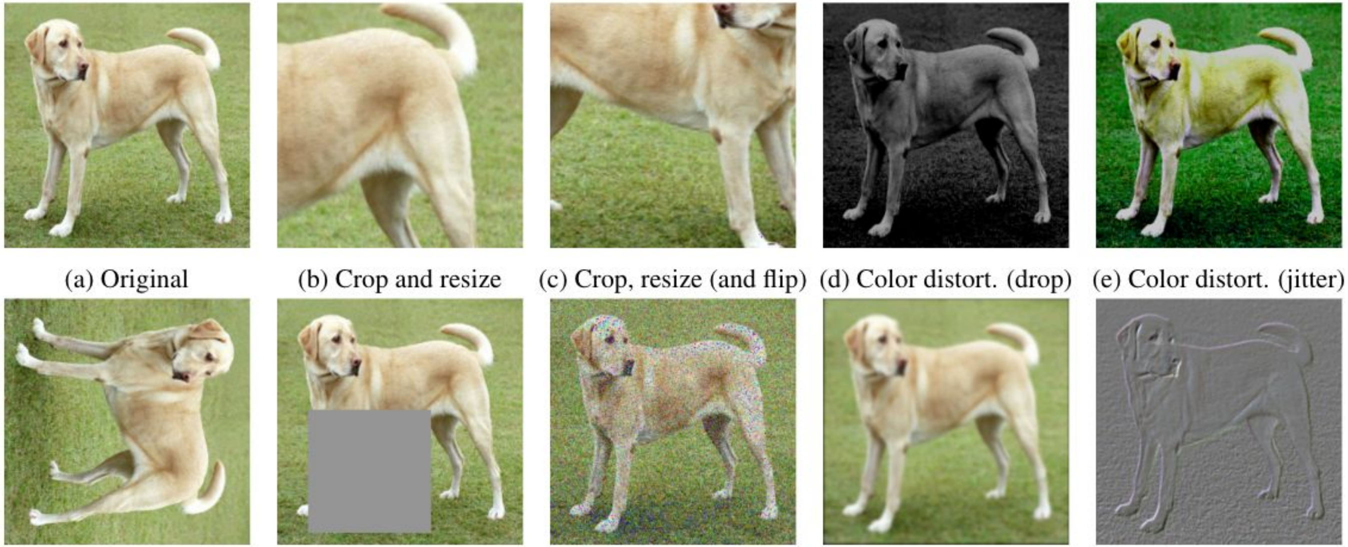
Finetune the encoder with 1% / 10% of labeled data on ImageNet.

Source: Chen et al., 2020

Google Research

A set of transformations studied in SimCLR

Systematically study a set of augmentation



(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

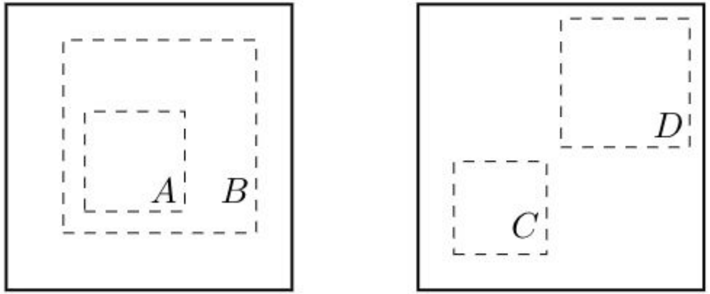
* Note that we only test these for ablation, the augmentation policy used to train our models only involves random crop (with flip and resize) + color distortion + Gaussian blur.

[Figures from SimCLR paper]

Random cropping gives the major learning signal

Simply via Random Crop (with resize to standard size), we can mimic (1) global to local view prediction, and (2) neighboring view prediction.

This simple transformation defines a family of predictive tasks.



(a) Global and local views. (b) Adjacent views.

Figure 3. By randomly cropping and resizing images (solid rectangles) to a standard size, we sample contrastive prediction tasks that mimic global to local view ($B \rightarrow A$) or neighbouring view ($D \rightarrow C$) prediction.

[Figures from SimCLR paper]

Google Research

Composition of augmentations are crucial

Composition of crop and color stands out!

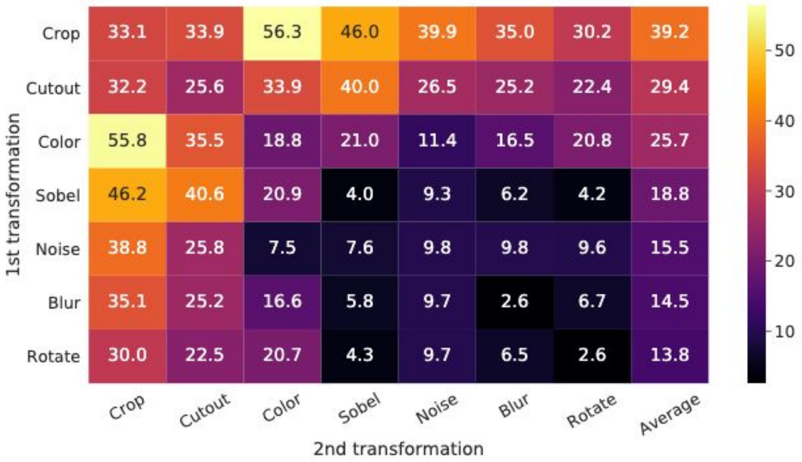
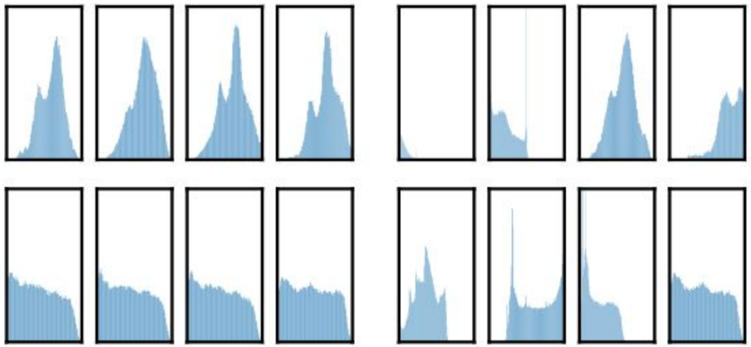


Figure 5. Linear evaluation (ImageNet top-1 accuracy) under individual or composition of data augmentations, applied only to one branch. For all columns but the last, diagonal entries correspond to single transformation, and off-diagonals correspond to composition of two transformations (applied sequentially). The last column reflects the average over the row.



(a) Without color distortion. (b) With color distortion.

Figure 6. Histograms of pixel intensities (over all channels) for different crops of two different images (i.e. two rows). The image for the first row is from Figure 4. All axes have the same range.

[Figures from SimCLR paper]

SimCLR design choices: large batch size

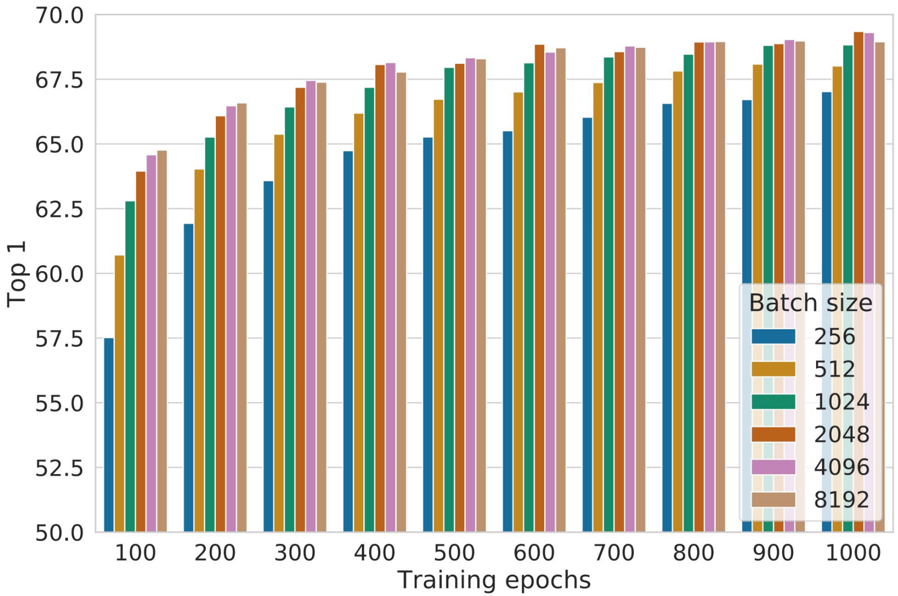


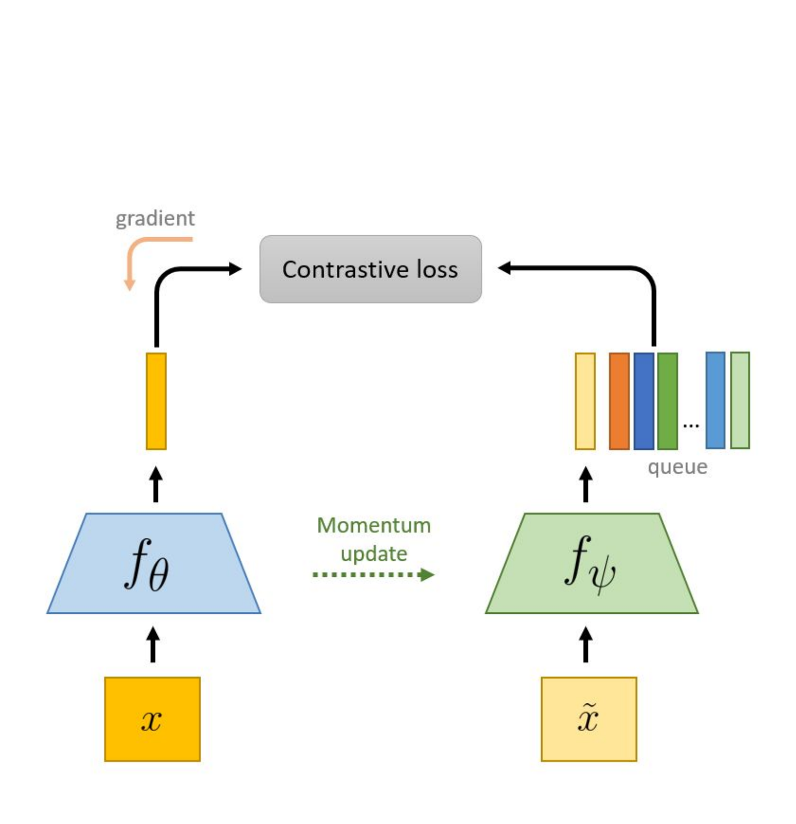
Figure 9. Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch.[10]

Large training batch size is crucial for SimCLR!

Large batch size causes large memory footprint during backpropagation: requires distributed training on TPUs (ImageNet experiments)

Source: [Chen et al., 2020]

Contrastive: *MoCo*



- **Contrastive learning loss (InfoNCE)**
- **The momentum encoder produces a memory bank of negatives and positives stored as a queue**
- **The momentum encoder f_ψ is slowly pursuing f_θ via exponential moving average (momentum) update:**

$$\psi \leftarrow m\psi + (1-m)\theta$$

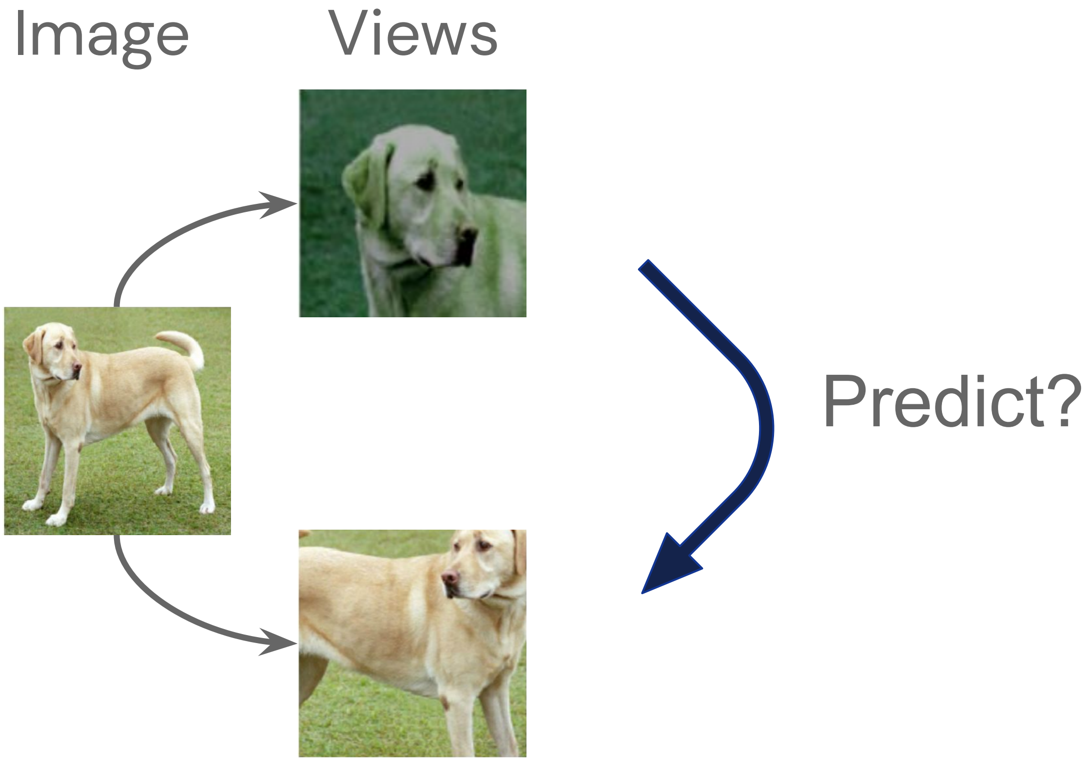
- **Elegant and effective solution for large dictionaries**

f_θ f_ψ : encoder (ResNet-50);

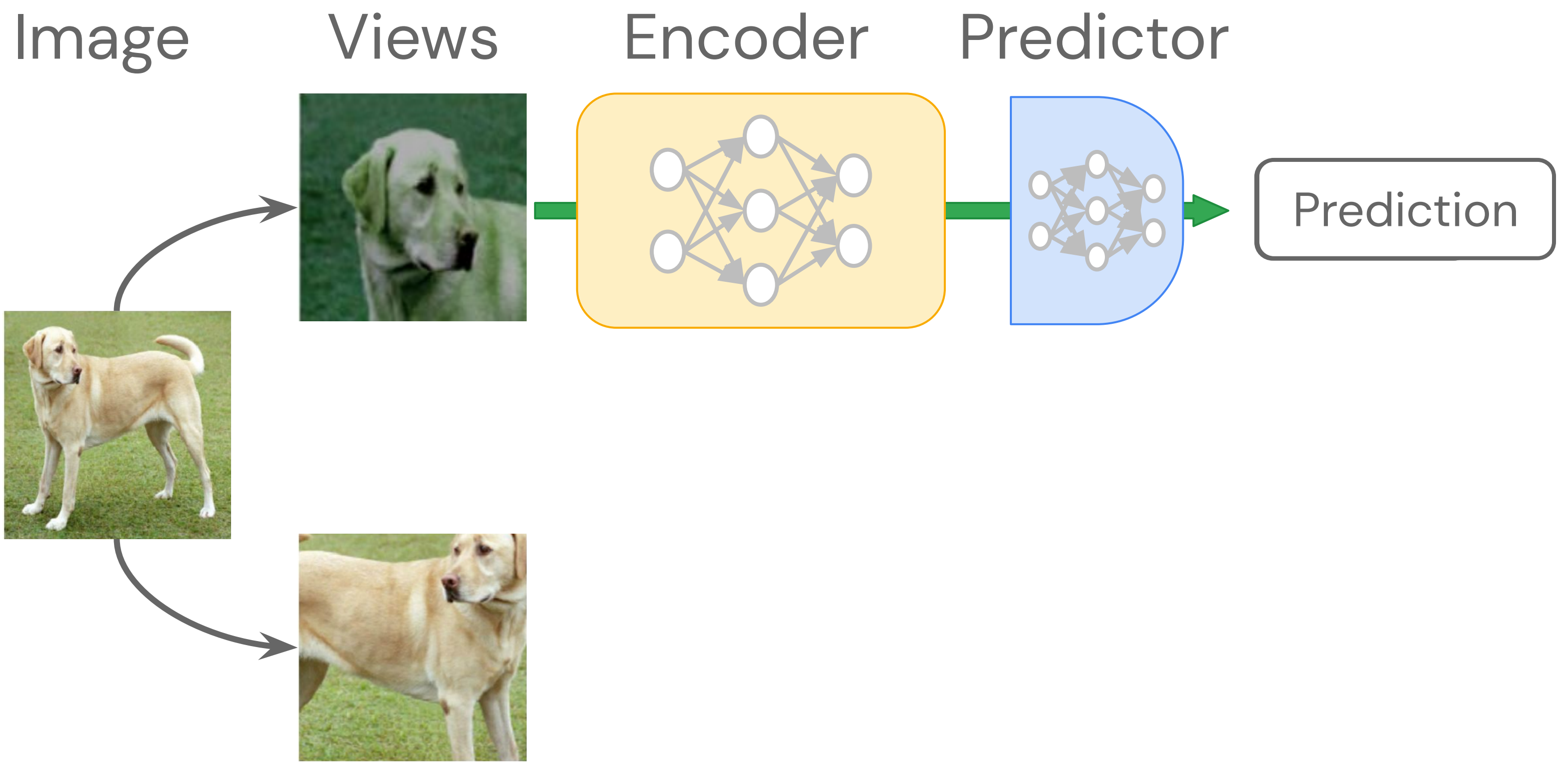
He et al., Momentum Contrast for Unsupervised Visual Representation Learning, CVPR 2020

