max planck institut
informatik

SIC Saarland Informatics Campus

High Level Computer Vision

Self-Supervised Learning — Part 1

@ June 14, 2023

Bernt Schiele

cms.sic.saarland/hlcvss23/

Max Planck Institute for Informatics & Saarland University,
Saarland Informatics Campus Saarbrücken

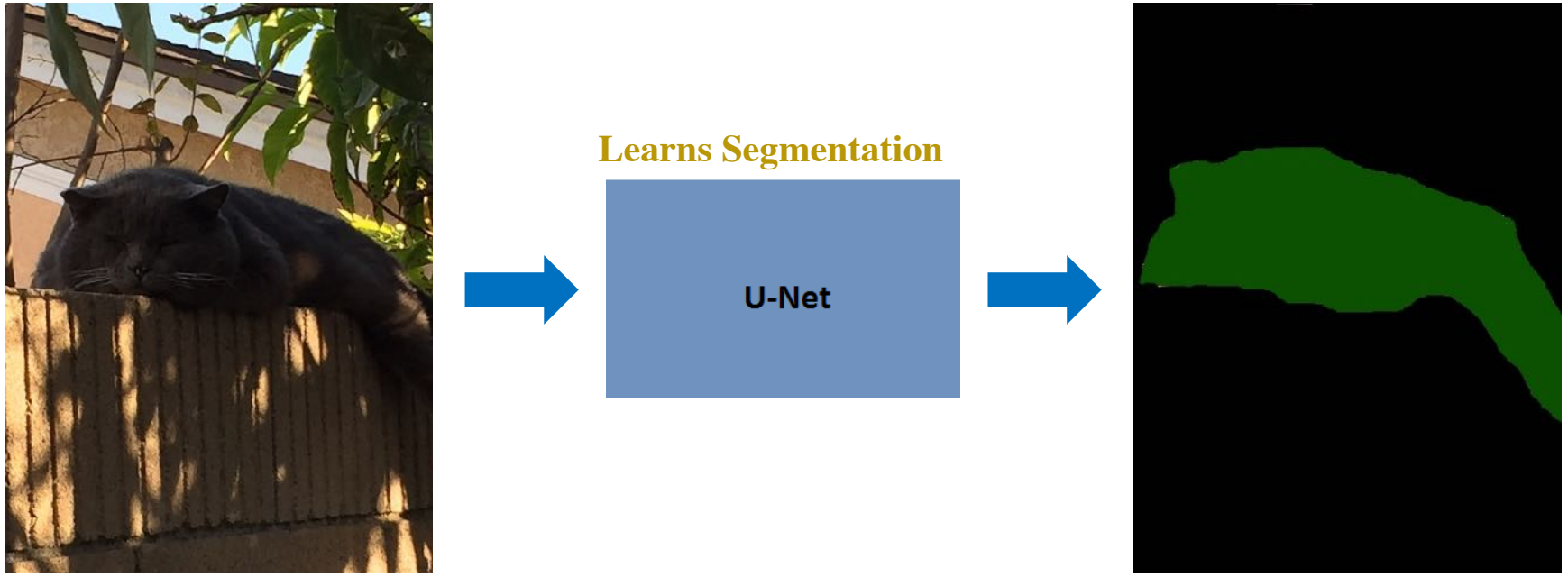
Overview of Today's Lecture

- U-Net Architecture for Semantic Segmentation

- Self-Supervised Learning - Part 1 (today):
  - Motivation of Self-Supervised Learning
    - setting of pretext tasks — how to evaluate
  - Pretext tasks from image transformations
    - rotation, inpainting, rearrangement, coloring
  - Contrastive representation learning
    - intuition and formulation
    - instance contrastive learning: SimCLR & MOCO
    - sequence contrastive learning: CPC

- Self-Supervised Learning - Part 2 (next time):
  - Teacher-Student "feature reconstruction"
    - motivation, setting
    - methods: BYOL, DINO

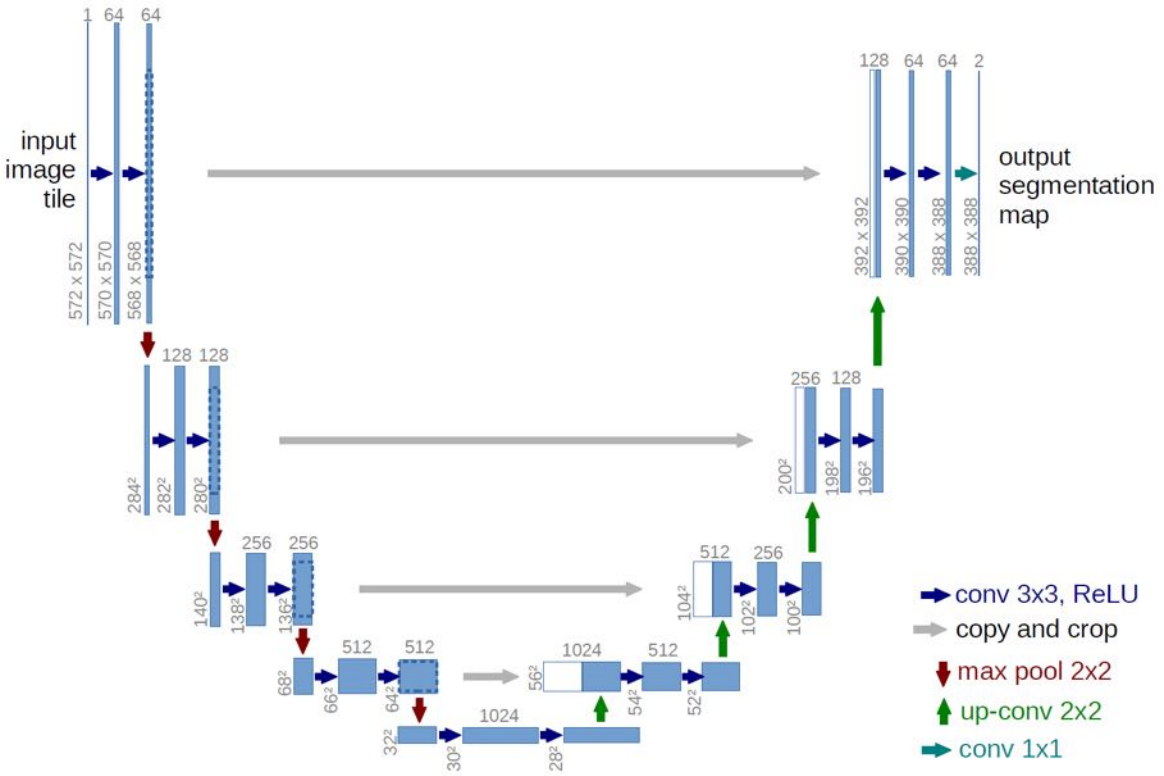
What does a U-Net do?