The regression/self-distillation SSL paradigm

The regression/self-distillation SSL paradigm

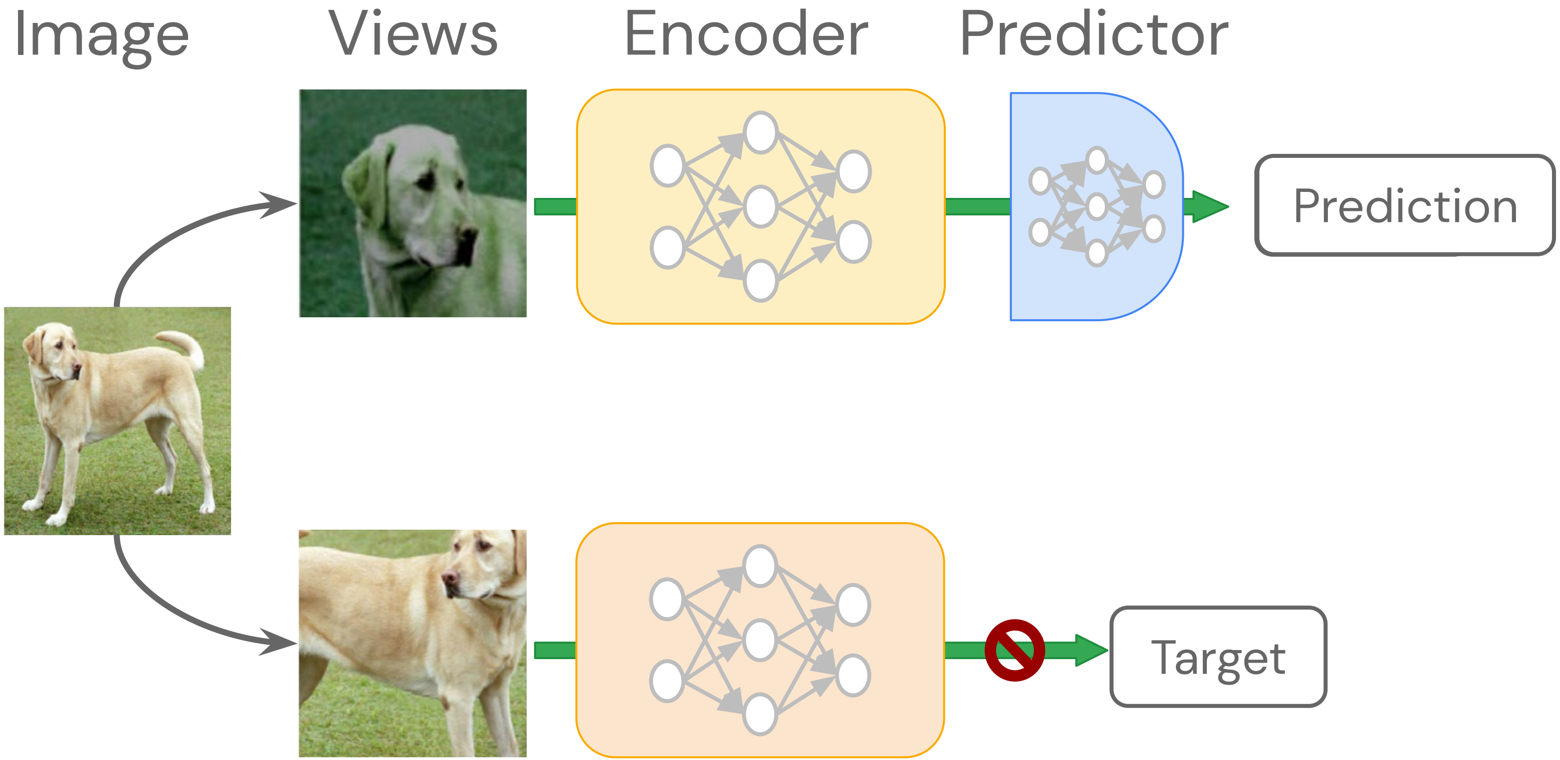


BYOL: Bootstrap your own latent
Grill et al., 2020



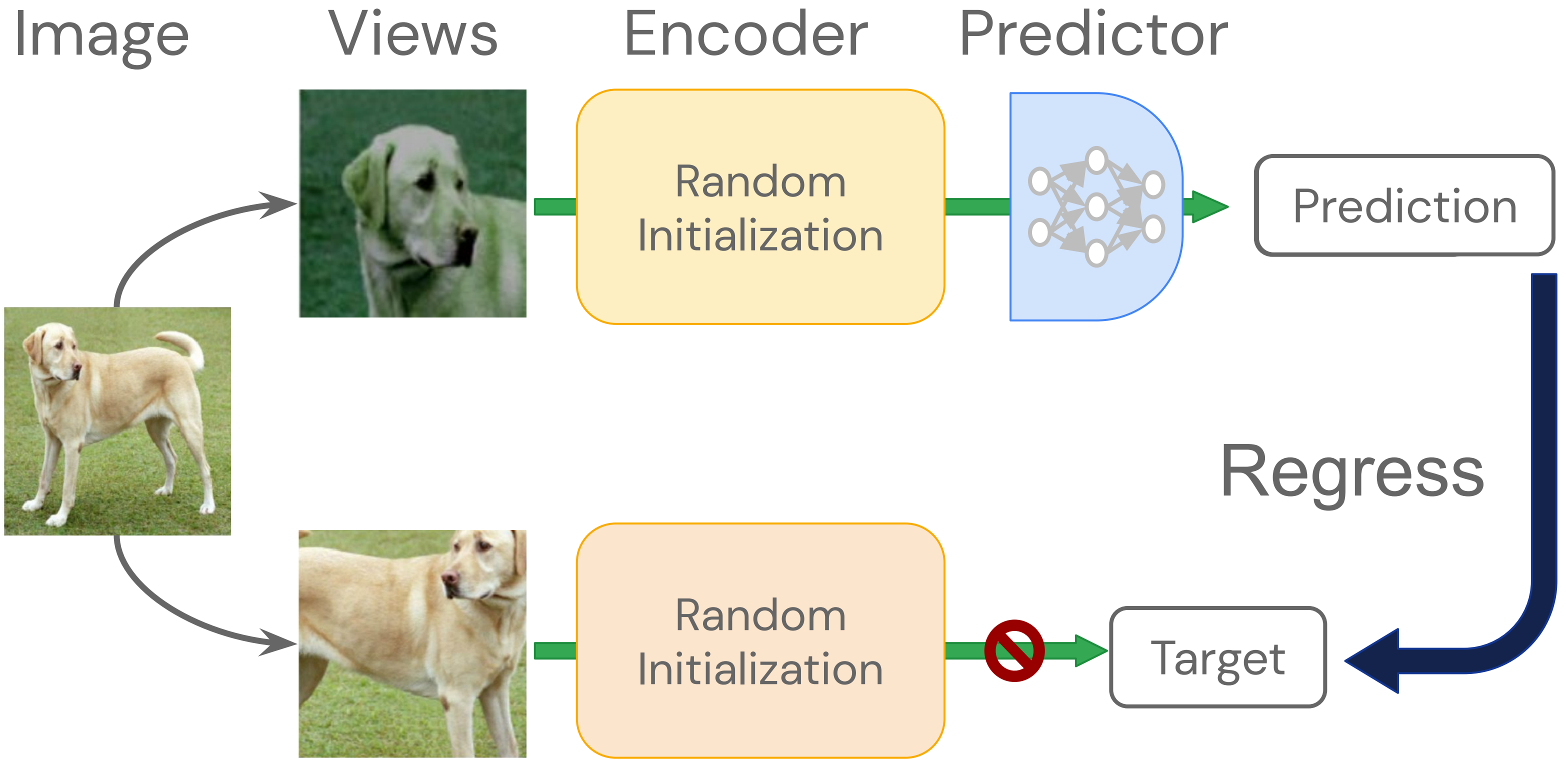
BYOL: Bootstrap your own latent

Grill et al., 2020

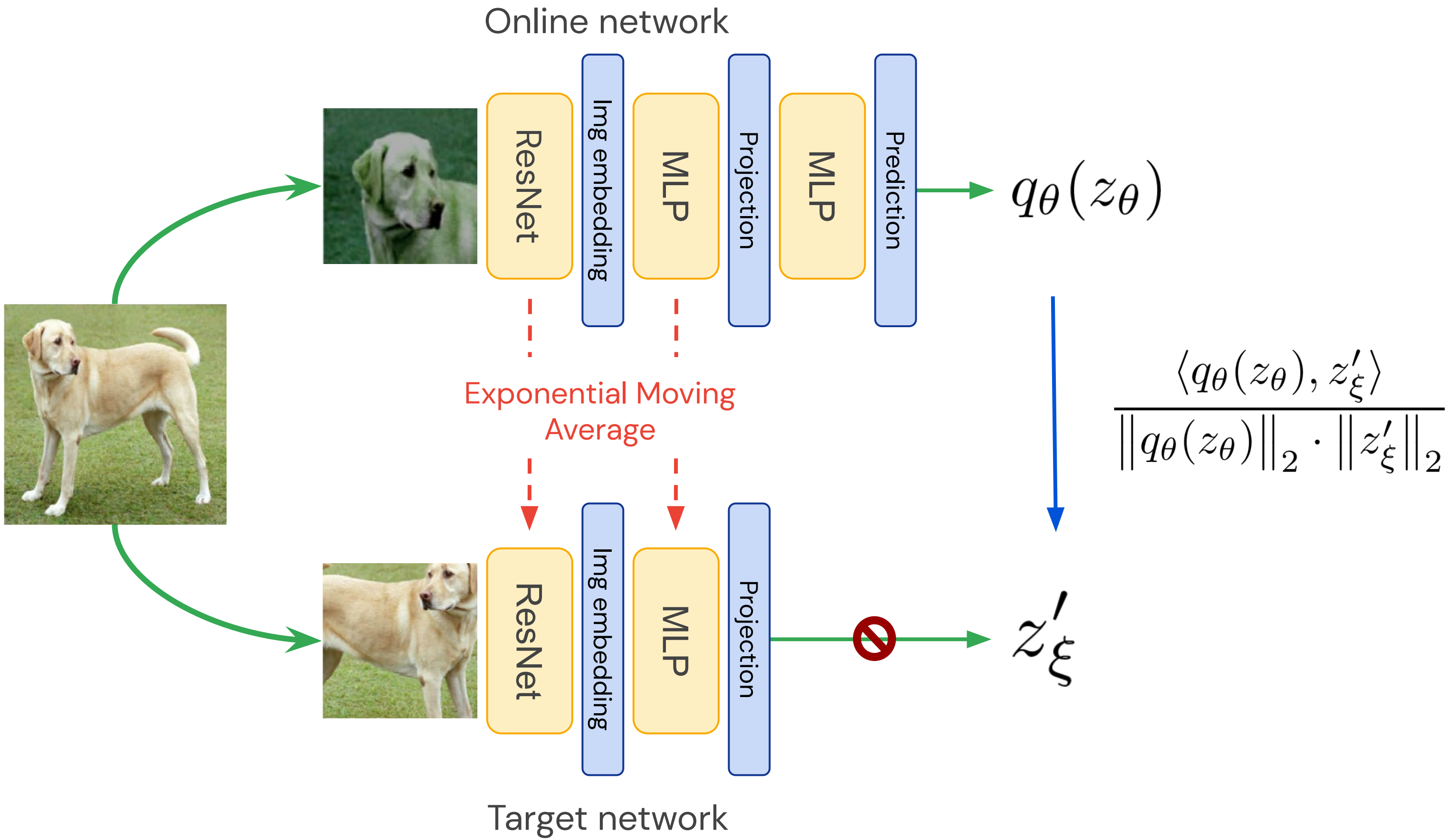


BYOL: Bootstrap your own latent

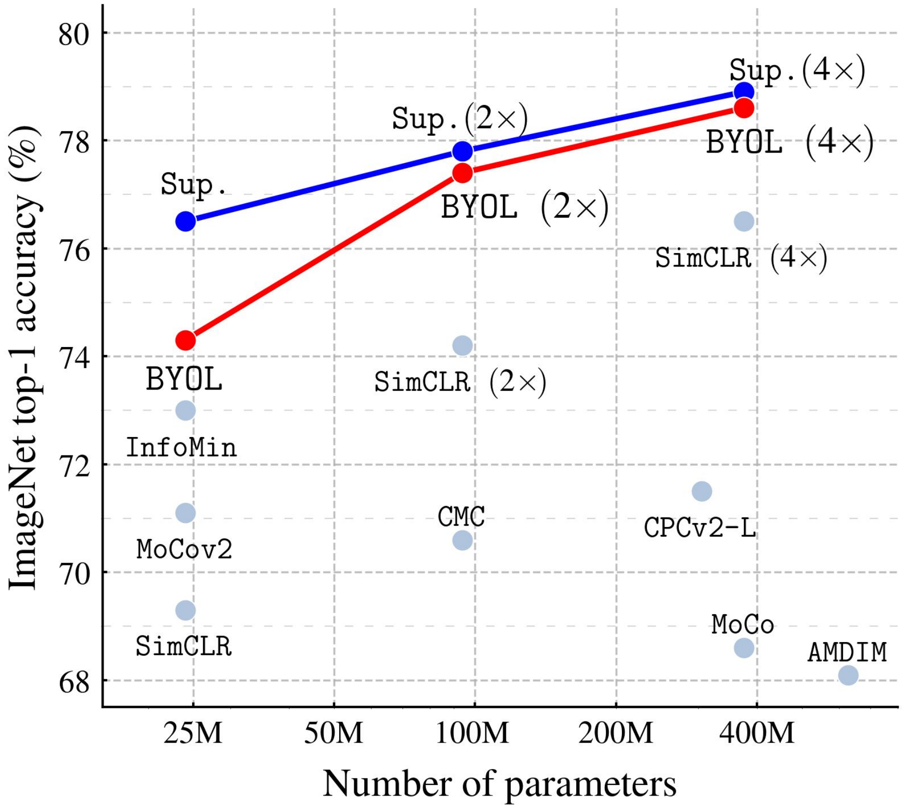
Grill et al., 2020



BYOL Architecture



Linear Evaluation Performance on ImageNet



Note: these supervised baselines are from SimCLR (Chen & Hinton, ICML 2020)

CPCv2: van den Oord et al., *Representation learning with contrastive predictive coding*. 2018

AMDIM: Bachman et al., *Learning representations by maximizing mutual information across views*. 2019

CMC: Tian et al.,*Contrastive multiview coding*. 2019.

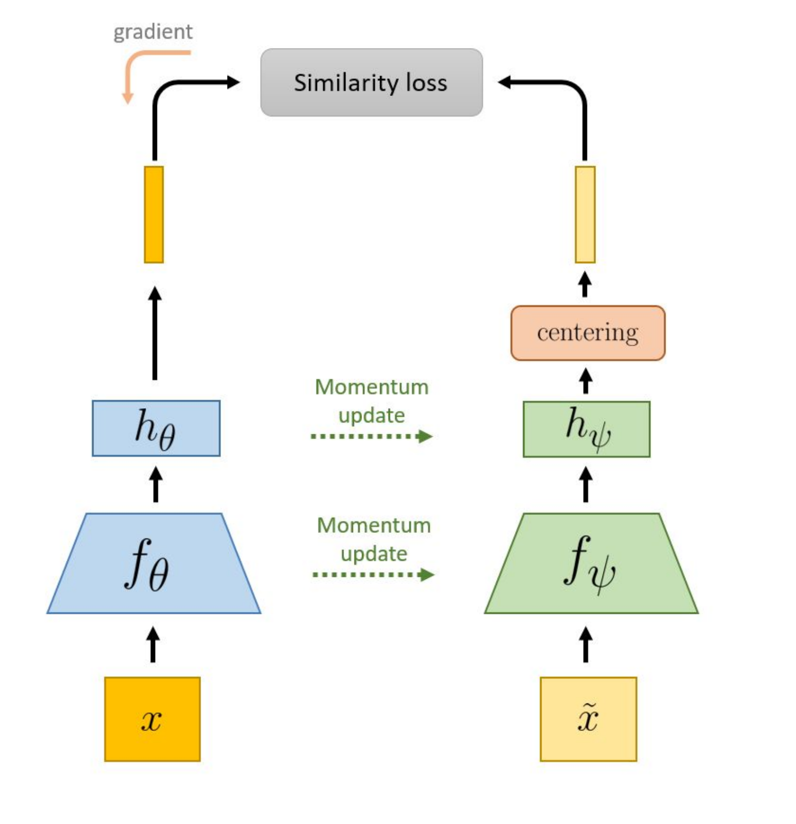
MoCo: He et al., *Momentum contrast for unsupervised visual representation learning*. 2019

InfoMin: Tian et al., *What makes for good views for contrastive learning*. 2020

MoCov2: Jain et al., *Improved baselines with momentum contrastive learning*. 2020

SimCLR: Chen et al., *A simple framework for contrastive learning of visual representations*. 2020

Self-distillation: *DINO*



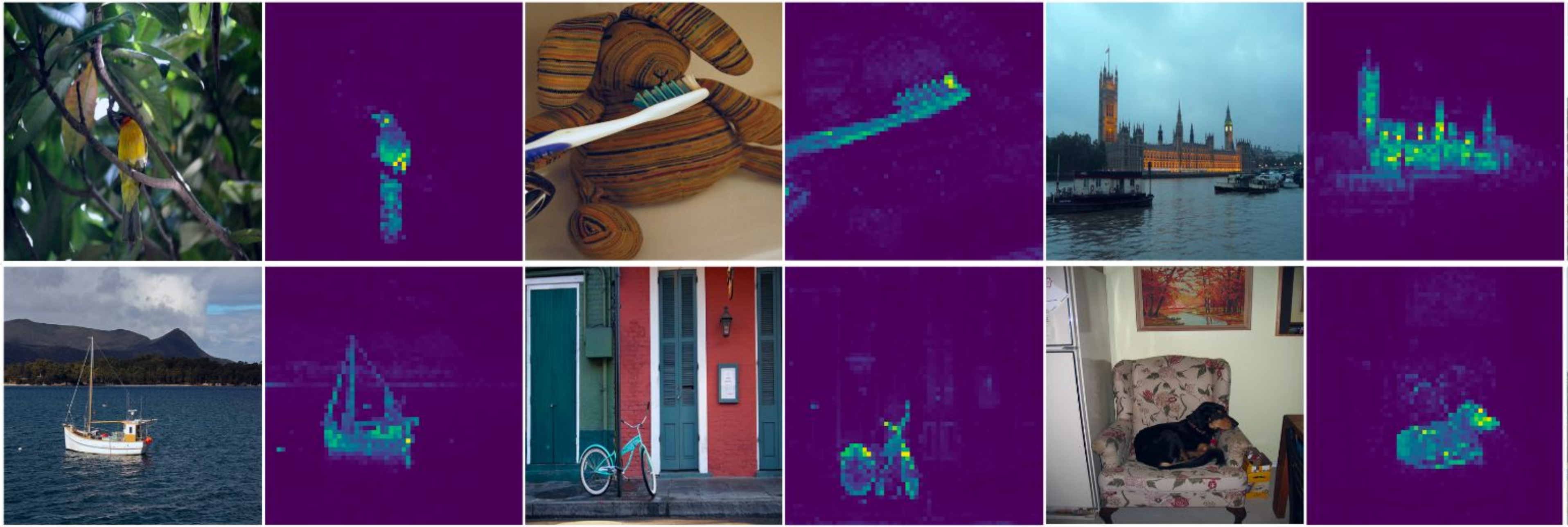
Main idea: No prediction head; post-processing of teacher outputs to avoid feature collapse

- **Centering by subtracting the mean feature:** prevents collapsing to constant 1-hot targets
- **Sharpening by using low softmax temperature:** prevents collapsing to a uniform target vector
- Cross-entropy loss

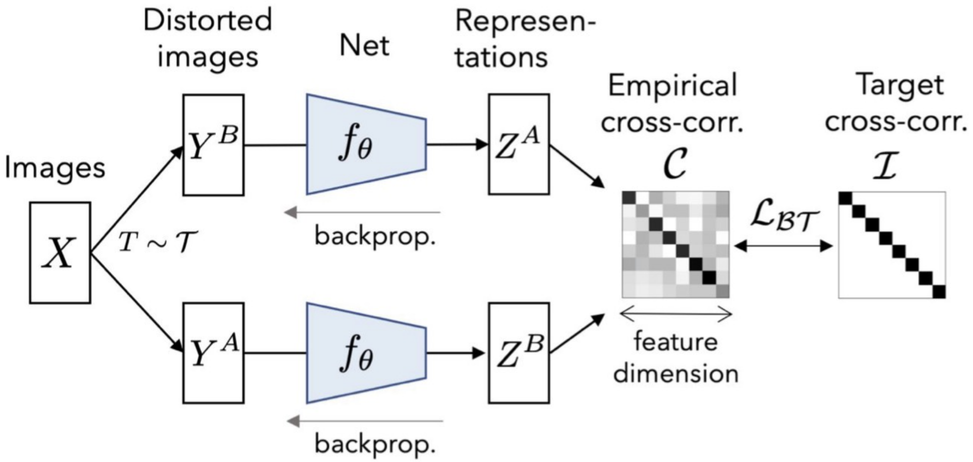
f_θ f_ψ : encoder (ViT, ResNet-50); h_θ h_ψ : projection (MLP)

Caron et al., Emerging Properties in Self-Supervised Vision Transformers, ICCV 2021

Attention weights between `[CLS]` token and patch tokens:



Barlow Twins (Zbontar et al. 2021)



$$\mathcal{L}_{\mathrm{BT}} = \underbrace{\sum_i (1 - \mathcal{C}_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{i \neq j} \mathcal{C}_{ij}^2}_{\text{redundancy reduction term}}$$

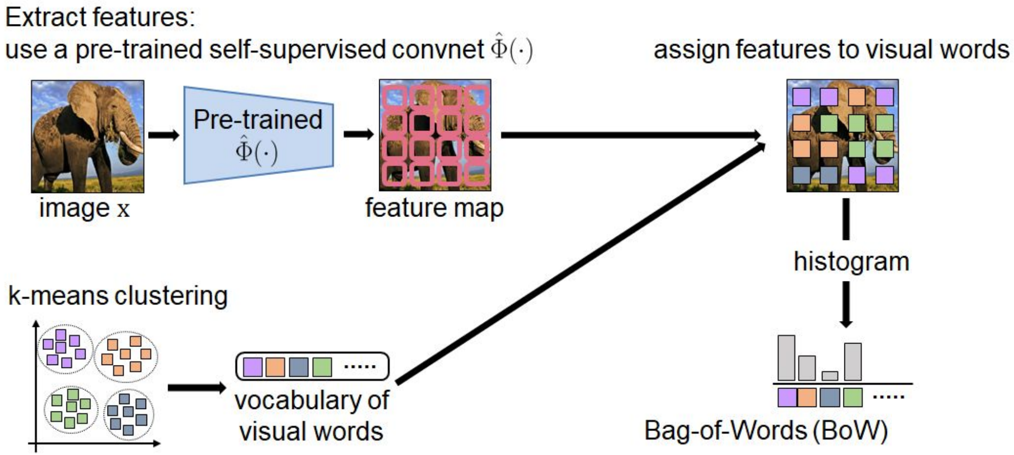
$$\text{where } \mathcal{C}_{ij} = \frac{\sum_b \mathbf{z}^A_{b,i} \mathbf{z}^B_{b,j}}{\sqrt{\sum_b (\mathbf{z}^A_{b,i})^2} \sqrt{\sum_b (\mathbf{z}^B_{b,j})^2}}$$

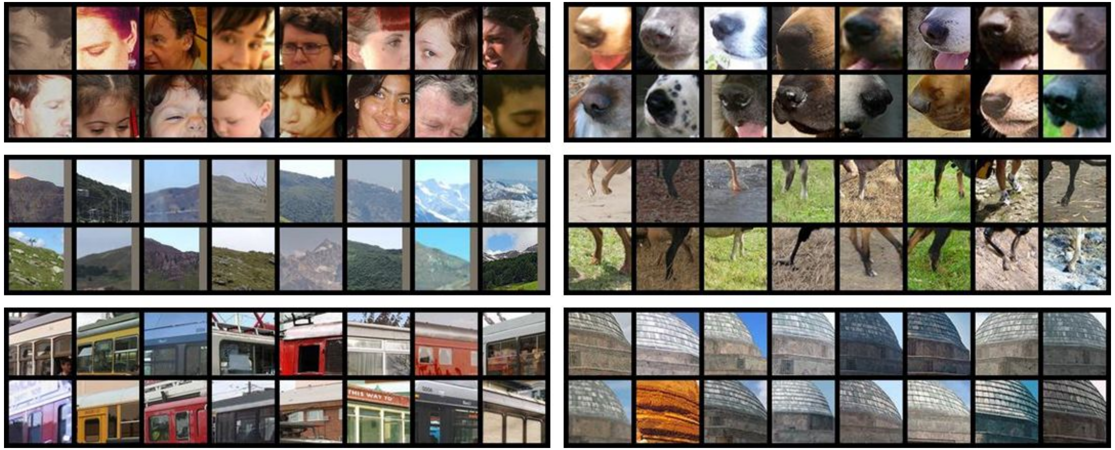
(Zbontar et al. 2021)

["Learning representations by predicting bags of visual words", Gidaris et al. 20]

Predicting bag-of-words

Bag-of-words reminder

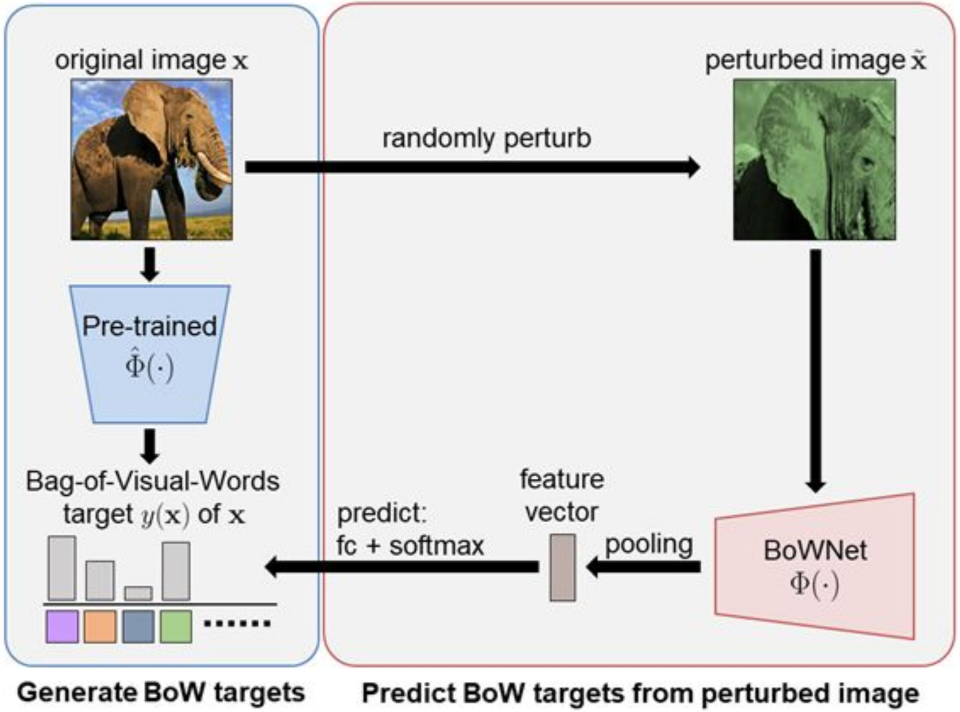




- Loses low-level details
- Encodes mid/high-level concepts

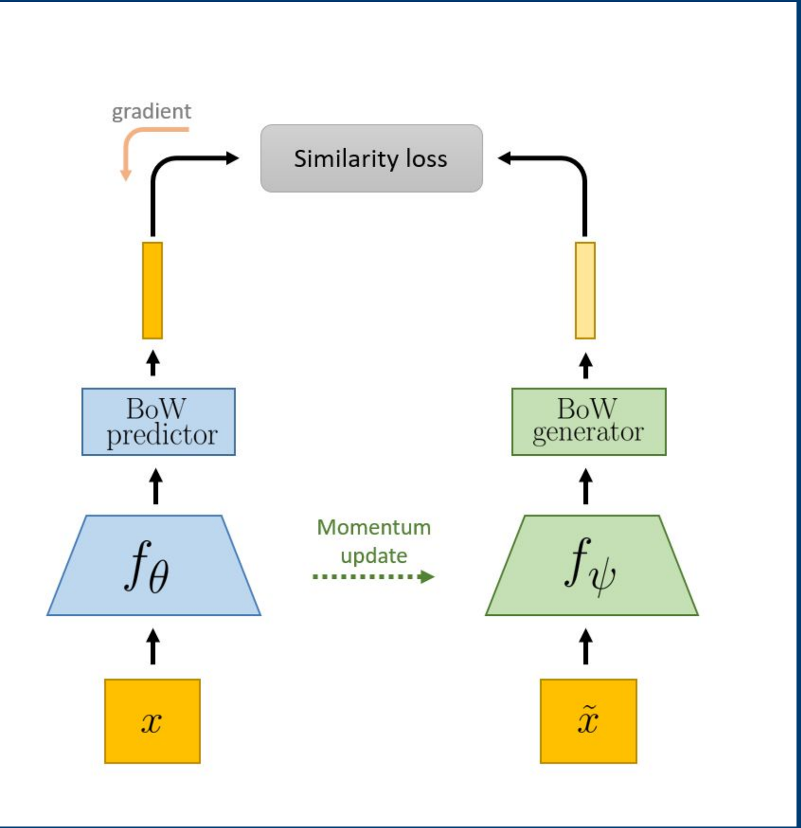
["Learning representations by predicting bags of visual words", Gidaris et al. 20]

Predicting bag-of-words



Inspired by NLP: targets = discrete concepts (words)

Self-distillation: *OBoW*



Online Bag-of-Visual-Words Generation (OBoW)

- **Teacher:** produce vocabulary of local features and BoW target vectors
- **Student:** predict teacher BoW vectors, given as input a different random view of the same image
- Mitigates feature collapse
- Focus on local feature representations
- Cross-entropy loss

f_θ f_ψ : encoder (ResNet-50);

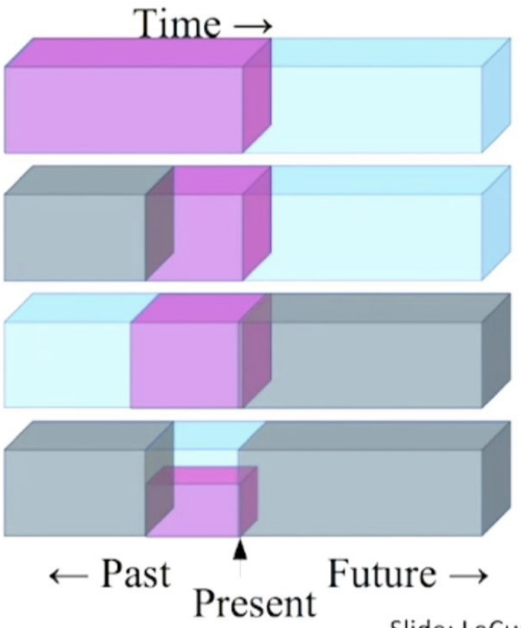
Gidaris et al., OBoW: Online Bag-of-Visual-Words Generation for Self-supervised Learning, CVPR 2021

Method	Epochs	Batch	Linear Classification			VOC Detection			Semi-supervised learning	
			ImageNet	Places205	VOC07	AP^{50}	AP^{75}	AP^{all}	1% Labels	10% Labels
Supervised	100	256	76.5	53.2	87.5	81.3	58.8	53.5	48.4	80.4
BoWNet [26]	325	256	62.1	51.1	79.3	81.3	61.1	55.8	-	-
PCL [48]	200	256	67.6	50.3	85.4	-	-	-	75.3	85.6
MoCo v2 [35]	200	256	67.5	-	-	82.4	63.6	57.0	-	-
SimCLR [9]	200	4096	66.8	-	-	-	-	-	-	-
SwAV [8]	200	256	72.7	$56.2^{\dagger}$	$87.2^{\dagger}$	$81.8^{\dagger}$	$60.0^{\dagger}$	$54.4^{\dagger}$	$76.7^{\dagger}$	$88.7^{\dagger}$
BYOL [33]	300	4096	72.5	-	-	-	-	-	-	-
OBoW (Ours)	200	256	**73.8**	**56.8**	**89.3**	**82.9**	**64.8**	**57.9**	**82.9**	**90.7**
PIRL [51]	800	1024	63.6	49.8	81.1	80.7	59.7	54.0	57.2	83.8
MoCo v2 [35]	800	256	71.1	52.9	87.1	82.5	64.0	57.4	-	-
SimCLR [9]	1000	4096	69.3	53.3	86.4	-	-	-	75.5	87.8
BYOL [33]	1000	4096	74.3	-	-	-	-	-	78.4	89.0
SwAV [8]	800	4096	**75.3**	56.5	88.9	82.6	62.7	56.1	78.5	89.9

Self-prediction/masking SSL paradigm

The self-prediction/masking paradigm

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



Slide: LeCun

(Famous illustration from Yann LeCun)

The success of self-supervised methods in NLP, e.g., word2vec, is inspiring

Input: The man went to the [MASK]$_1$. He bought a [MASK]$_2$ of milk .
Labels: [MASK]$_1$ = store; [MASK]$_2$ = gallon

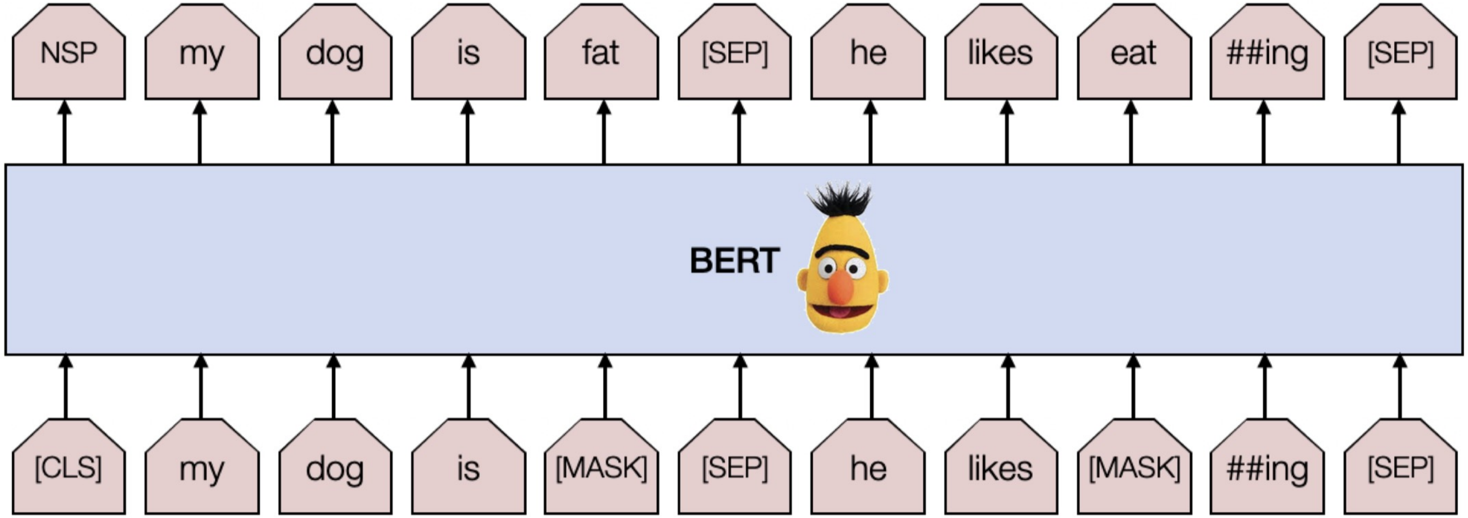
Missing word prediction task

Sentence A = The man went to the store.	**Sentence A** = The man went to the store.
Sentence B = He bought a gallon of milk.	**Sentence B** = Penguins are flightless.
Label = IsNextSentence	**Label** = NotNextSentence