Input Image

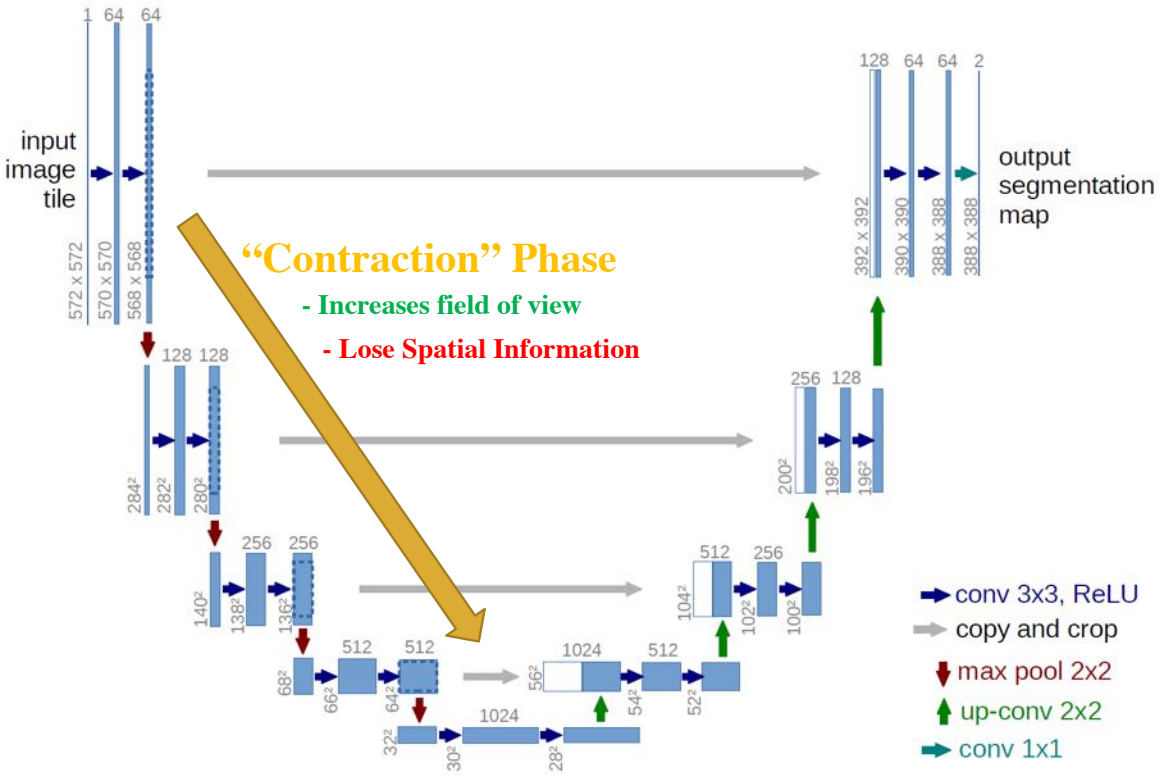
Output Segmentation Map

U-Net Architecture



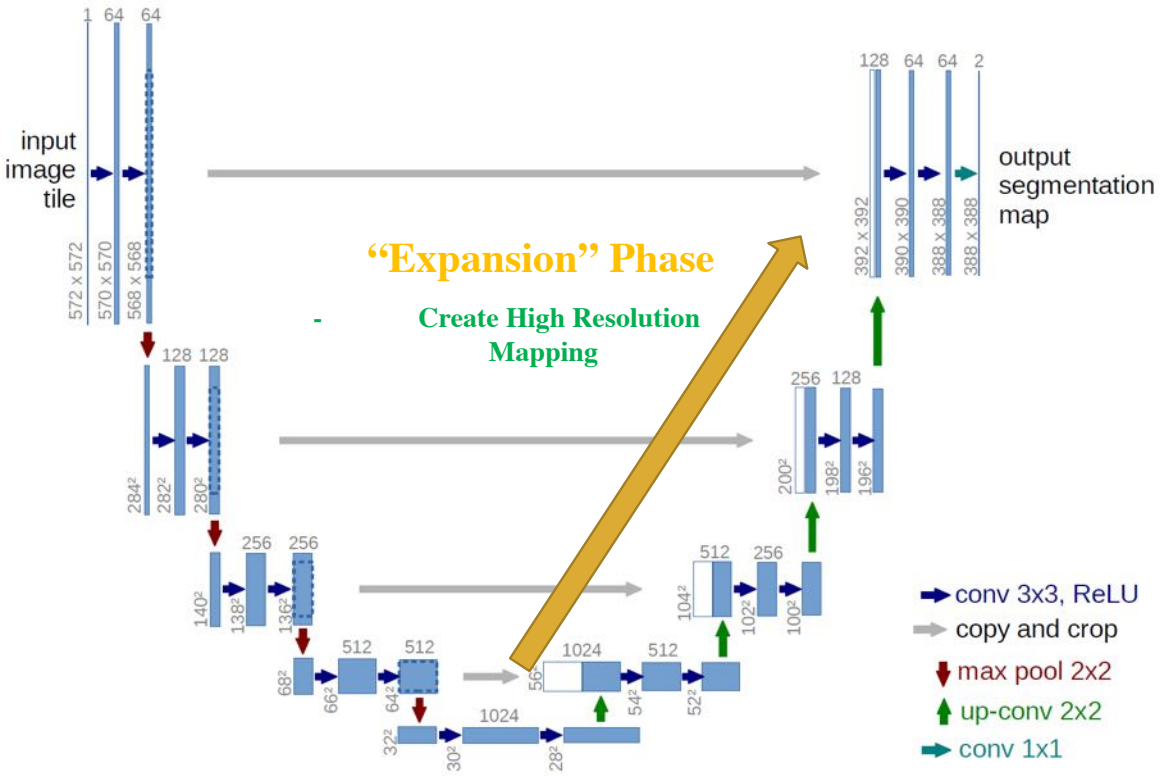
Ronneberger et al. (2015) U-net Architecture

U-Net Architecture



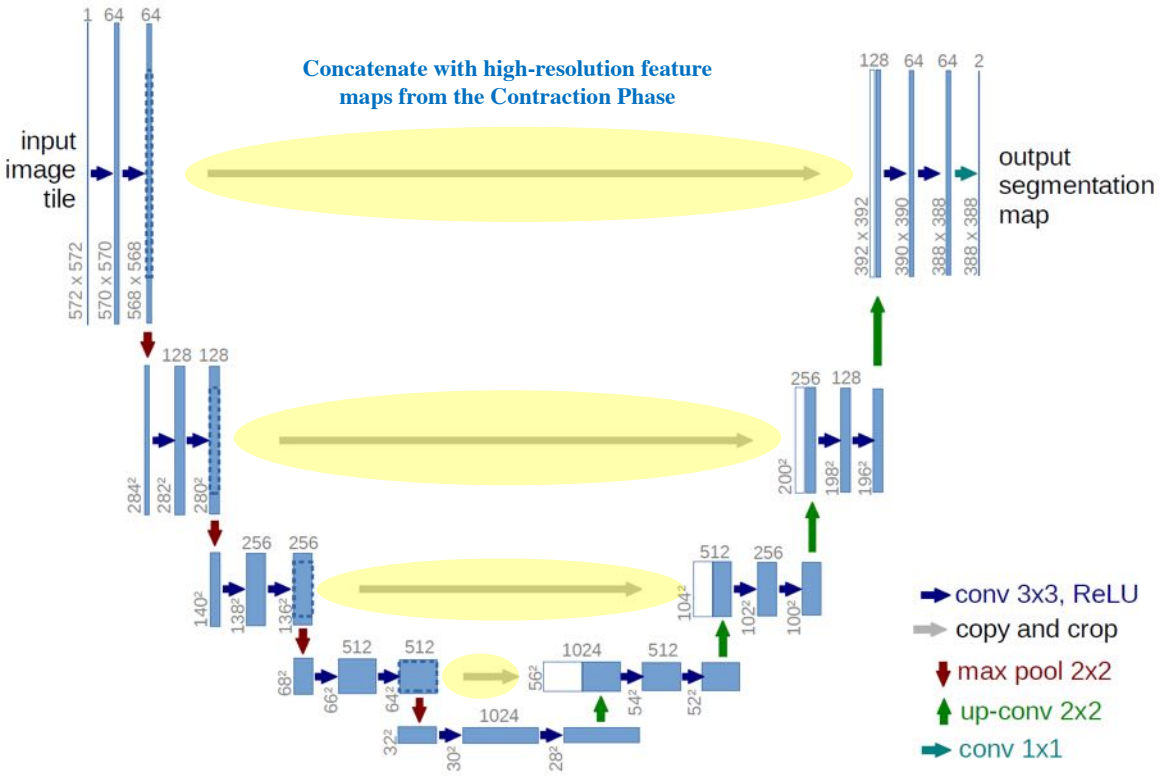
Ronneberger et al. (2015) U-net Architecture

U-Net Architecture



Ronneberger et al. (2015) U-net Architecture

U-Net Architecture

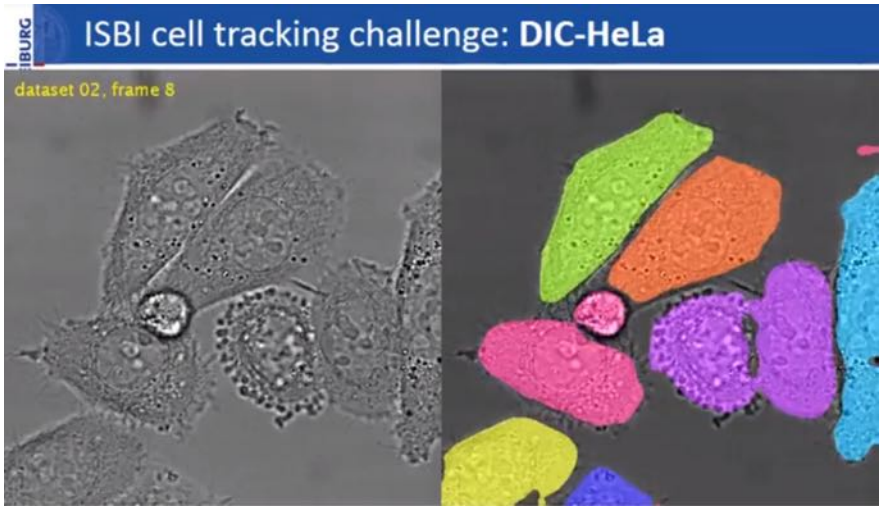


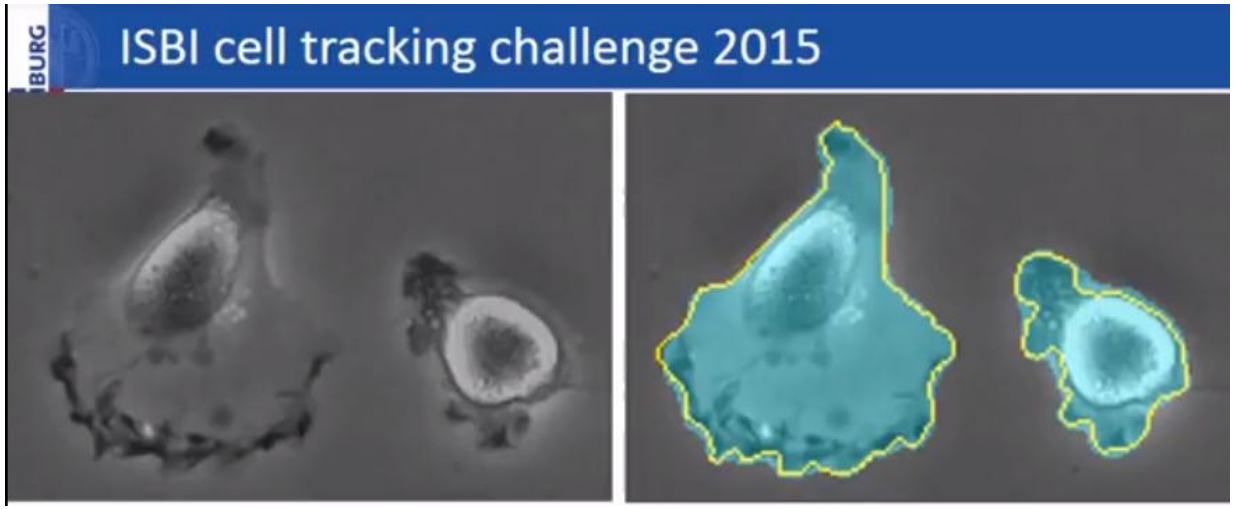
Ronneberger et al. (2015) U-net Architecture

U-Net Summary

- Contraction Phase
  - Reduce spatial dimension, but increases the “what.”
- Expansion Phase
  - Recovers object details and the dimensions, which is the “where.”
- Concatenating feature maps from the Contraction phase helps the Expansion phase with recovering the “where” information.