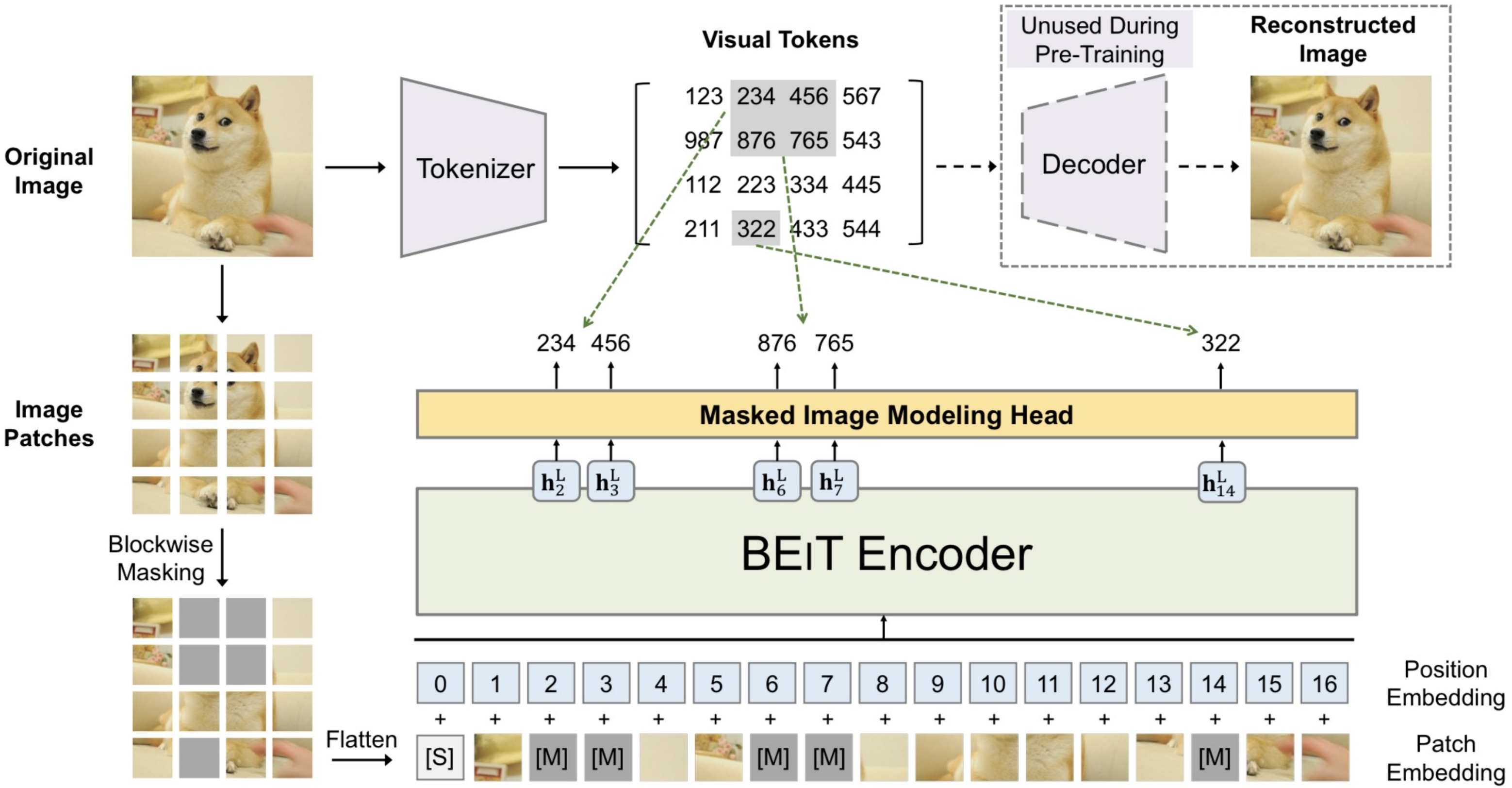
Next sentence prediction task

Mikolov et al., Efficient estimation of word representations in vector space, ArXiv 2013
Mikolov et al., Distributed representations of words and phrases and their compositionality, NeurIPS 2013
Devlin, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, ArXiv 2018

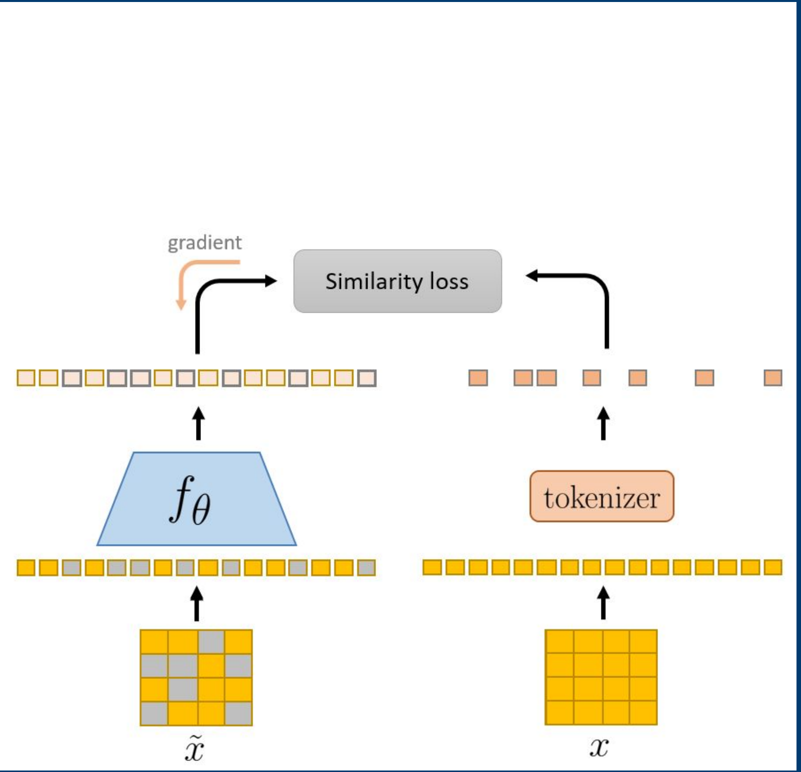
- BERT: encoder-*only* pre-training



BEiT: BERT Pre-Training of Image Transformers



Masked Image Modelling: *BEiT*



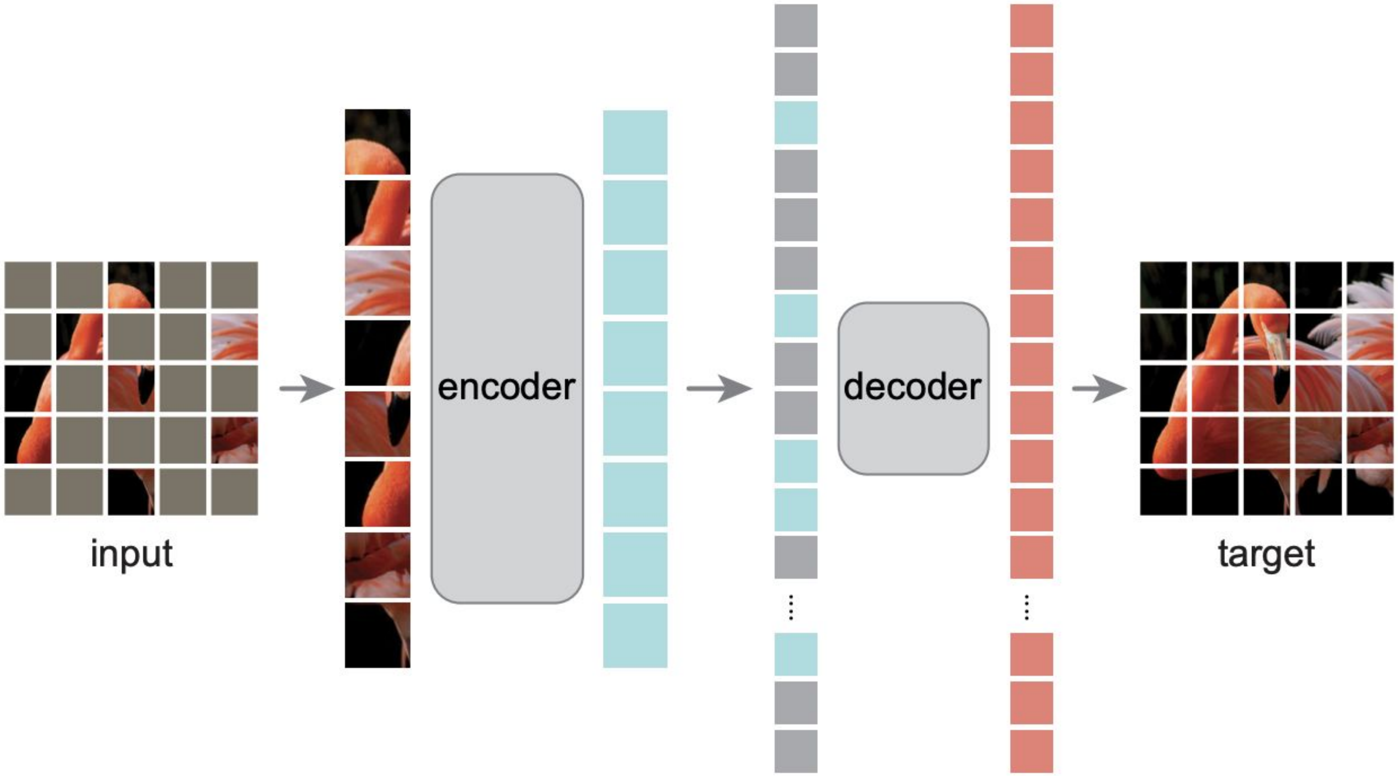
Main idea: pre-train ViTs by learning to predict tokens of masked patches

- **Mimicking practices from large language models (BERT)**
- **Learn to produce discrete visual tokens from masked input images**
- **Use learnable mask-token for masked patches**
- **Trained with cross-entropy loss over masked tokens**

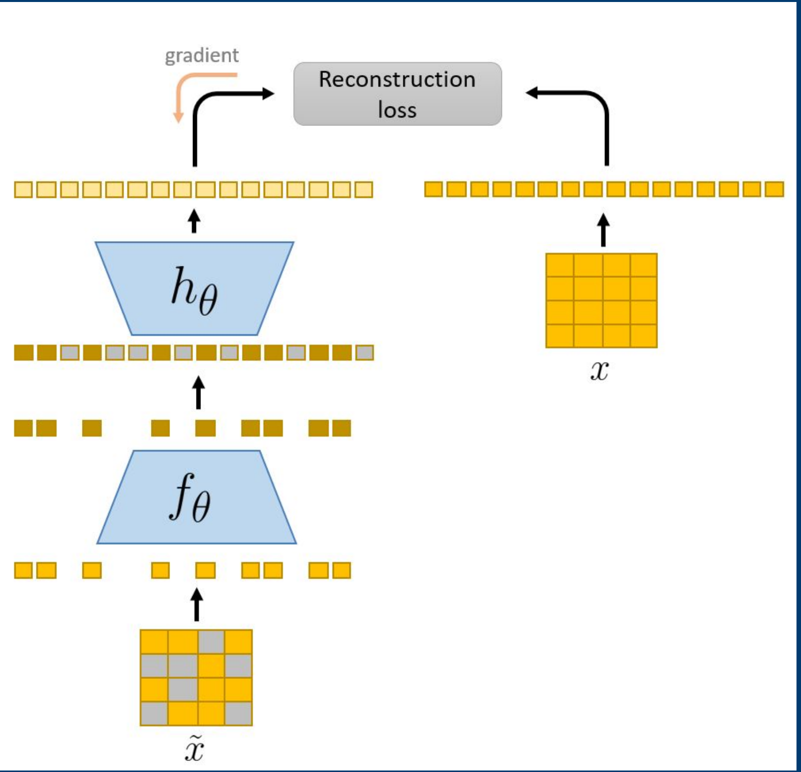
f_θ : encoder (ViT); $\mathrm{tokenizer}$: pretrained autoencoder (DALL-E)

Bao et al., BEiT: BERT Pre-Training of Image Transformers, ICLR 2022

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



Masked Image Modelling: *MAE*

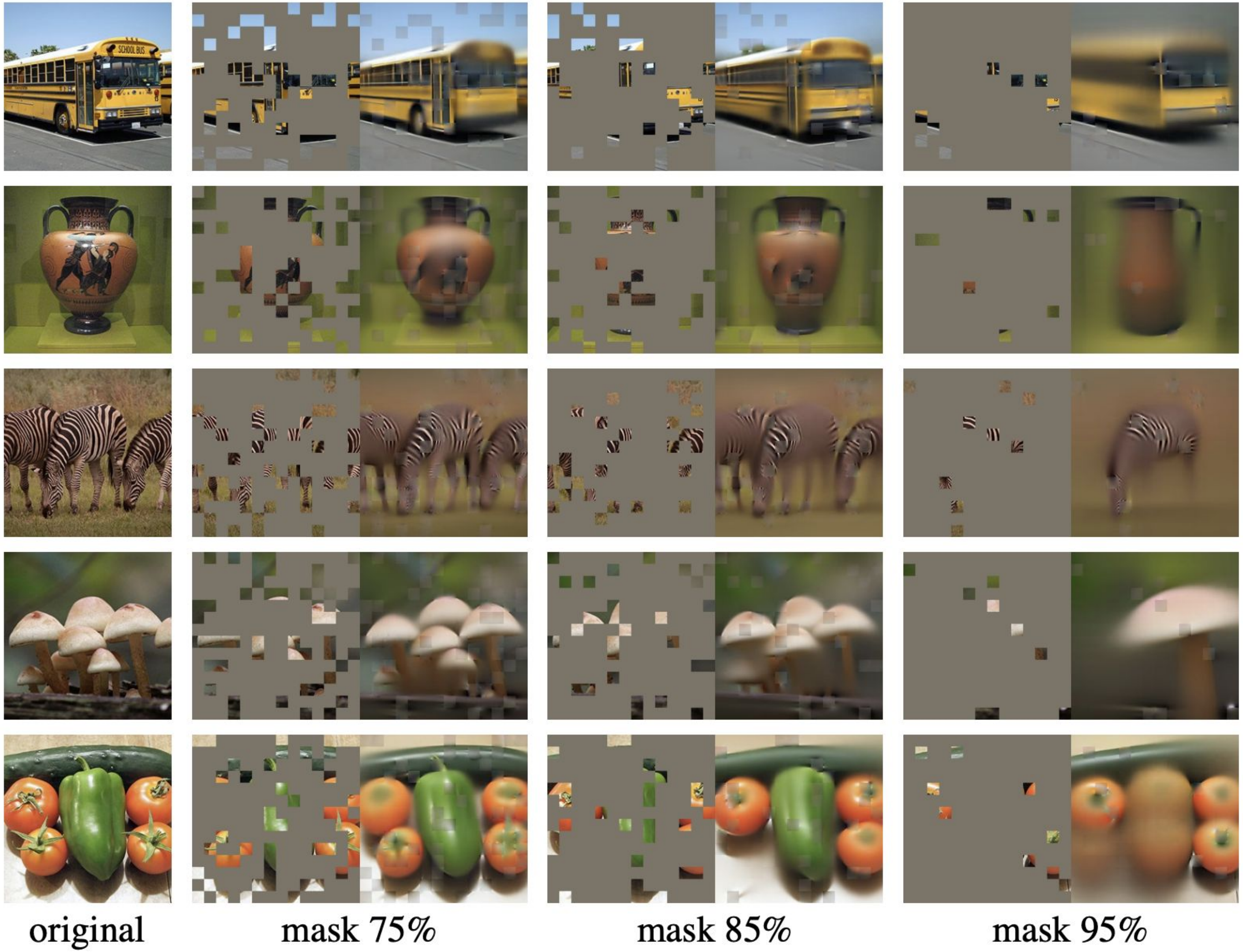


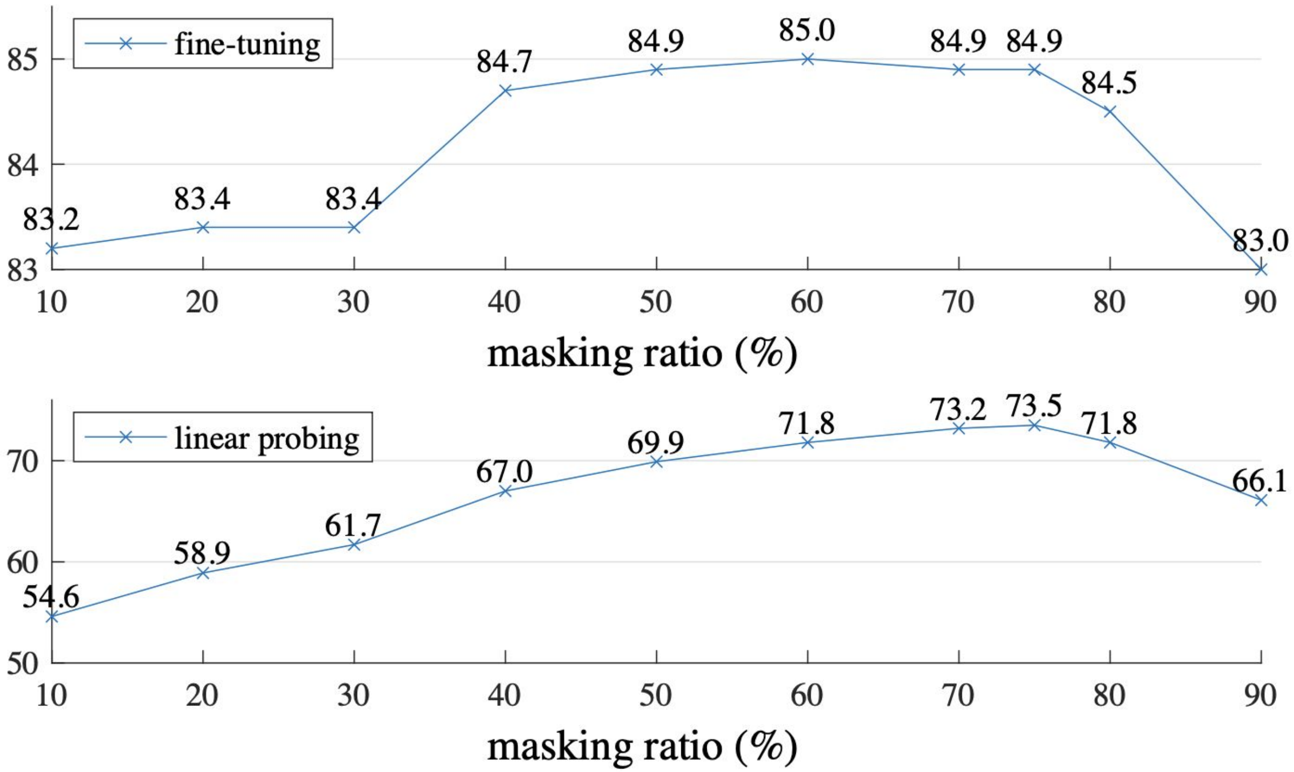
Main idea: learn to reconstruct masked pixels

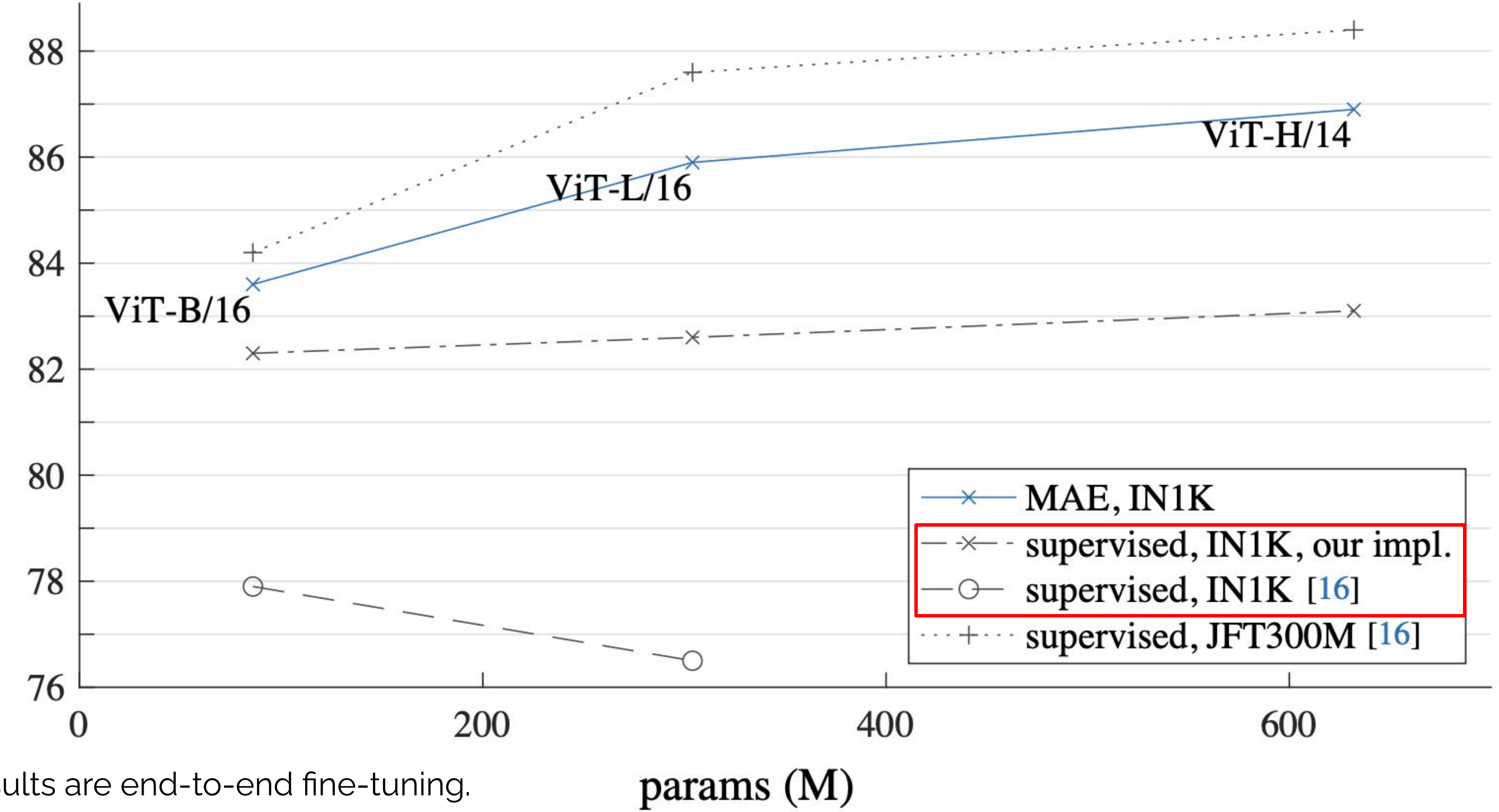
- Simplified MIM pipeline without pre-trained tokenizer nor data augmentation
- Encoder operates only on visible patches without mask tokens
- Lightweight ViT decoder (removed after pre-training)
- Aggressive masking (up to 75% of patches)
- Shines when fine-tuned on the downstream task

f_θ : encoder (ViT); h_θ : decoder (ViT)

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

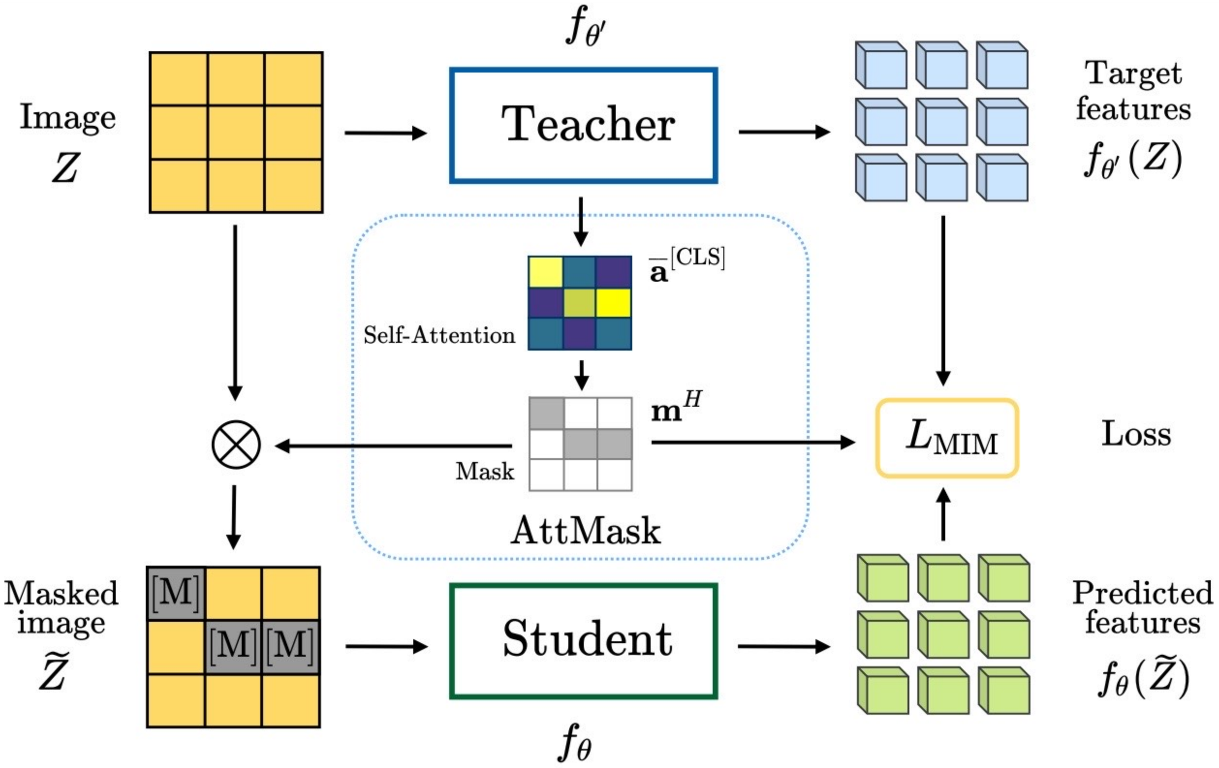

SCHOOL BUS
original
mask 75%
mask 85%
mask 95%


fine-tuning
85
84
83
83.2
83.4
83.4
84.7
84.9
85.0
84.9
84.9
84.5
83.0
10
20
30
40
50
60
70
80
90
masking ratio (%)
linear probing
70
60
50
54.6
58.9
61.7
67.0
69.9
71.8
73.2
73.5
71.8
66.1
10
20
30
40
50
60
70
80
90
masking ratio (%)

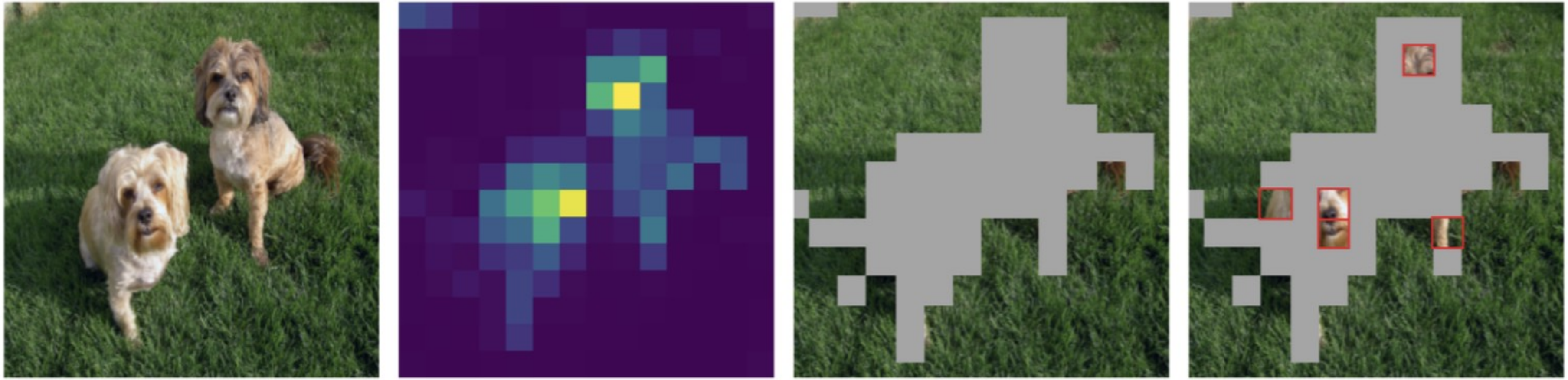


*Results are end-to-end fine-tuning.

What to Hide from Your Students: Attention-Guided Masked Image Modeling



(a) Input Image | (b) Random (30) | (c) Random (75) | (d) Block Wise | (e) Attention Map | (f) AttMask High | (g) AttMask Low



(a) Input Image (b) Attention Map (c) AttMask-High (d) AttMask-Hint

Table 2. Top-1 k-NN accuracy on ImageNet-1k validation for iBOT pre-training on different percentage (%) of ImageNet-1k. †: default iBOT masking strategy from BEiT [2]

% ImageNet-1k	5	10	20	100
Random Block-Wise†	15.7	31.9	46.7	71.5
AttMask-High (ours)	**17.5**	**33.8**	**49.7**	**72.5**

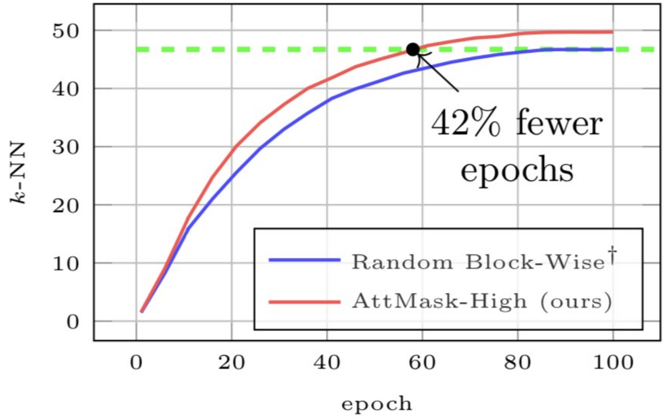
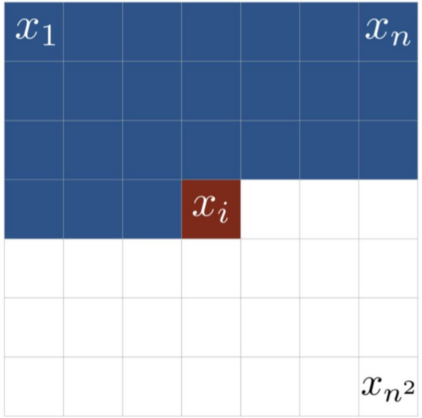


Fig. 4. Top-1 k-NN accuracy on ImageNet-1k validation for iBOT training *vs.* training epoch on 20% ImageNet training set. †: default iBOT masking strategy from BEiT [2]

The generative-based SSL paradigm

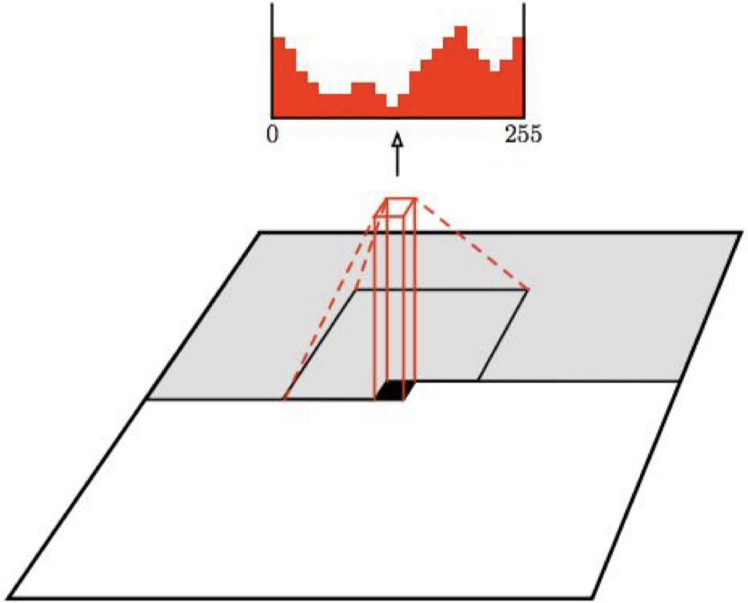
PixelRNN, PixelCNN (Oord et al. 2016)

$$p(X) = p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | x_1, \ldots, x_{i-1})$$