Author Results





Ronneberger et al. (2015) ISBI cell tracking challenge

Overview of Today's Lecture

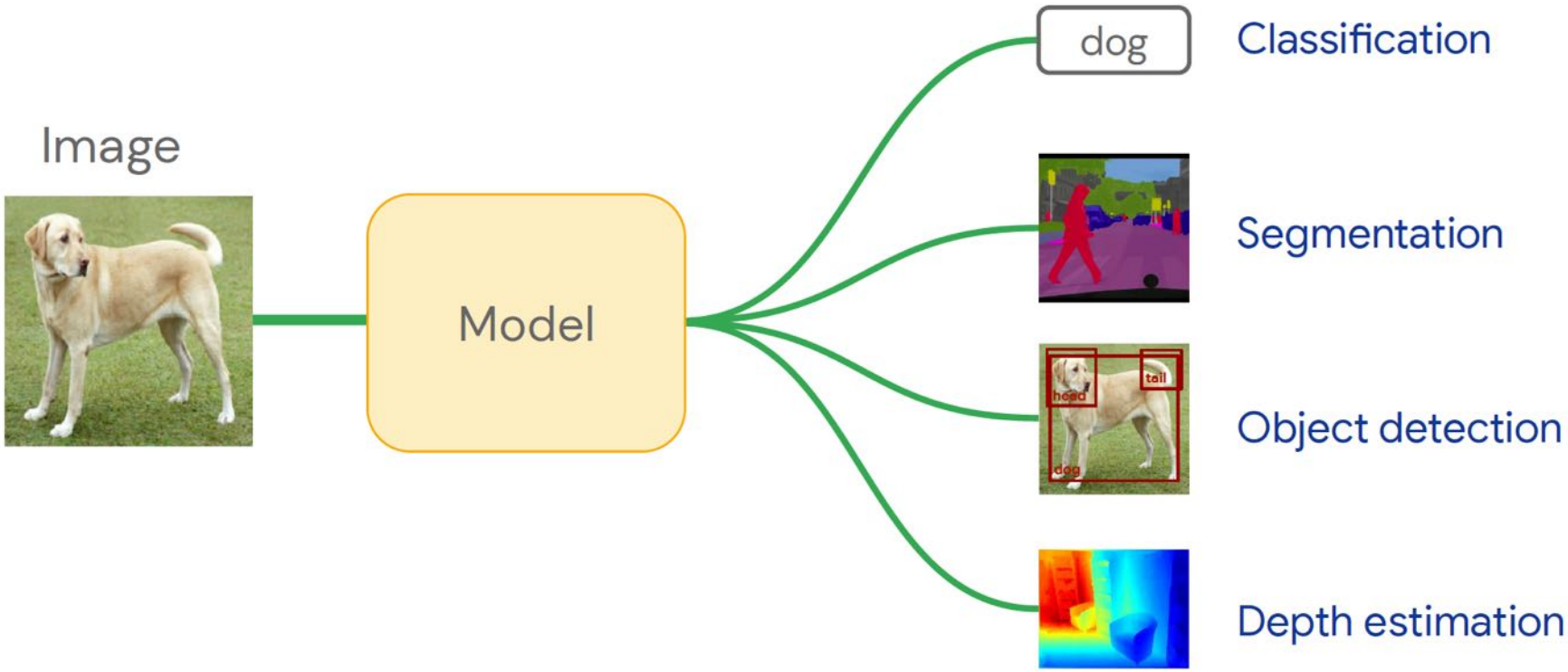
- U-Net Architecture for Semantic Segmentation

- Self-Supervised Learning - Part 1 (today):
  - Motivation of Self-Supervised Learning
    - setting of pretext tasks — how to evaluate
  - Pretext tasks from image transformations
    - rotation, inpainting, rearrangement, coloring
  - Contrastive representation learning
    - intuition and formulation
    - instance contrastive learning: SimCLR & MOCO
    - sequence contrastive learning: CPC

- Self-Supervised Learning - Part 2 (next time):
  - Teacher-Student "feature reconstruction"
    - motivation, setting
    - methods: BYOL, DINO

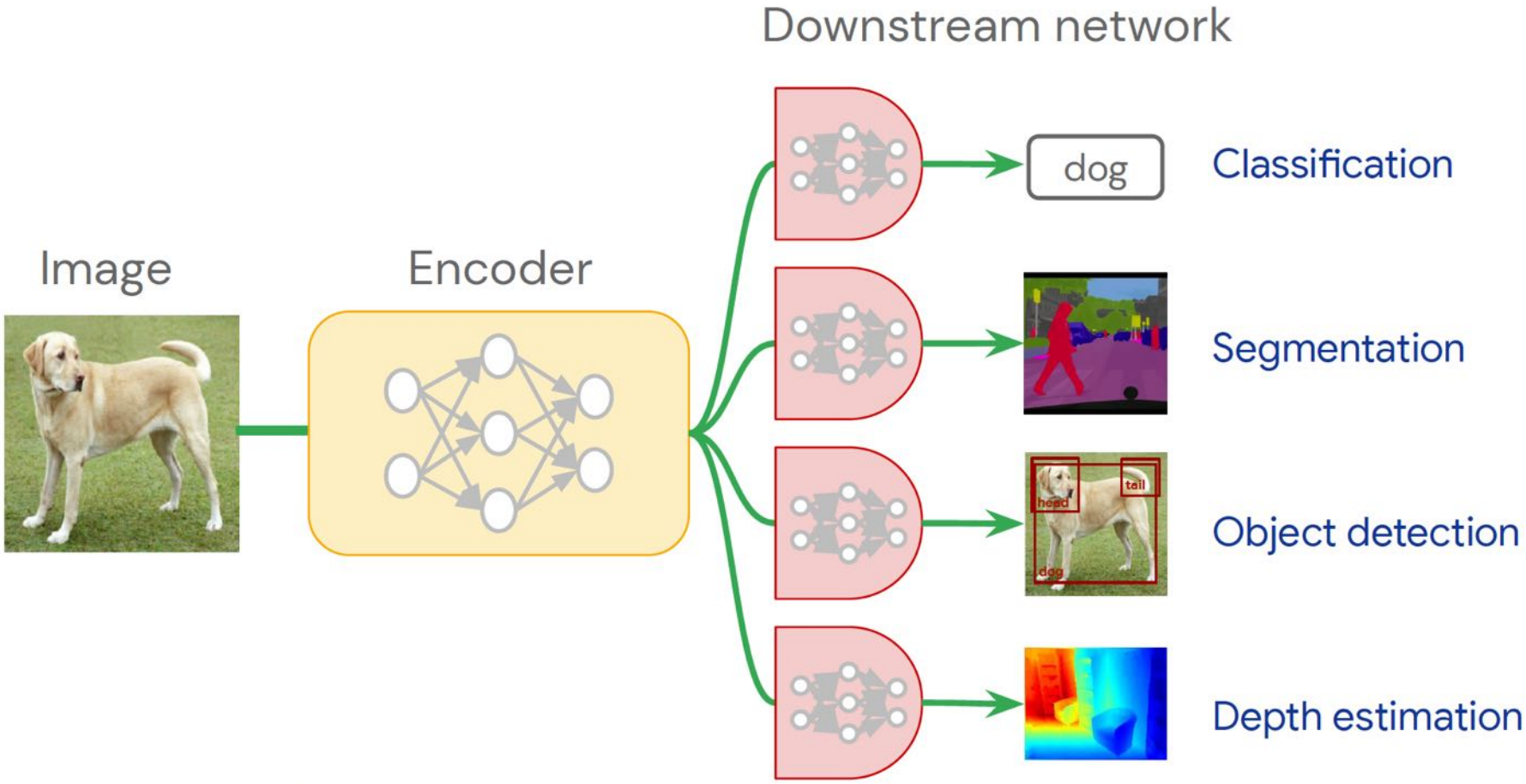
slide credit: Jean-Basien Grill, et al.

Goals of Computer Vision



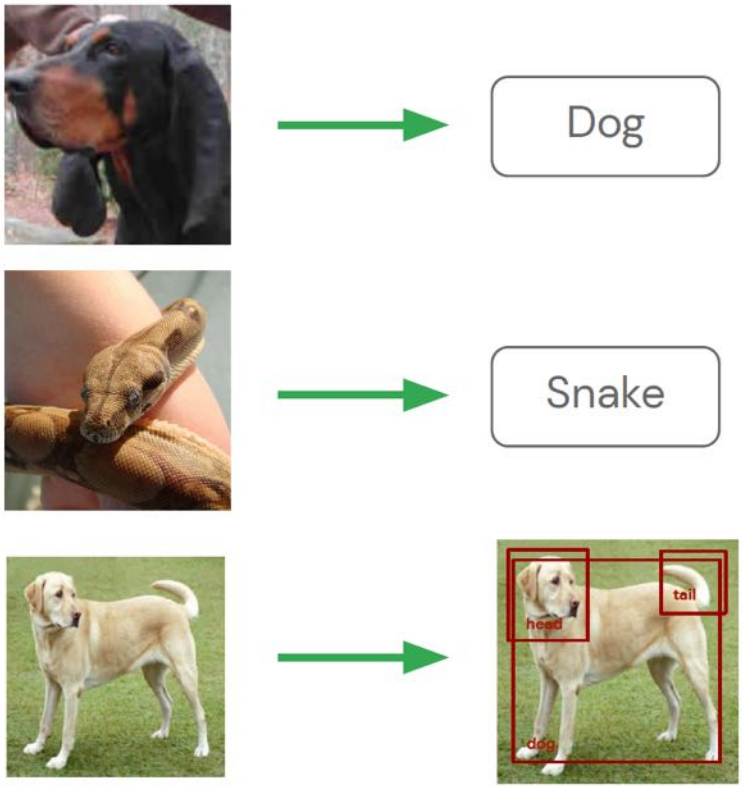
slide credit: Jean-Basien Grill, et al.

Can We Train an “Encoder” for Many Downstream Tasks?

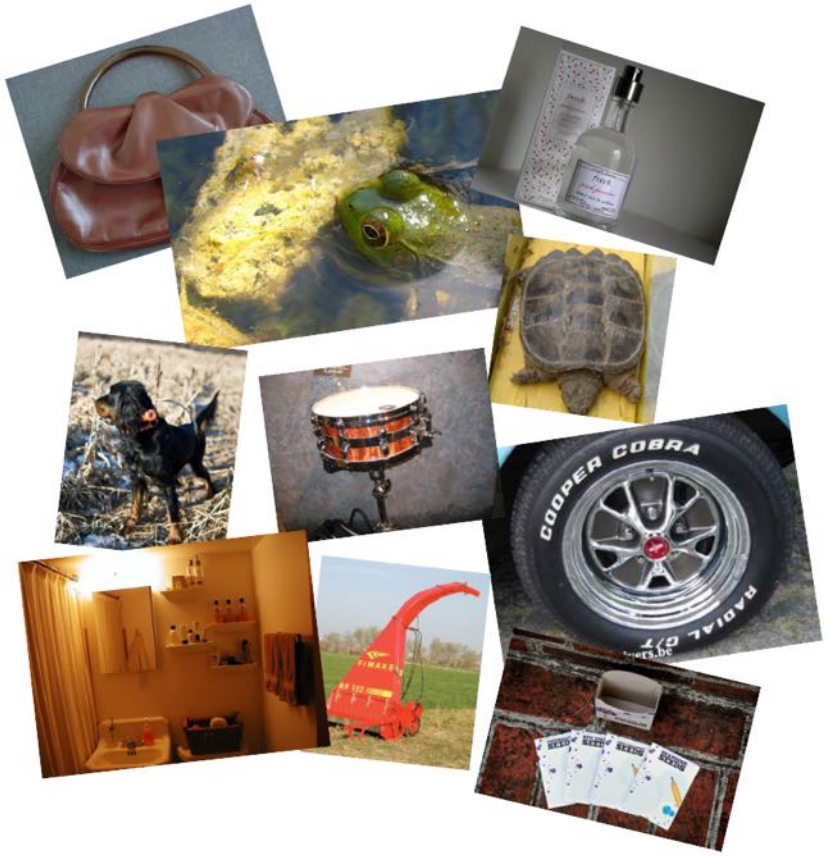


slide credit: Jean-Basien Grill, et al.

Labelled vs. Unlabelled data…



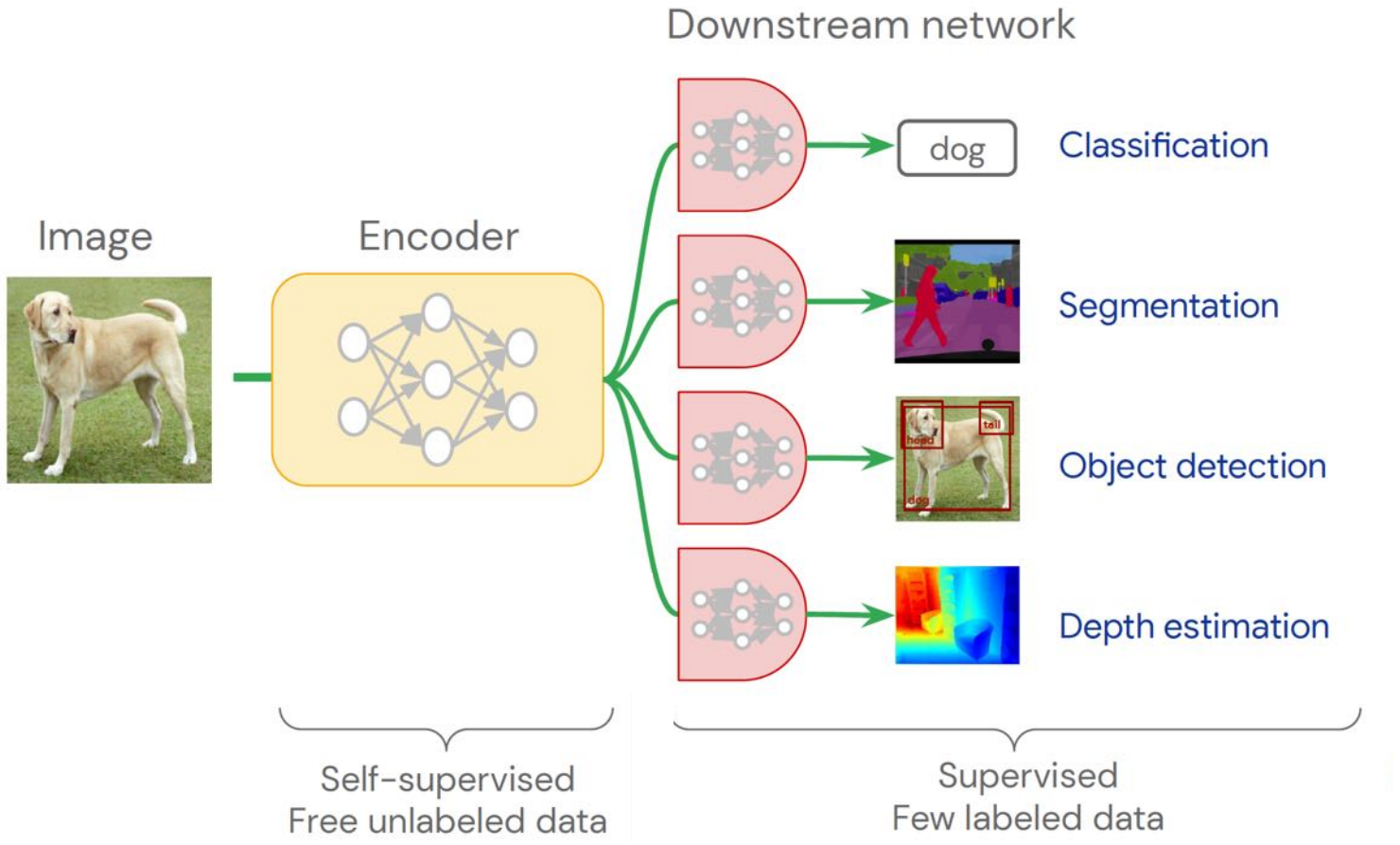
Labelled, but costly/few data



Unlabelled, **free** data!

slide credit: Jean-Basien Grill, et al.