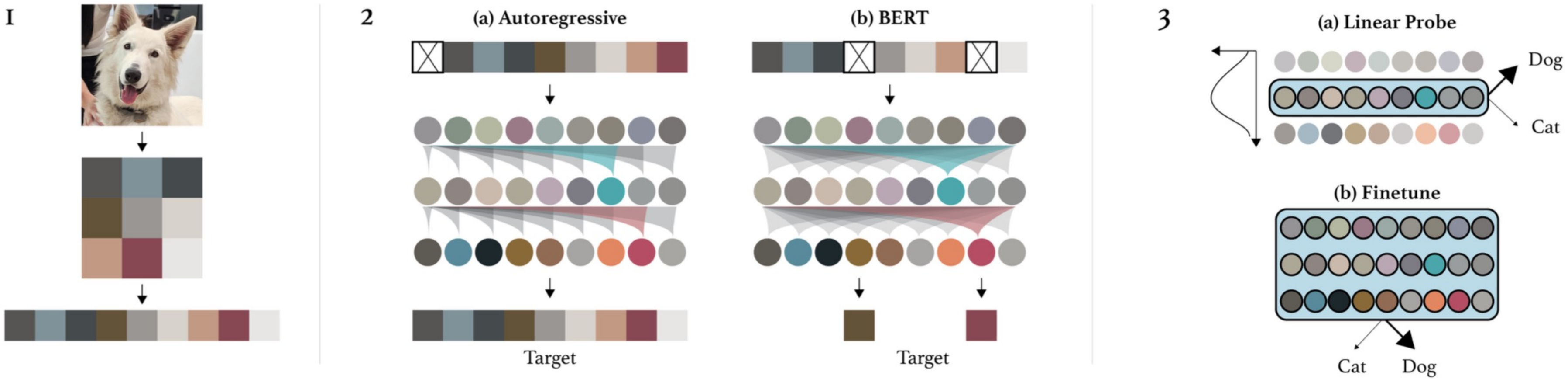


Raster scan order

Softmax loss at each pixel

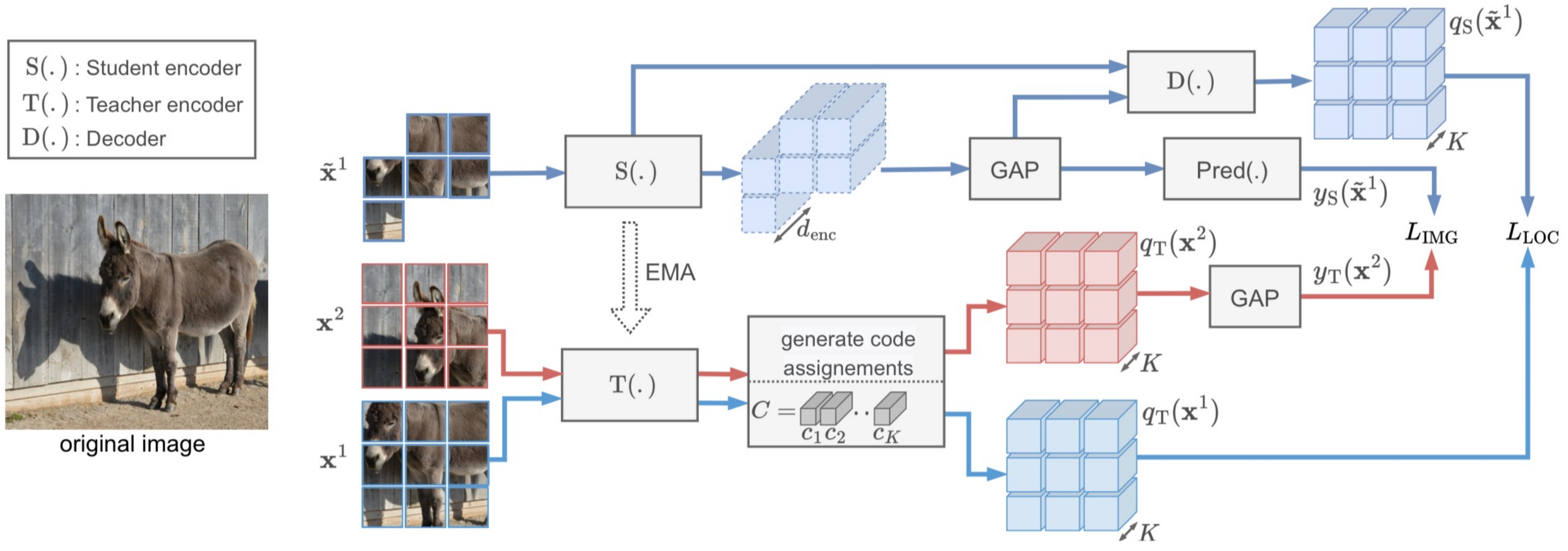


Generative Pretraining from Pixels



Some final remarks

MOCA ☕: Self-supervised Representation Learning by Predicting Masked Online Codebook Assignments

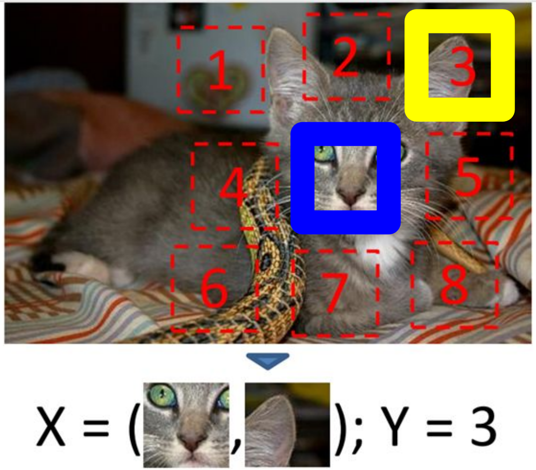


Shortcut prevention

[“Unsupervised visual representation learning by context prediction”, Doersh et al. 15]
[“Unsupervised learning of visual representations by solving jigsaw puzzles”, Noroozi et al. 17]

Exploiting local content

Recall: Context prediction



Edge continuity and shared boundary patterns

- Leave a gap between patches [Context prediction, Jigsaw]
- Jitter patch locations [Context prediction, Jigsaw]

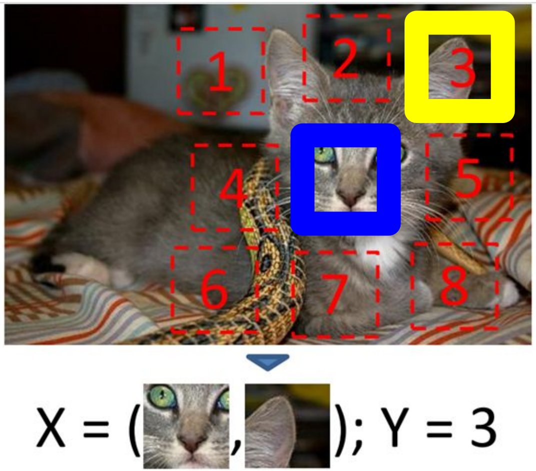
Similar low level statistics

- Normalize by mean and std of each patch independently [Jigsaw]

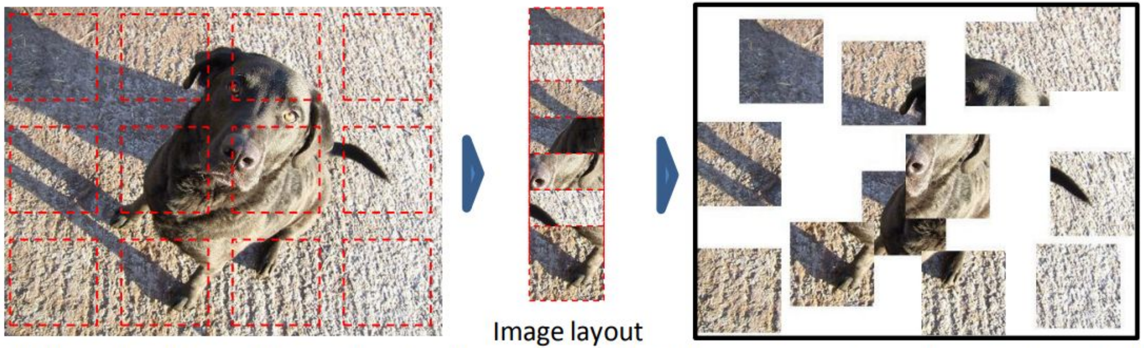
["Unsupervised visual representation learning by context prediction", Doersh et al. 15]
["Unsupervised learning of visual representations by solving jigsaw puzzles", Noroozi et al. 17]

Exploiting the capturing process

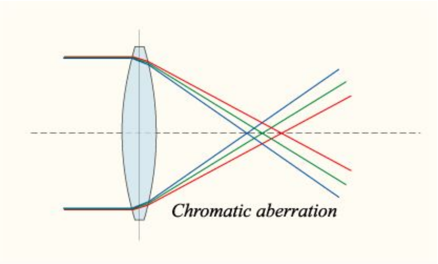
Recall: Context prediction



Networks can learn to predict the absolute patch position!



Initial layout, with sampled patches in red — Image layout is discarded — We can recover image layout automatically



source: [Wikipedia]

Prevent by keeping only 1 channel

- Increases the train-eval gap

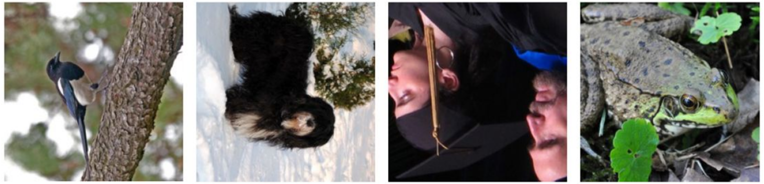
Alternatives

- Spatially jitter the channels [Jigsaw]

["Unsupervised representation learning by predicting image rotations", Gidaris et al. 18]

Exploiting low-level artefacts: Images

Recall: Rotation prediction



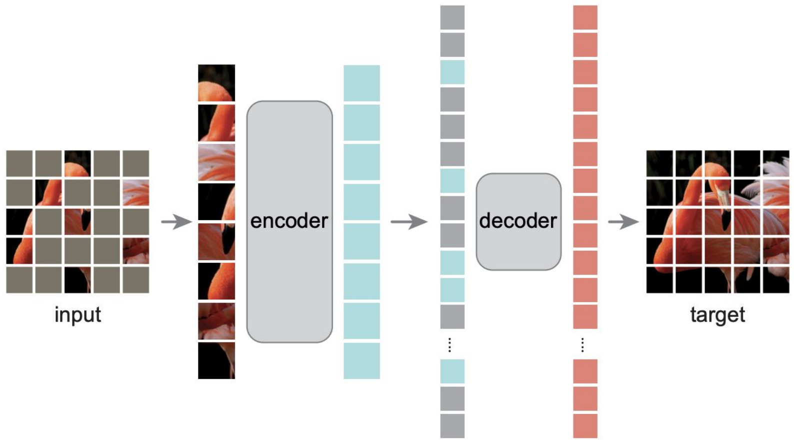
90° rotation 270° rotation 180° rotation 0° rotation

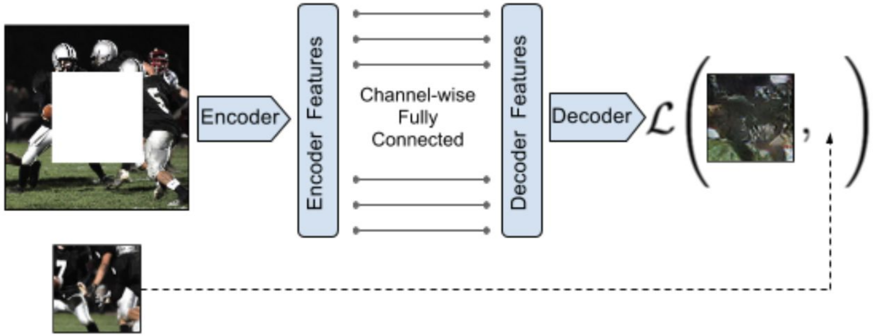
For more complicated transformations (more rotation angles, scales)

- Network can detect low-level artefacts of the transformations
- Forced to do only 4 rotations as they are implemented purely with flip and transpose artefact-less operations

Implementation choices matter

Autoencoders show the importance of architecture.





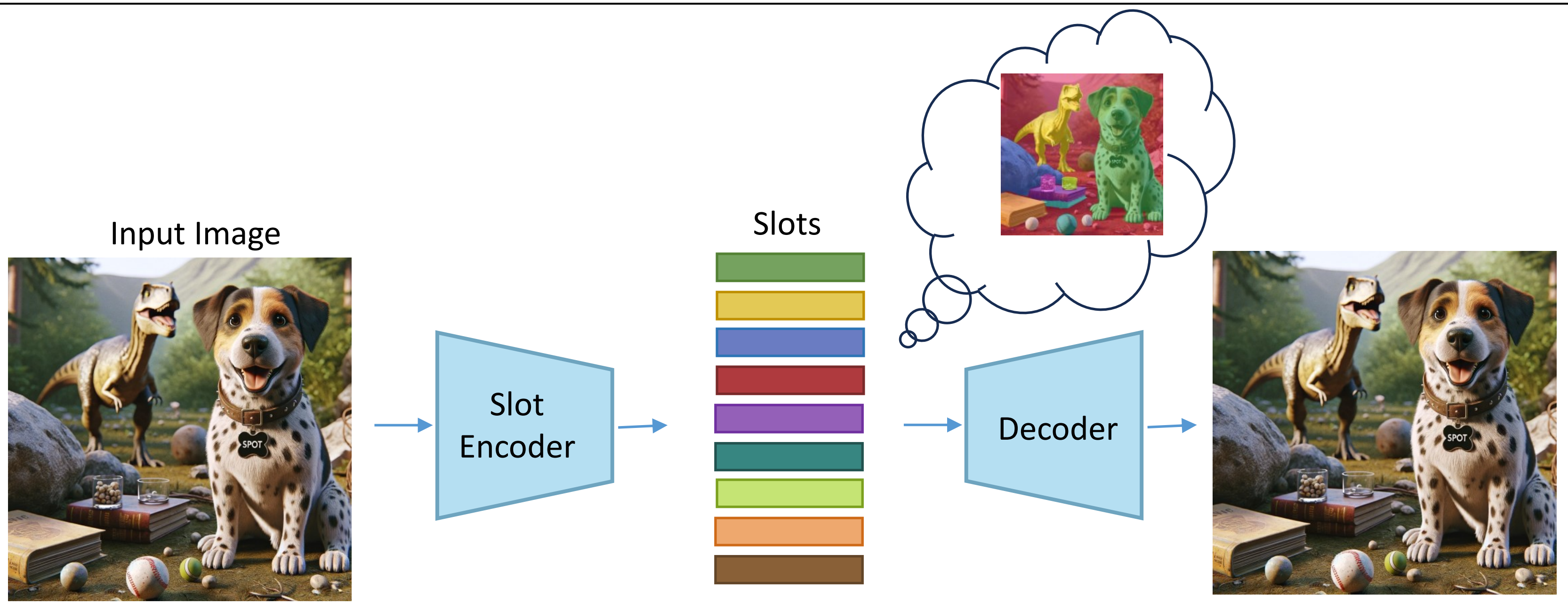
Masked Autoencoders Are Scalable Vision Learners
He et al. CVPR 2022

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

Other applications of self-supervision

Unsupervised Object-Centric Learning

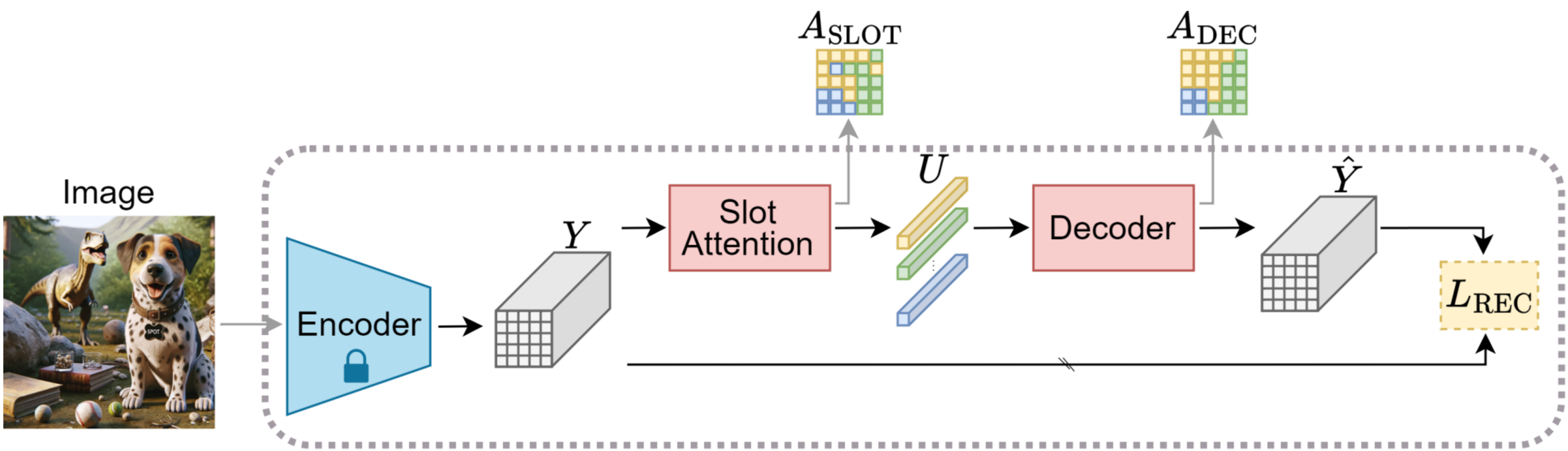
Goal: Unsupervised decomposition of images into set-based representations where distinct vectors represent different objects



- Learning without supervision such as segmentation masks or text
- Provide representations and segmentations of objects

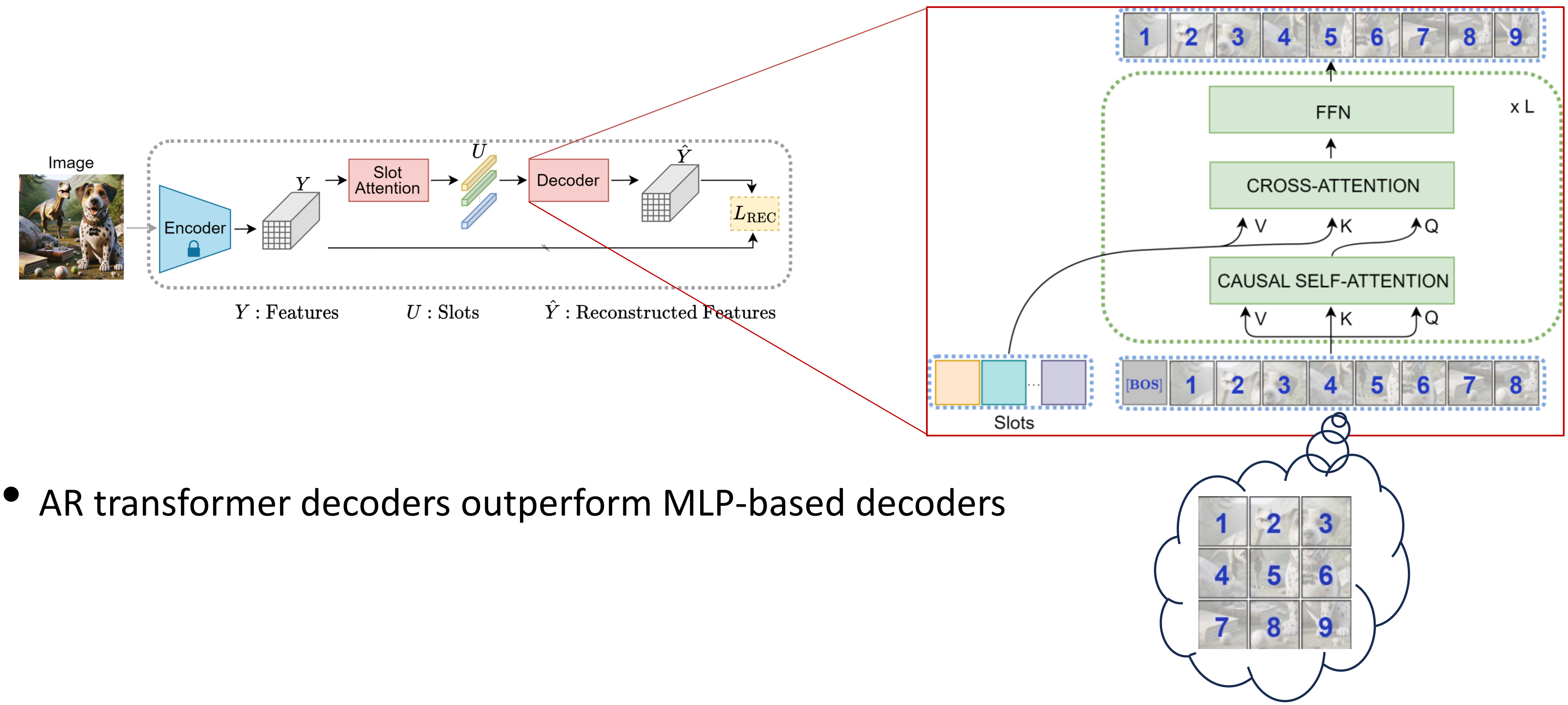
Slot-based Autoencoders

- Image Encoder + Slot-Attention: generate a set of slot vectors, each intended to represent an individual object within an image
- Decoder: reconstructs the target signal from the slots
- Segmentations come from the cross-attention layers