Idea of Self-Supervised Learning

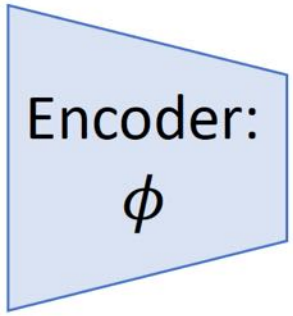


slide credit: Justin Johnson

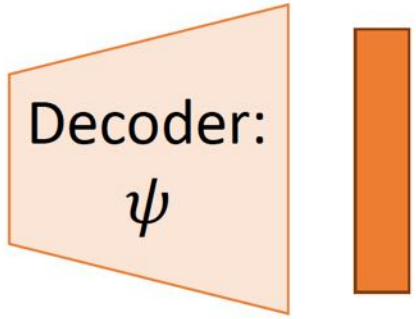
Self-Supervised Learning: Pretraining — then Transfer

Step 1: Pretrain a network on a pretext task that doesn't require supervision

Input Image: x



Features: $\phi(x)$

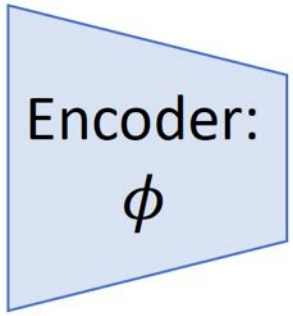


Prediction: $\hat{y}$

Loss: $L(\hat{y}, y)$

Step 2: Transfer encoder to downstream tasks via linear classifiers, KNN, finetuning

Input Image: x



Features: $\phi(x)$

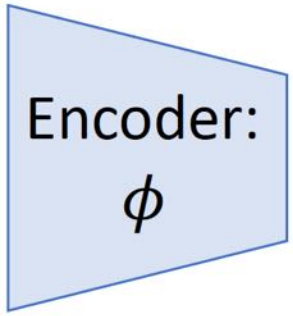
Downstream tasks: Image classification, object detection, semantic segmentation

slide credit: Justin Johnson

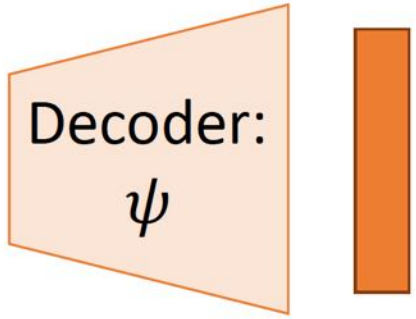
Self-Supervised Learning: Pretraining — then Transfer

Step 1: Pretrain a network on a pretext task that doesn't require supervision

Input Image: x



Features: $\phi(x)$

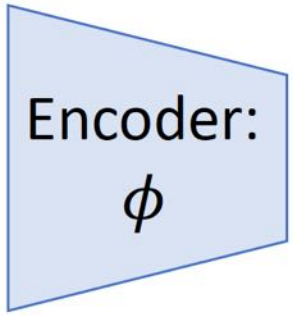


Prediction: $\hat{y}$

Loss: $L(\hat{y}, y)$

Step 2: Transfer encoder to downstream tasks via linear classifiers, KNN, finetuning

Input Image: x



Features: $\phi(x)$

Goal: Pretrain + Transfer does better than supervised pretraining, and better than directly training on downstream

Self-Supervised Learning

- Different Types of Methods

  - **Pretext tasks** from image transformations
    - rotation, inpainting, rearrangement, coloring

  - **Contrastive representation learning**
    - instance contrastive learning: SimCLR & MOCO
    - sequence contrastive learning: CPC

  - **Teacher-Student** “feature reconstruction”
    - methods: BYOL, DINO

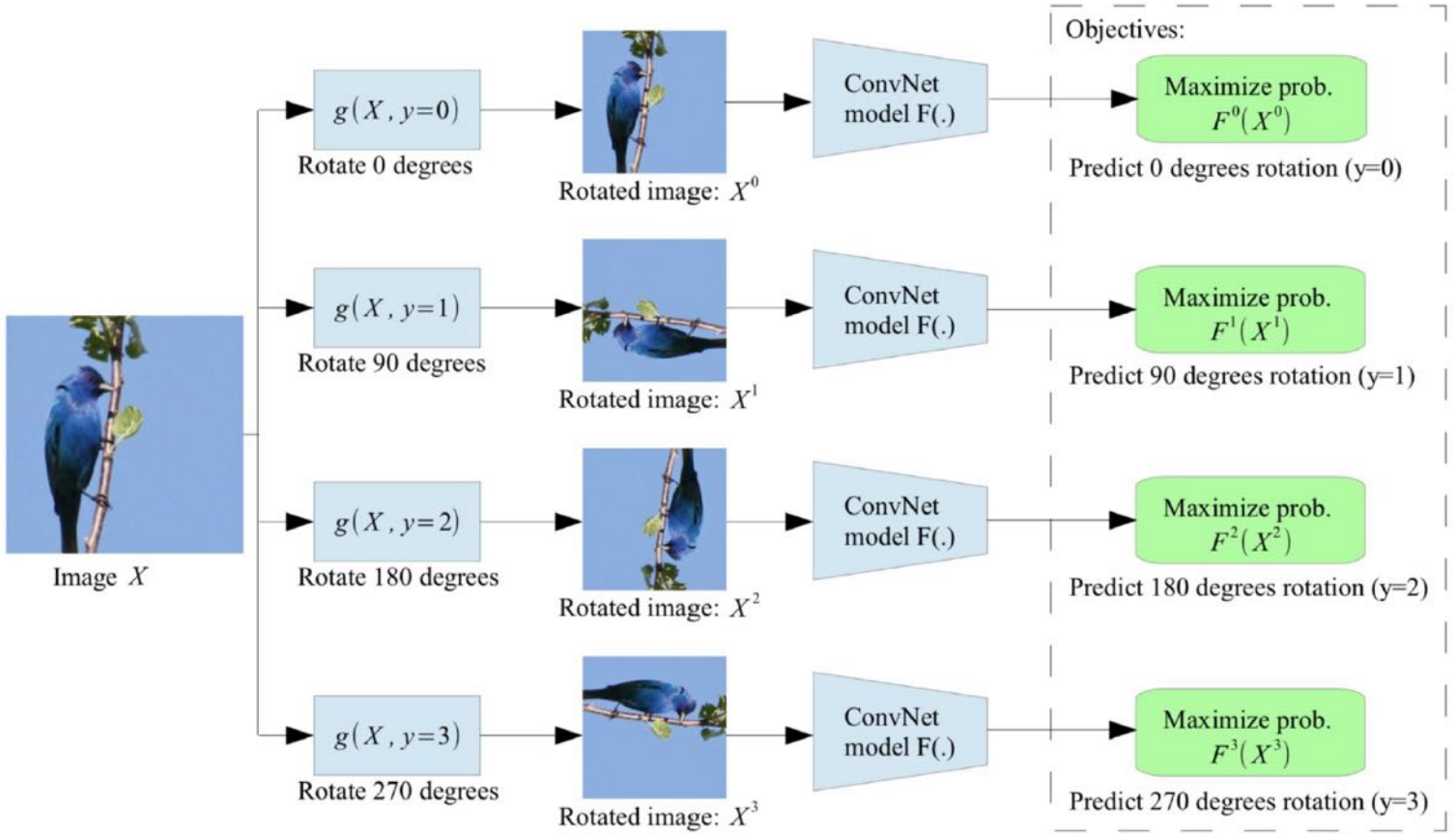
slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Pretext Task: Predict Rotations

- 4-way classification task: how much was each image rotated? (0, 90, 180, 270 degrees)
- Hypothesis: a model can predict the correct rotation of an image/object only if it has "visual commonsense" of what the image/object should look like unperturbed

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Pretext Task: Predict Rotations



Self-supervised learning by rotating the entire input images.

The model learns to predict which rotation is applied (4-way classification)

(Image source: Gidaris et al. 2018)

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

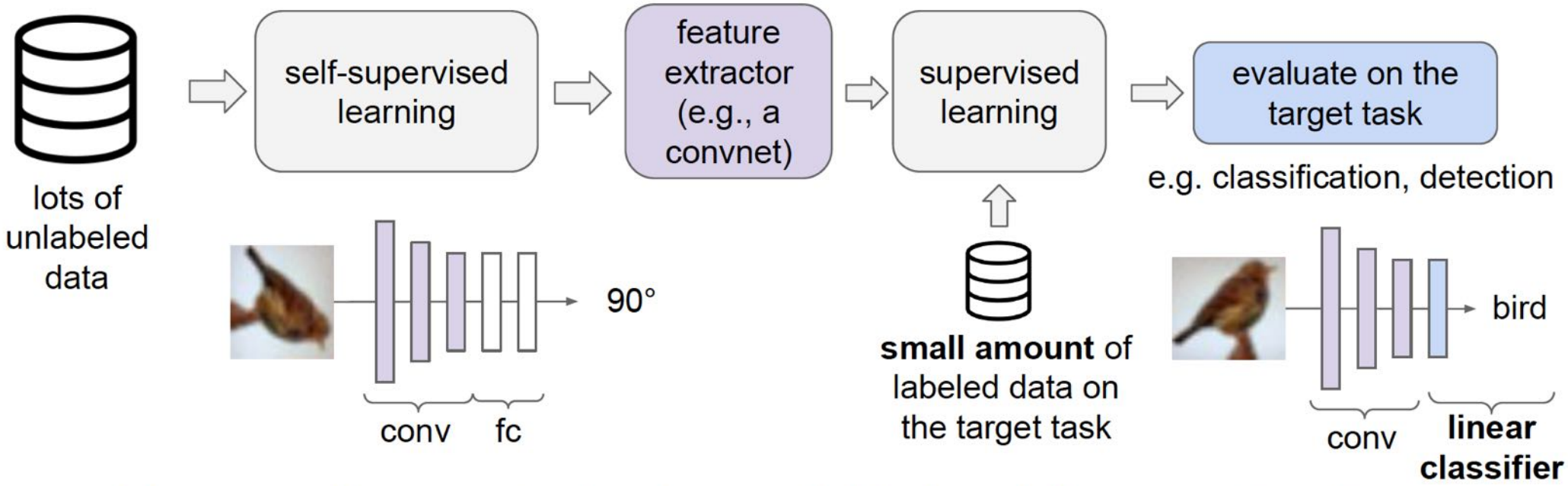
How to Evaluate a Self-Supervised Learning Method?