Y : Features $\quad U$: Slots $\quad \hat{Y}$: Reconstructed Features

Autoregressive (AR) Transformers

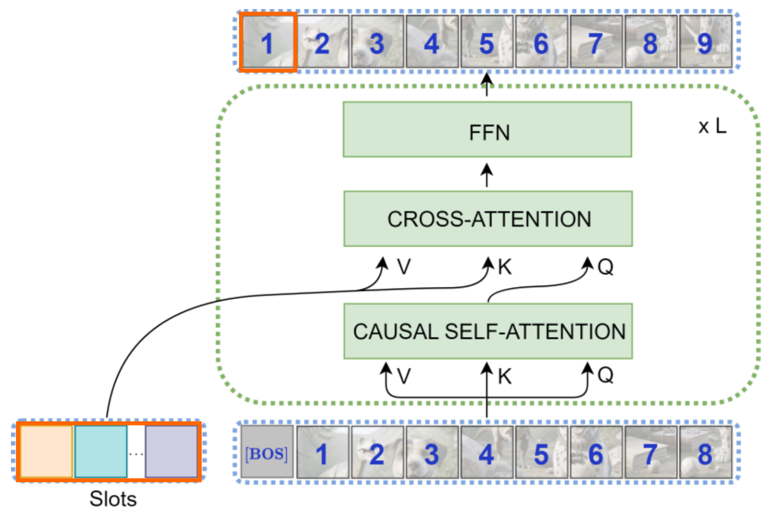


- AR transformer decoders outperform MLP-based decoders

Autoregressive (AR) Transformers

- Next token prediction

Standard order

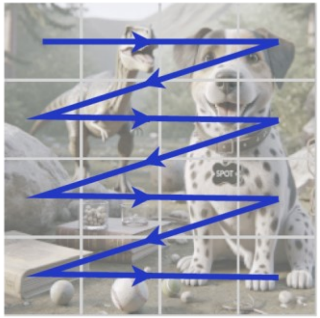


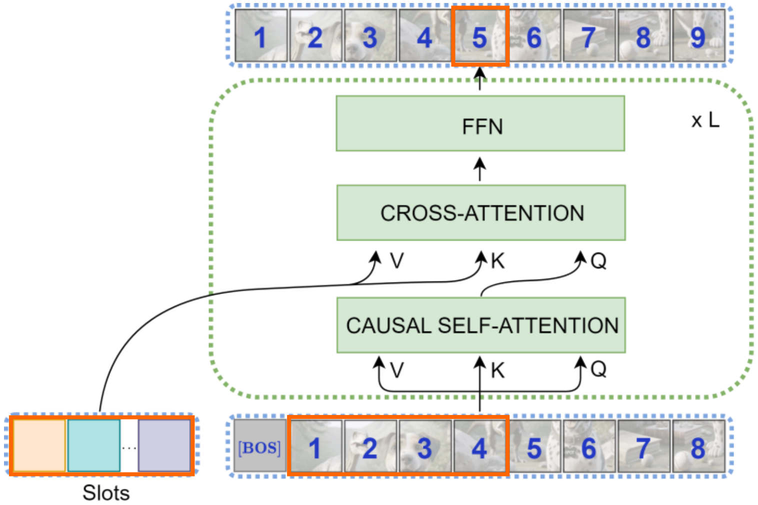
- AR transformer decoders outperform MLP-based decoders

Autoregressive (AR) Transformers

- Next token prediction

Standard order



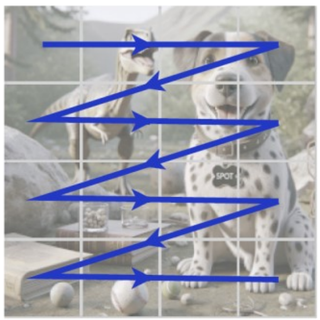


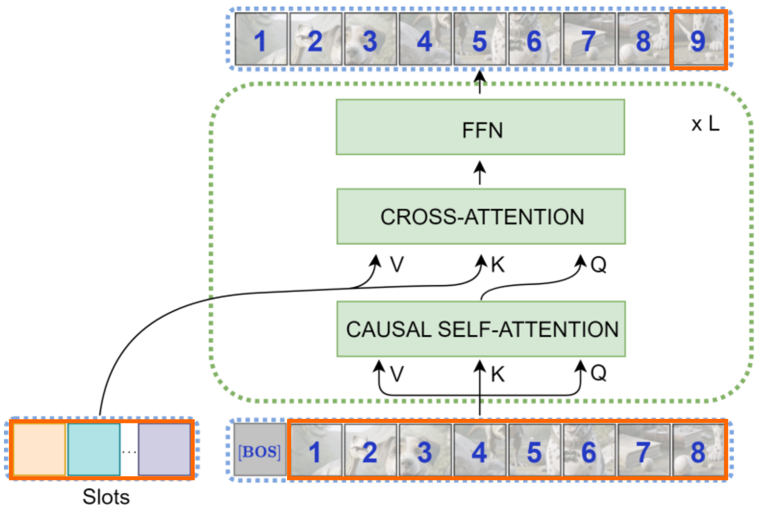
- AR transformer decoders outperform MLP-based decoders

Autoregressive (AR) Transformers

- Next token prediction

Standard order

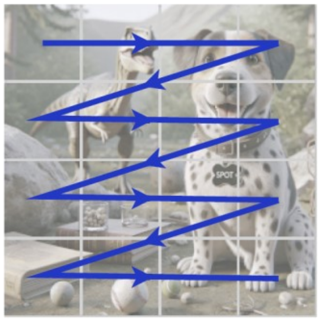


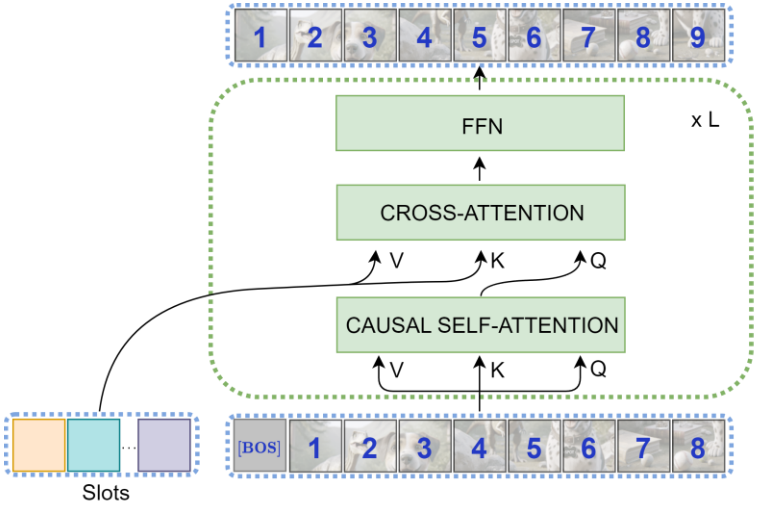


- AR transformer decoders outperform MLP-based decoders

Autoregressive (AR) Transformers

Standard order

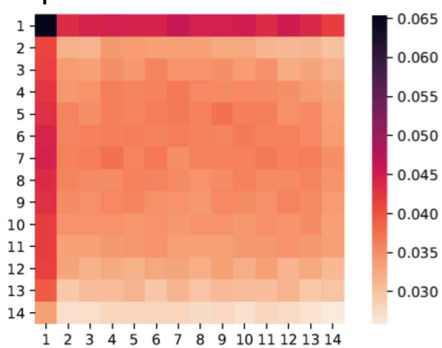


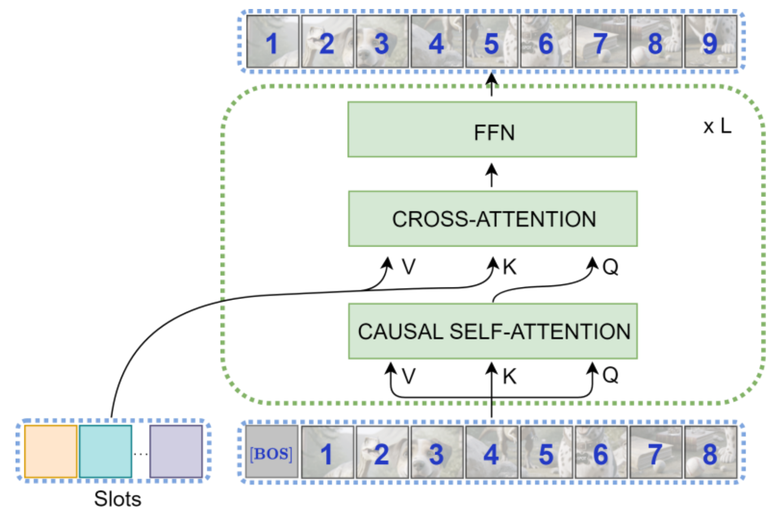


- AR transformer decoders outperform MLP-based decoders, but they have overfitting issues
  - Later tokens rely too much on past tokens → ignoring slot vectors

Autoregressive (AR) Transformers

Gradient norms for each patch w.r.t. the slots



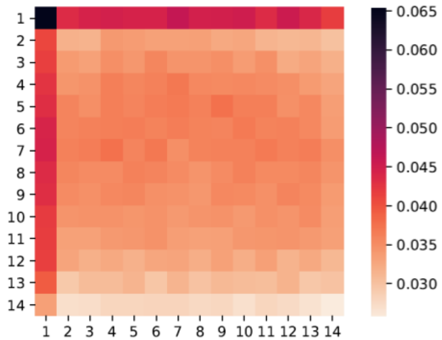


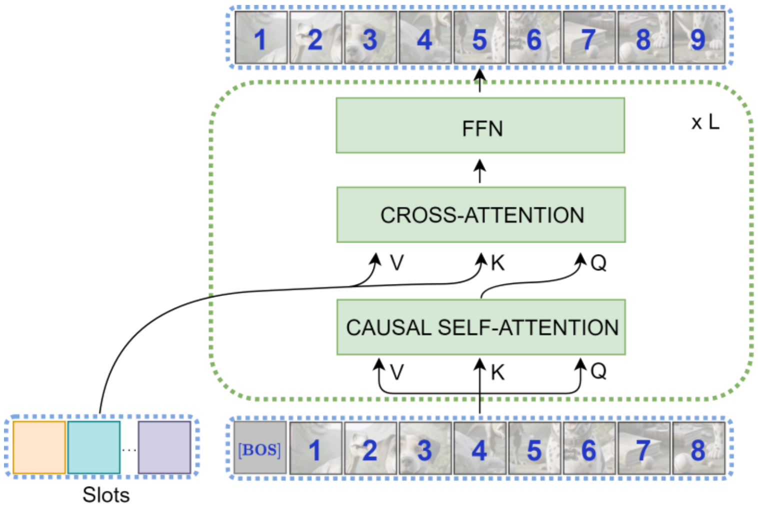
- AR transformer decoders outperform MLP-based decoders, but they have overfitting issues
  - Later tokens rely too much on past tokens → ignoring slot vectors

Left to right
Top to bottom

Sequence Permutations on Autoregressive Transformers

Gradient norms for each patch w.r.t. the slots

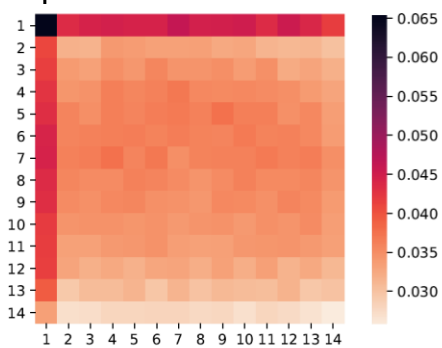


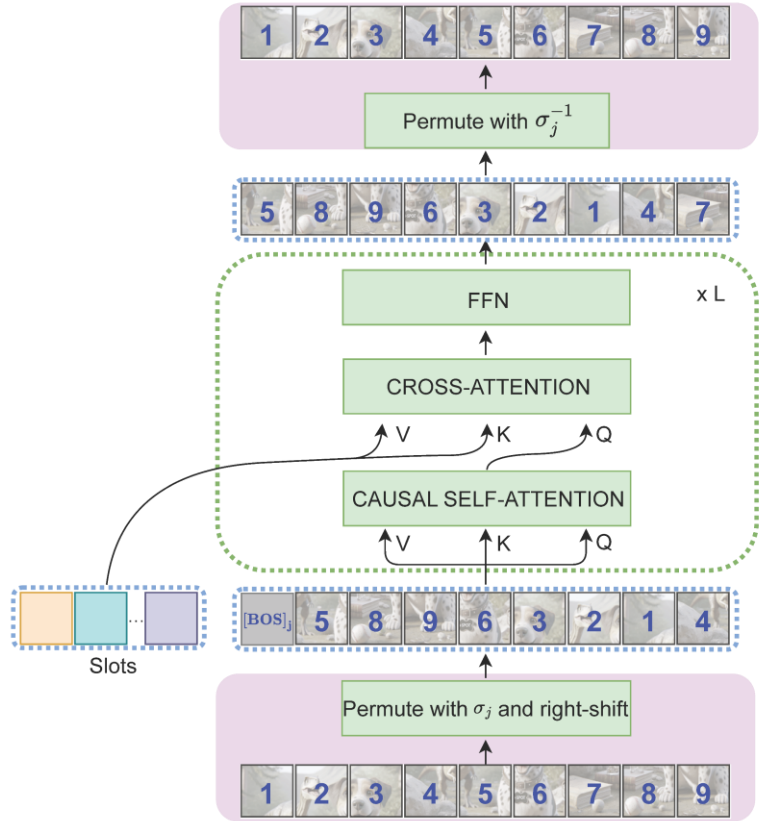


- To fix this, we introduce sequence permutations, altering the AR transformer's prediction order:
  - Permutations can move later tokens to initial positions → force them to use slot vectors

Sequence Permutations on Autoregressive Transformers

Gradient norms for each patch w.r.t. the slots



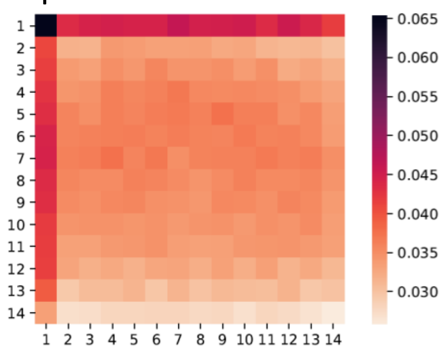


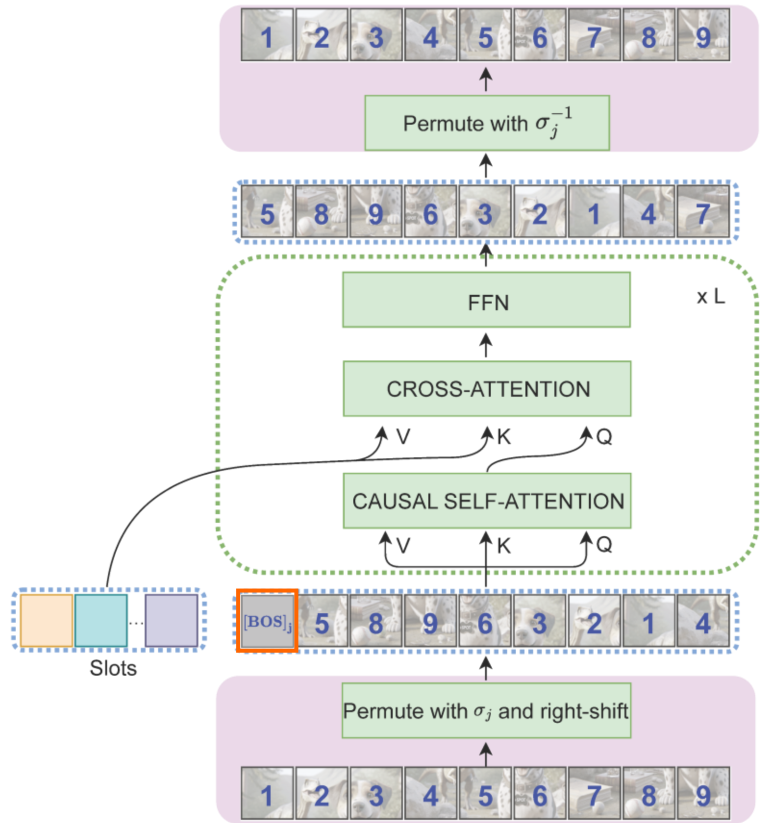
- To fix this, we introduce sequence permutations, altering the AR transformer's prediction order:
  - Permutations can move later tokens to initial positions → force them to use slot vectors