We usually don't care about the performance of the self-supervised learning task, e.g., we don't care if the model learns to predict image rotation perfectly.

Evaluate the learned feature encoders on downstream *target tasks*

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

How to Evaluate a Self-Supervised Learning Method?

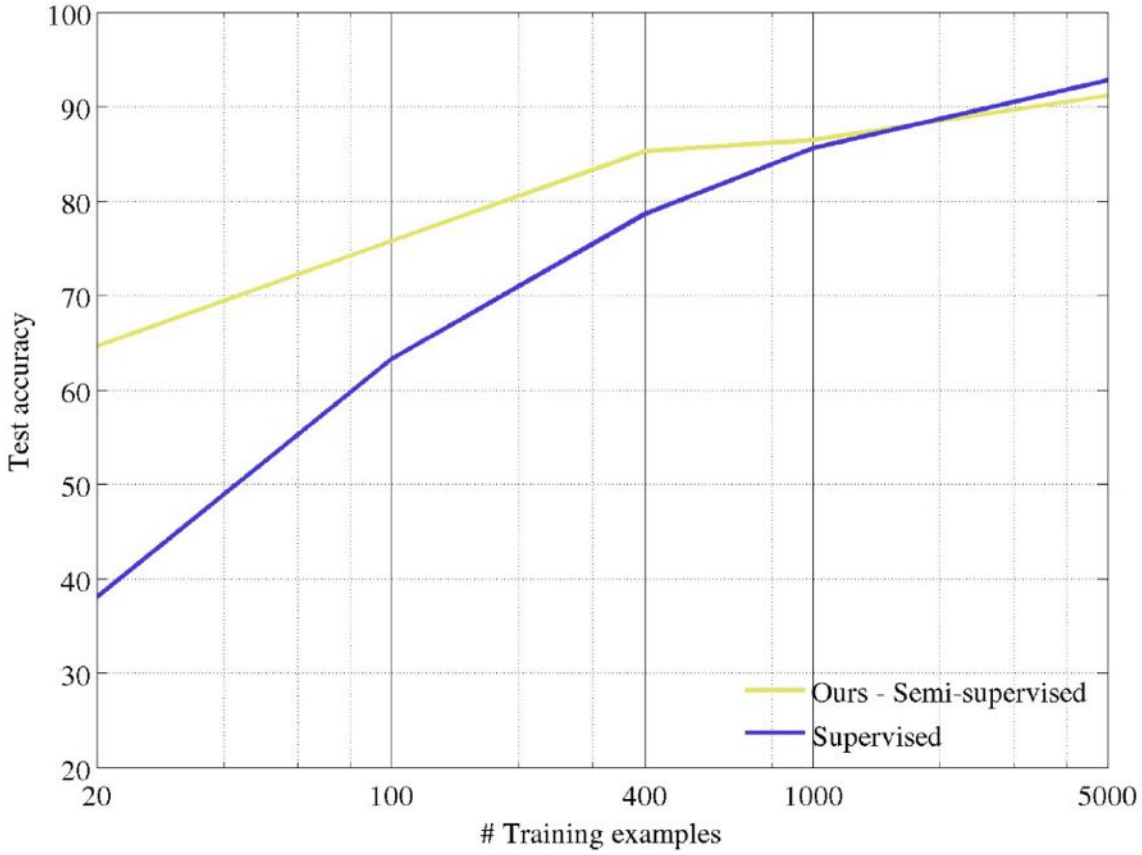


1. Learn good feature extractors from self-supervised pretext tasks, e.g., predicting image rotations

2. Attach a shallow network on the feature extractor; train the shallow network on the target task with small amount of labeled data

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Evaluation on Semi-Supervised Learning



Self-supervised learning on **CIFAR10** (entire training set).

Freeze conv1 + conv2
Learn **conv3 + linear** layers with subset of labeled CIFAR10 data (classification).

(Image source: Gidaris et al. 2018)

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Transfer of 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

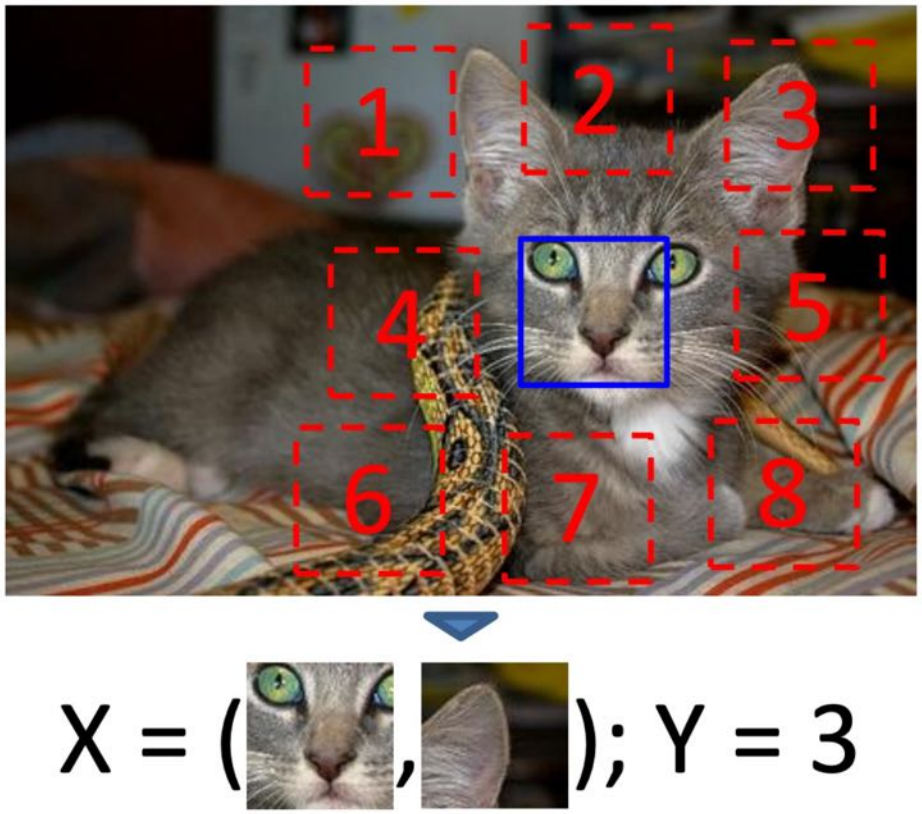
slide credit: Justin Johnson

Pretext Task: Predict Relative Patch Locations

Context Prediction

Model predicts relative location of two patches from the same image.
Discriminative pretraining task

Intuition: Requires understanding objects and their parts



Doersch et al, "Unsupervised Visual Representation Learning by Context Prediction", ICCV 2015

slide credit: Justin Johnson

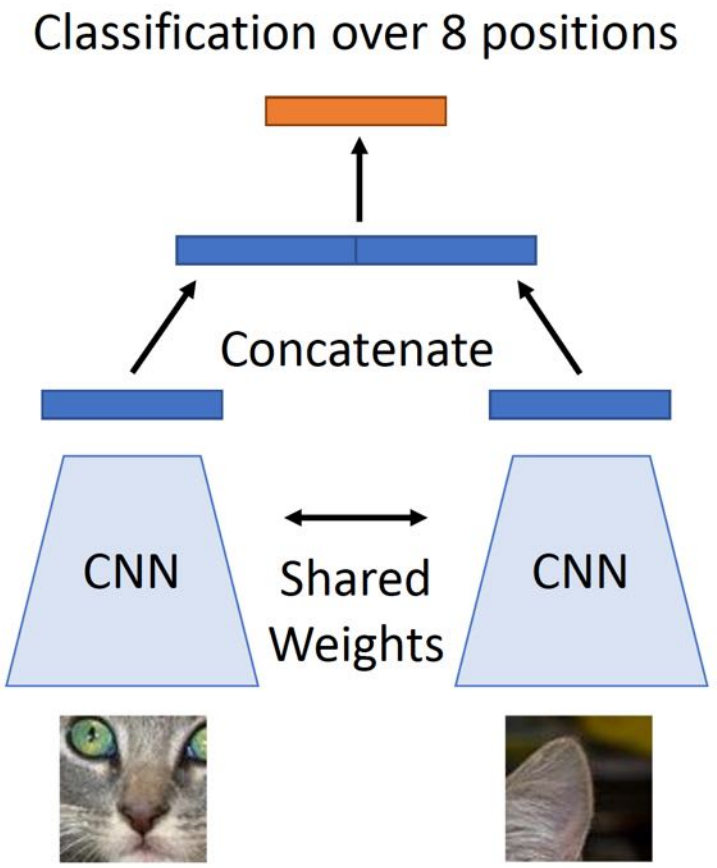
Pretext Task: Predict Relative Patch Locations

Context Prediction

Model predicts relative location of two patches from the same image.
Discriminative pretraining task