Left to right Top to bottom | Top to bottom Left to right | Top to bottom Right to left | Right to left Top to bottom | Bottom to top Right to left | Right to left Bottom to top | Bottom to top Left to right | Left to right Bottom to top | Spiral

Sequence Permutations on Autoregressive Transformers

Gradient norms for each patch w.r.t. the slots



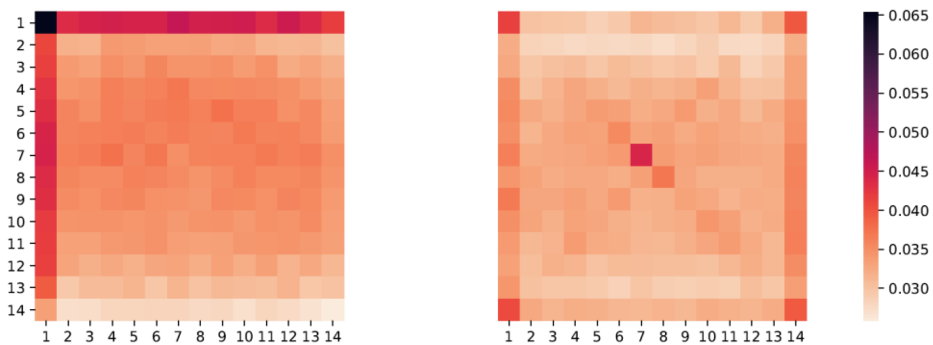


- To fix this, we introduce sequence permutations, altering the AR transformer's prediction order:
  - Permutations can move later tokens to initial positions → force them to use slot vectors

Left to right Top to bottom | Top to bottom Left to right | Top to bottom Right to left | Right to left Top to bottom | Bottom to top Right to left | Right to left Bottom to top | Bottom to top Left to right | Left to right Bottom to top | Spiral

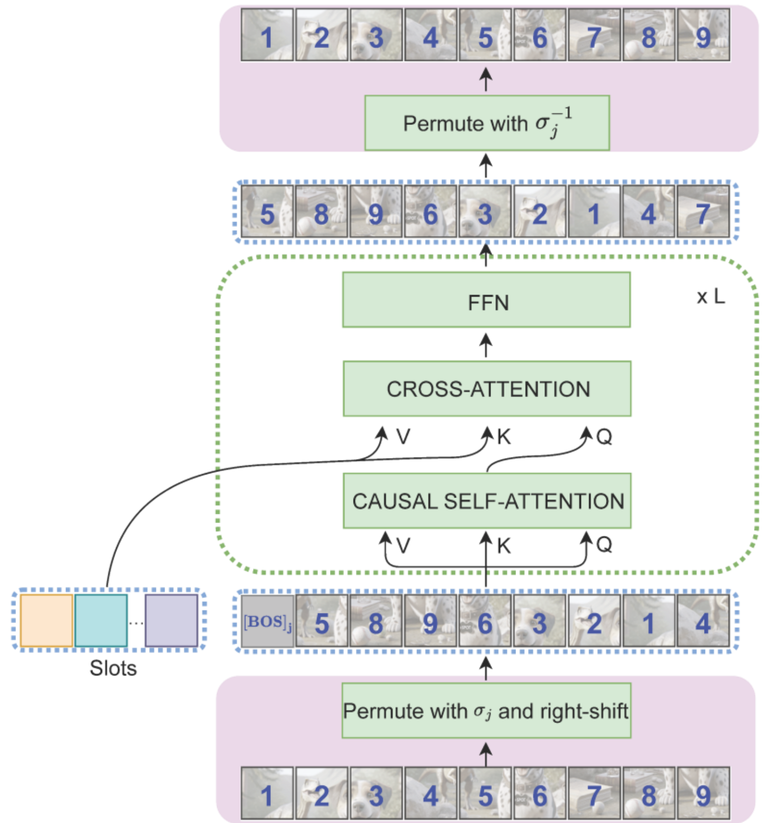
Sequence Permutations on Autoregressive Transformers

Gradient norms for each patch w.r.t. the slots



Decoder w/o permutations grads. of default perm.

Decoder with permutations grads. of random perm.



- To fix this, we introduce sequence permutations, altering the AR transformer's prediction order:
  - Permutations can move later tokens to initial positions → force them to use slot vectors

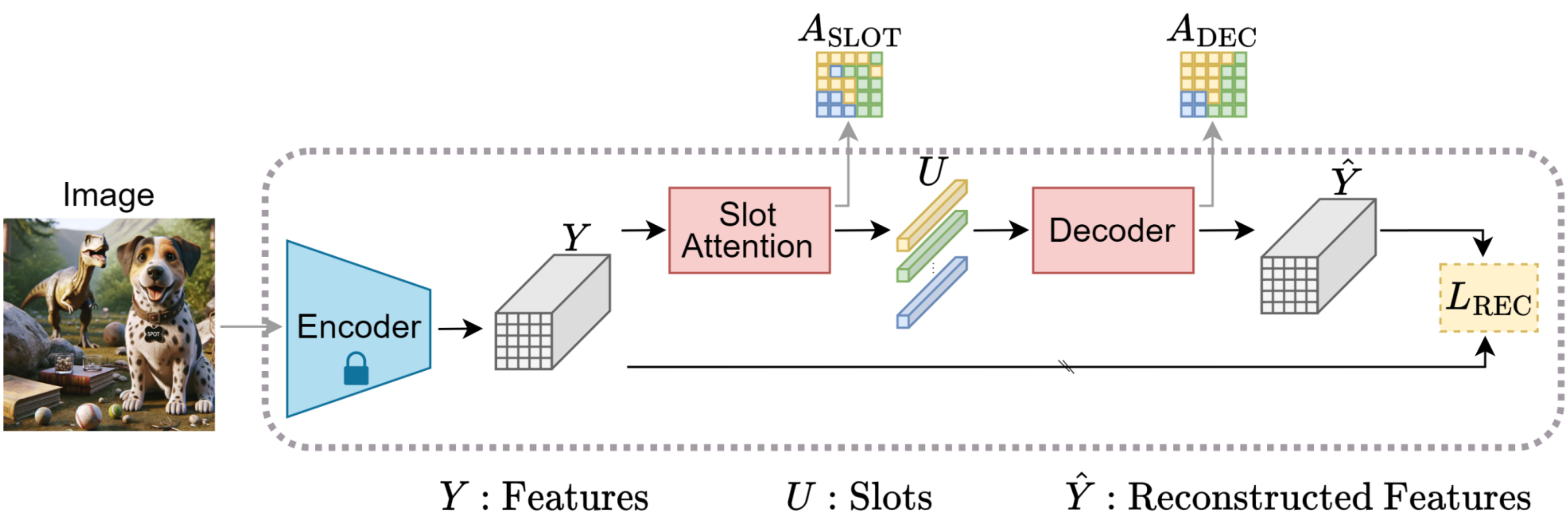
Left to right Top to bottom | Top to bottom Left to right | Top to bottom Right to left | Right to left Top to bottom | Bottom to top Right to left | Right to left Bottom to top | Bottom to top Left to right | Left to right Bottom to top | Spiral

Encoder & Decoder Slot-based Masks

- Slot-based attention masks can be generated by both the encoder (A_{SLOT}) and the decoder (A_{DEC})
- Decoder's masks demonstrate superior object segmentation

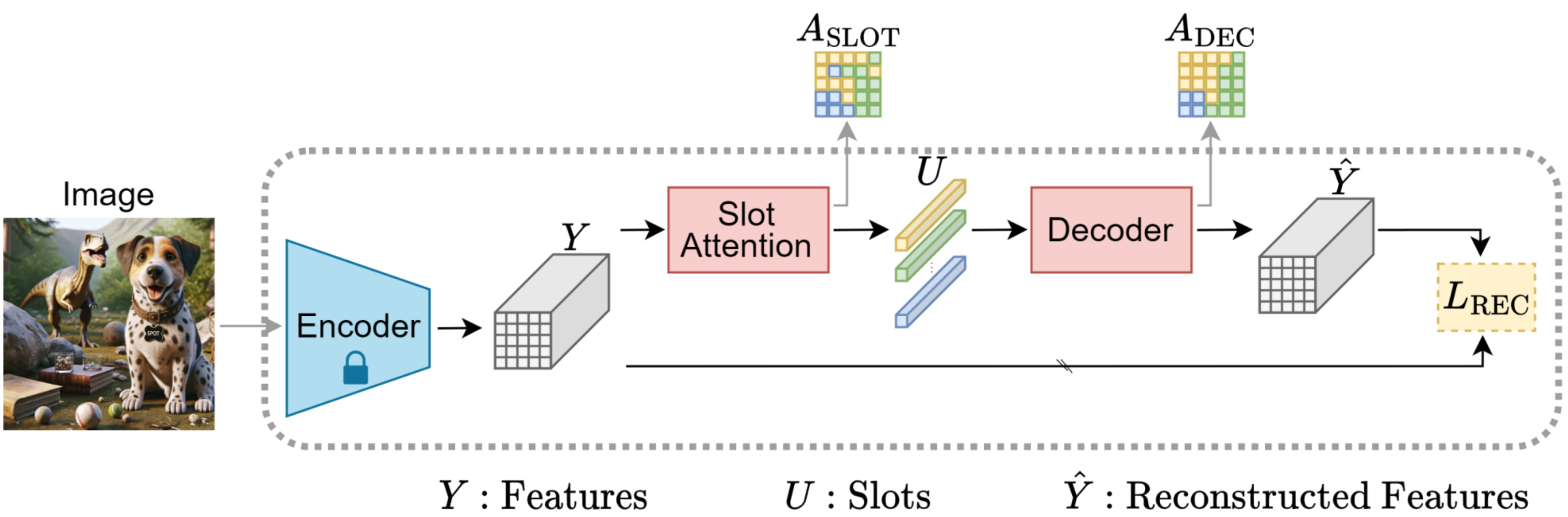
Mean Best Overlap with instance segmentation masks (MBO[i]) on COCO

	A_{SLOT}		A_{DEC}
MBO^i	30.0	<	32.0



Two-Stage Training Approach via Self-Training

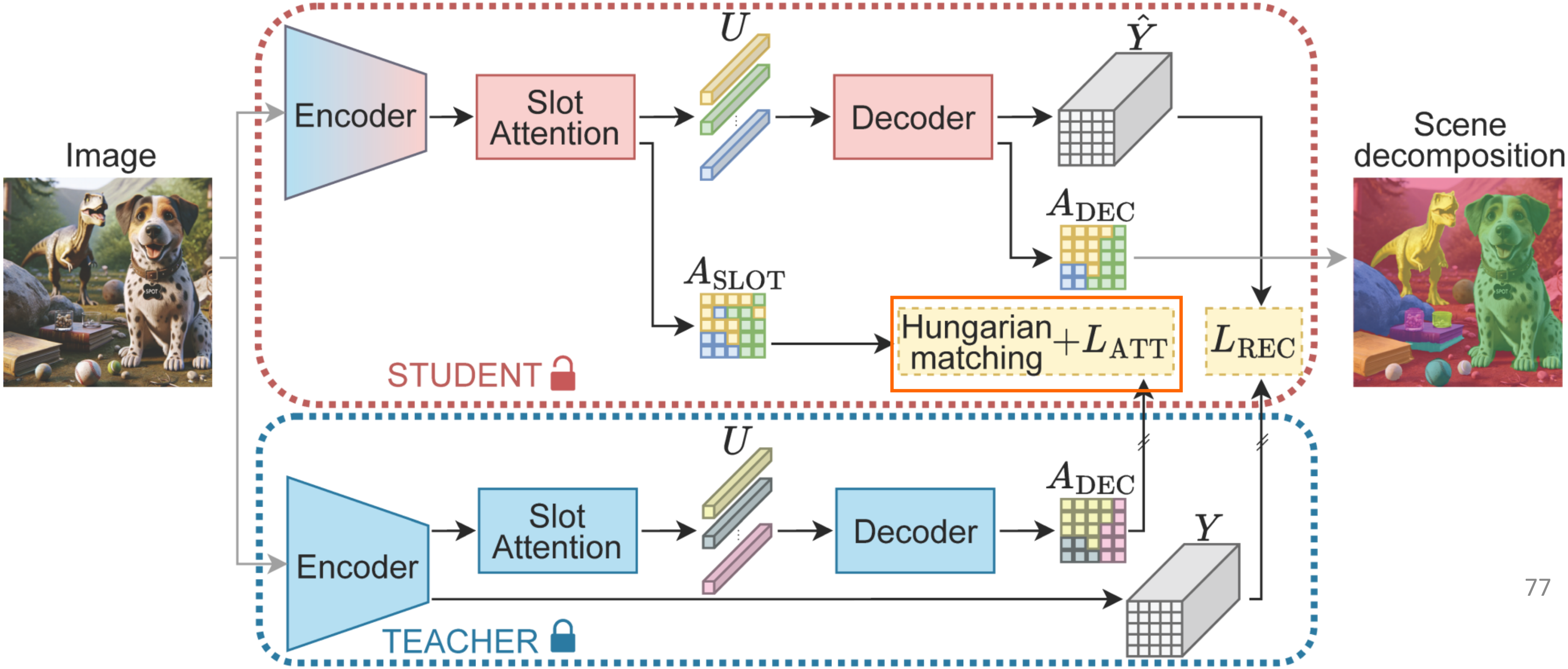
Stage-1: Train SPOT teacher using only the reconstruction loss L_{REC}



Two-Stage Training Approach via Self-Training

Stage-2: Train SPOT (student) using an additional self-training loss L_{ATT}

- The L_{ATT} loss distills slot-based attention masks from teacher's decoder to student's encoder
- This enhances the student's slot-attention grouping → improved slot representations



Experimental Results

Evaluating object-centric learning methods on object discovery:

SPOT achieves state-of-the-art results

Method	COCO		PASCAL		MOVi-C		MOVi-E	
	mBO^i	mBO^c	mBO^i	mBO^c	mBO^i	mIoU	mBO^i	mIoU
SA	17.2	19.2	24.6	24.9	26.2$_{\pm 1.0}$	-	24.0$_{\pm 1.2}$	-
SLASH	-	-	-	-	-	27.7$_{\pm 5.9}$	-	-
SLATE	29.1	33.6	35.9	41.5	39.4$_{\pm 0.8}$	37.8$_{\pm 0.7}$	30.2$_{\pm 1.7}$	28.6$_{\pm 1.7}$
CAE	-	-	32.9$_{\pm 0.9}$	37.4$_{\pm 1.0}$	-	-	-	-
DINOSAUR	32.3$_{\pm 0.4}$	38.8$_{\pm 0.4}$	44.0$_{\pm 1.9}$	51.2$_{\pm 1.9}$	42.4	-	-	-
DINOSAUR-MLP	27.7$_{\pm 0.2}$	30.9$_{\pm 0.2}$	39.5$_{\pm 0.1}$	40.9$_{\pm 0.1}$	39.1$_{\pm 0.2}$	-	35.5$_{\pm 0.2}$	-
Rotating Features	-	-	40.7$_{\pm 0.1}$	46.0$_{\pm 0.1}$	-	-	-	-
SlotDiffusion	31.0	35.0	**50.4**	<u>55.3</u>	-	-	30.2	30.2
(Stable-)LSD	30.4	-	-	-	45.6$_{\pm 0.8}$	44.2$_{\pm 0.9}$	39.0$_{\pm 0.5}$	37.6$_{\pm 0.5}$
SPOT (ours)	**35.0**$_{\pm 0.1}$	**44.7**$_{\pm 0.3}$	<u>48.3$_{\pm 0.4}$</u>	**55.6**$_{\pm 0.4}$	**47.3**$_{\pm 1.2}$	**46.7**$_{\pm 1.3}$	**40.1**$_{\pm 1.2}$	**39.3**$_{\pm 1.2}$

✓ Outperforms prior state-of-the-art DINOSAUR by +2.7% mBO^i & +5.9% mBO^c in real-world object-centric learning

✓ Excels also in simpler or synthetic datasets adopted by object-centric learning community

Experimental Results

More qualitative results on COCO

SPOT is applicable to other encoders

Encoder	Method	mBO^i	FG-ARI
DINO	DINOSAUR	31.6$_{\pm 0.7}$	34.1$_{\pm 1.0}$
	SPOT	**35.0**$_{\pm 0.1}$	37.0$_{\pm 0.2}$
MoCo-v3	DINOSAUR	31.4$_{\pm 0.2}$	35.2$_{\pm 0.2}$
	SPOT	32.9$_{\pm 0.2}$	34.8$_{\pm 0.3}$
MAE	DINOSAUR	30.2$_{\pm 1.8}$	32.8$_{\pm 3.7}$
	SPOT	33.4$_{\pm 0.3}$	**37.7**$_{\pm 1.0}$

Self-training is effective even with an MLP decoder

Decoder	Self-Training	mBO^i	mIoU	FG-ARI
MLP	✗	26.7	25.6	38.7
	✓	**28.4**	**27.0**	**42.5**

Sequence permutation is superior to parallel decoding

Decoder	mBO^i	mIoU	FG-ARI
Transformer	32.0	30.0	32.3
Transformer w/ PA	27.8	26.5	35.3
Transformer w/ SP	**32.7**	**30.8**	**35.6**

Summarizing Insights

SPOT:

- ✓ Outperforms the other unsupervised slot-based object-centric learning methods in real-world images, achieving state-of-the-art results
- ✓ Autoregressive (AR) decoding in object-centric learning with sequence permutations is superior to default AR decoding, parallel masked decoding or simple MLP decoding
- ✓ Sequence permutation may benefit other computer vision tasks with autoregressive decoders

Source code: https://github.com/gkakogeorgiou/spot

THANK YOU