Intuition: Requires understanding objects and their parts

Two networks with shared weights sometimes called a "Siamese network" – but I don't really like this term

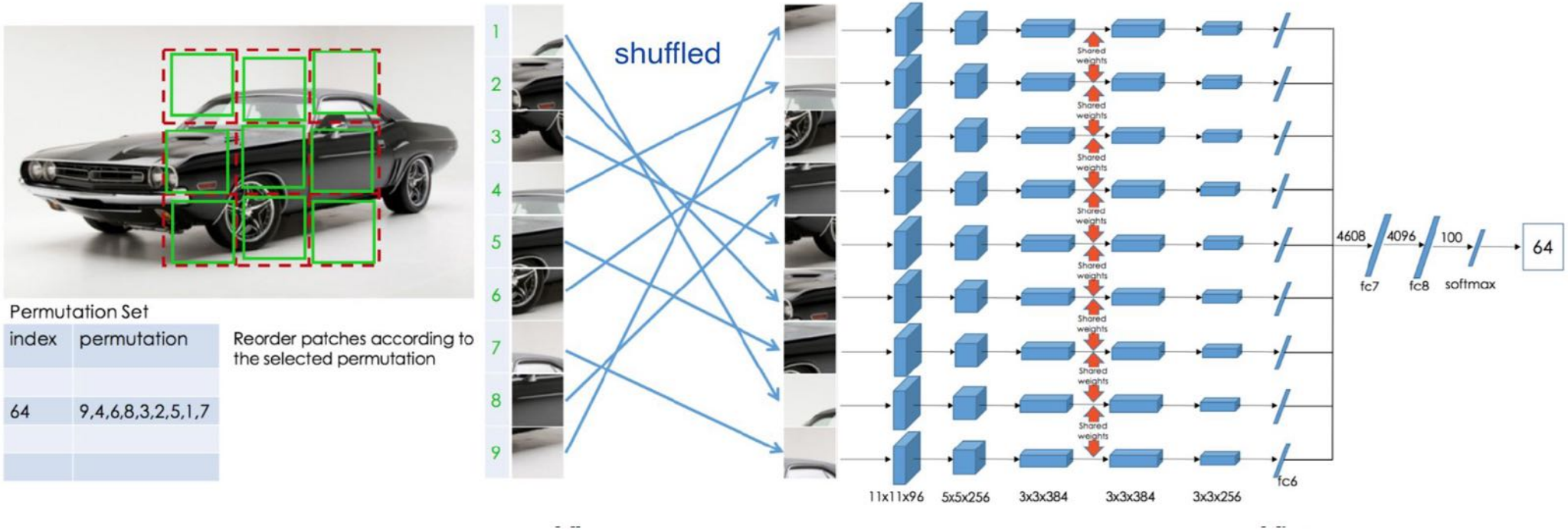


Doersch et al, "Unsupervised Visual Representation Learning by Context Prediction", ICCV 2015

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Pretext Task: Solving Jigsaw Puzzle

- Jigsaw puzzle: predict permutation to “unscramble” 9 shuffled patches
  - here: goal is to predict permutation index “64”



(Image source: Noroozi & Favaro, 2016)

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Transfer of Learned Features to Supervised Learning

Table 1: Results on PASCAL VOC 2007 Detection and Classification. The results of the other methods are taken from Pathak *et al.* [30].

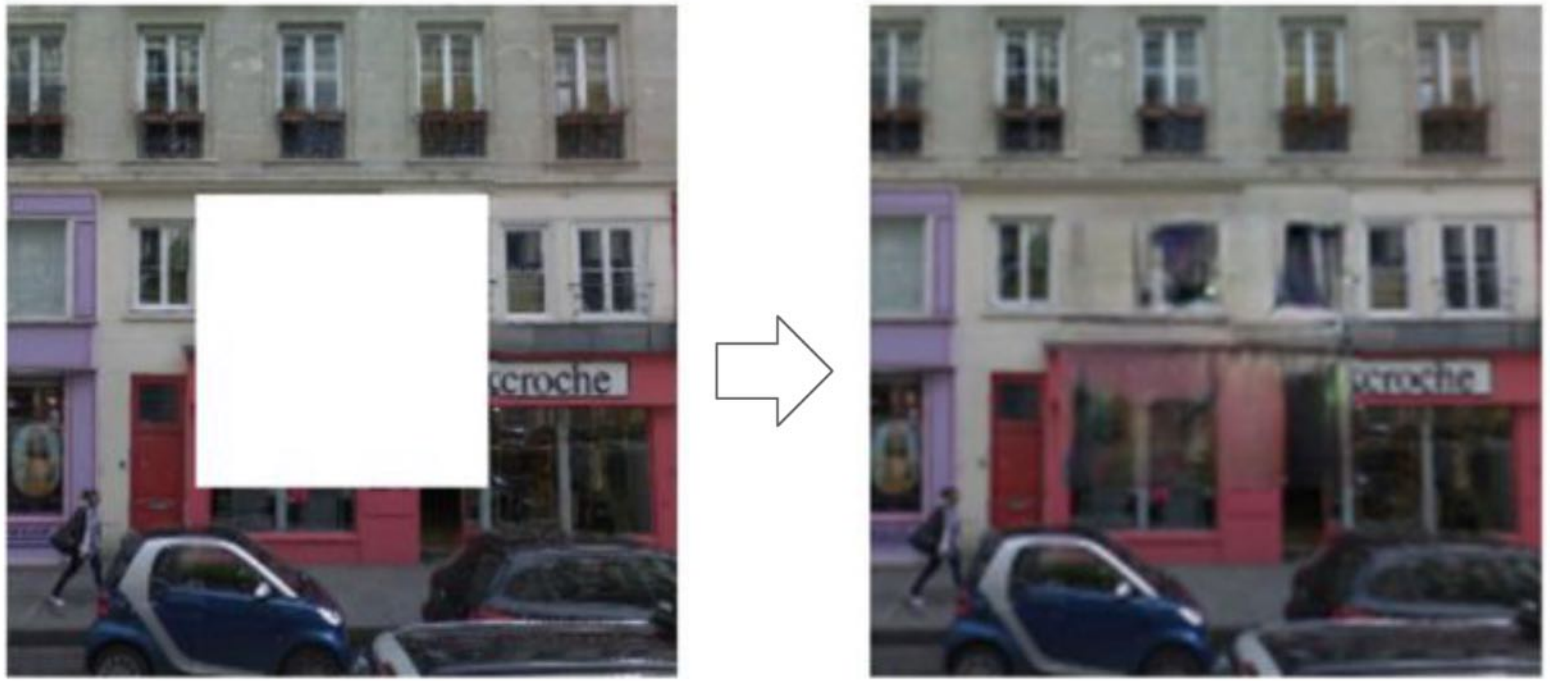
Method	Pretraining time	Supervision	Classification	Detection	Segmentation
Krizhevsky*et al.* [25]	3 days	1000 class labels	**78.2%**	**56.8%**	**48.0%**
Wang and Gupta[39]	1 week	motion	58.4%	44.0%	-
Doersch *et al.* [10]	4 weeks	context	55.3%	46.6%	-
Pathak *et al.* [30]	14 hours	context	56.5%	44.5%	29.7%
Ours	2.5 days	context	**67.6%**	**53.2%**	**37.6%**

"Ours" is feature learned from solving image Jigsaw puzzles (Noroozi & Favaro, 2016). Doersch et al. is the method with relative patch location

(source: Noroozi & Favaro, 2016)

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Pretext Task: Inpaiting (Predicting Missing Pixels)

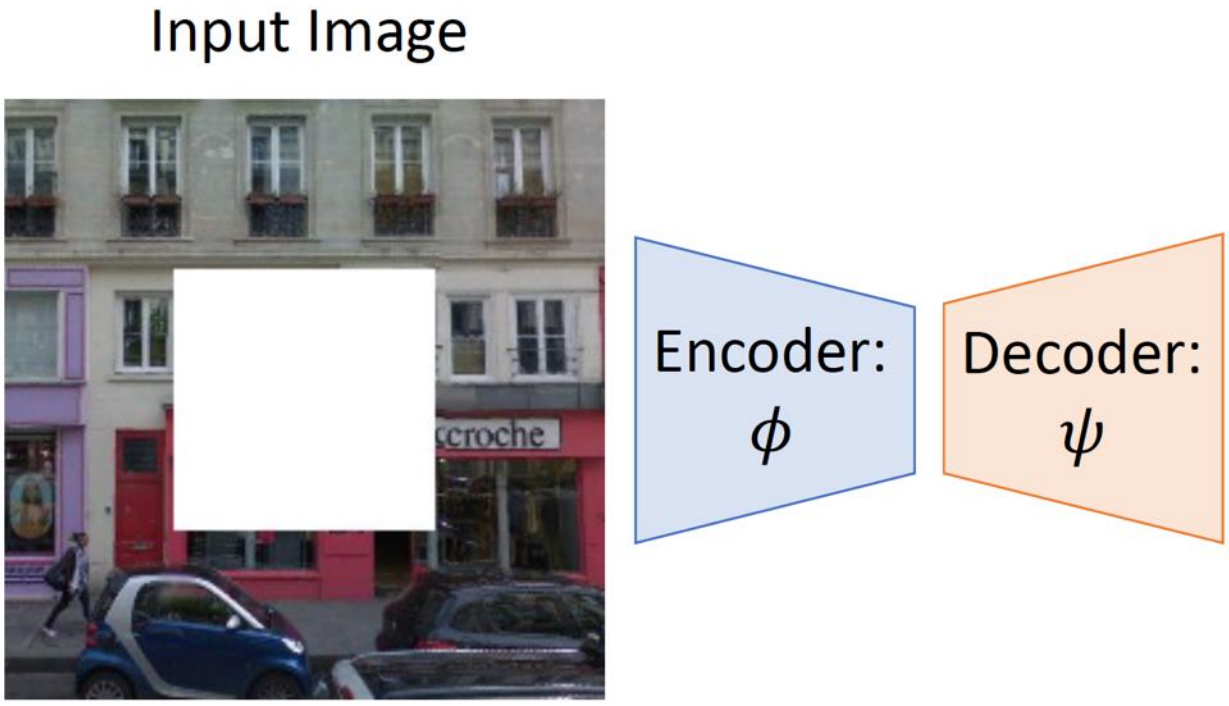


Context Encoders: Feature Learning by Inpainting (Pathak et al., 2016)

Source: Pathak et al., 2016

slide credit: Justin Johnson

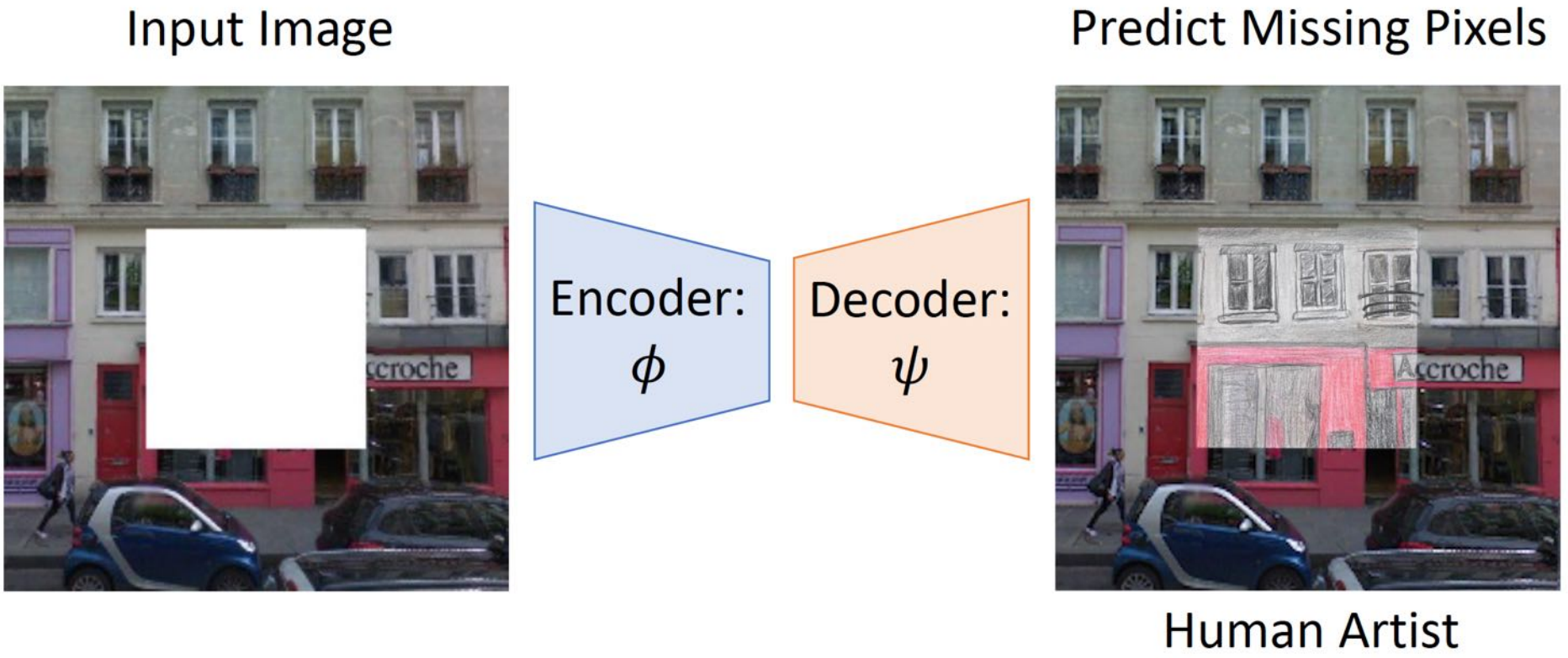
Pretext Task: Inpaiting (Predicting Missing Pixels)



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

slide credit: Justin Johnson

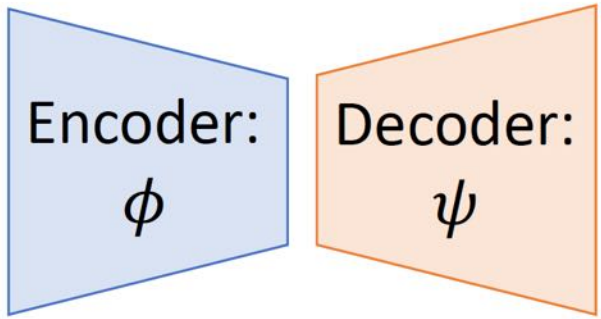
Pretext Task: Inpaiting (Predicting Missing Pixels)



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

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Learning to Inpaint by Reconstruction

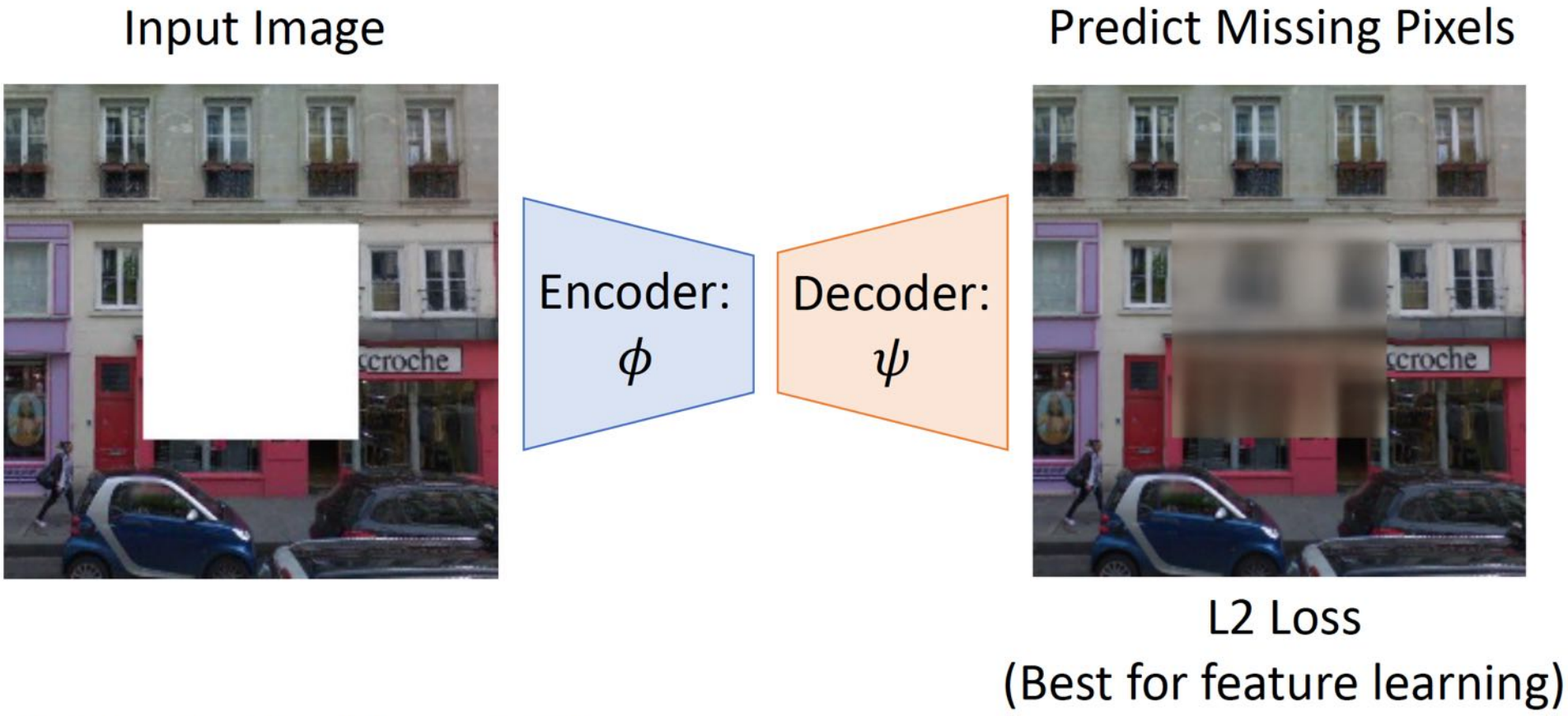


Learning to reconstruct the missing pixels

Source: Pathak et al., 2016

slide credit: Justin Johnson

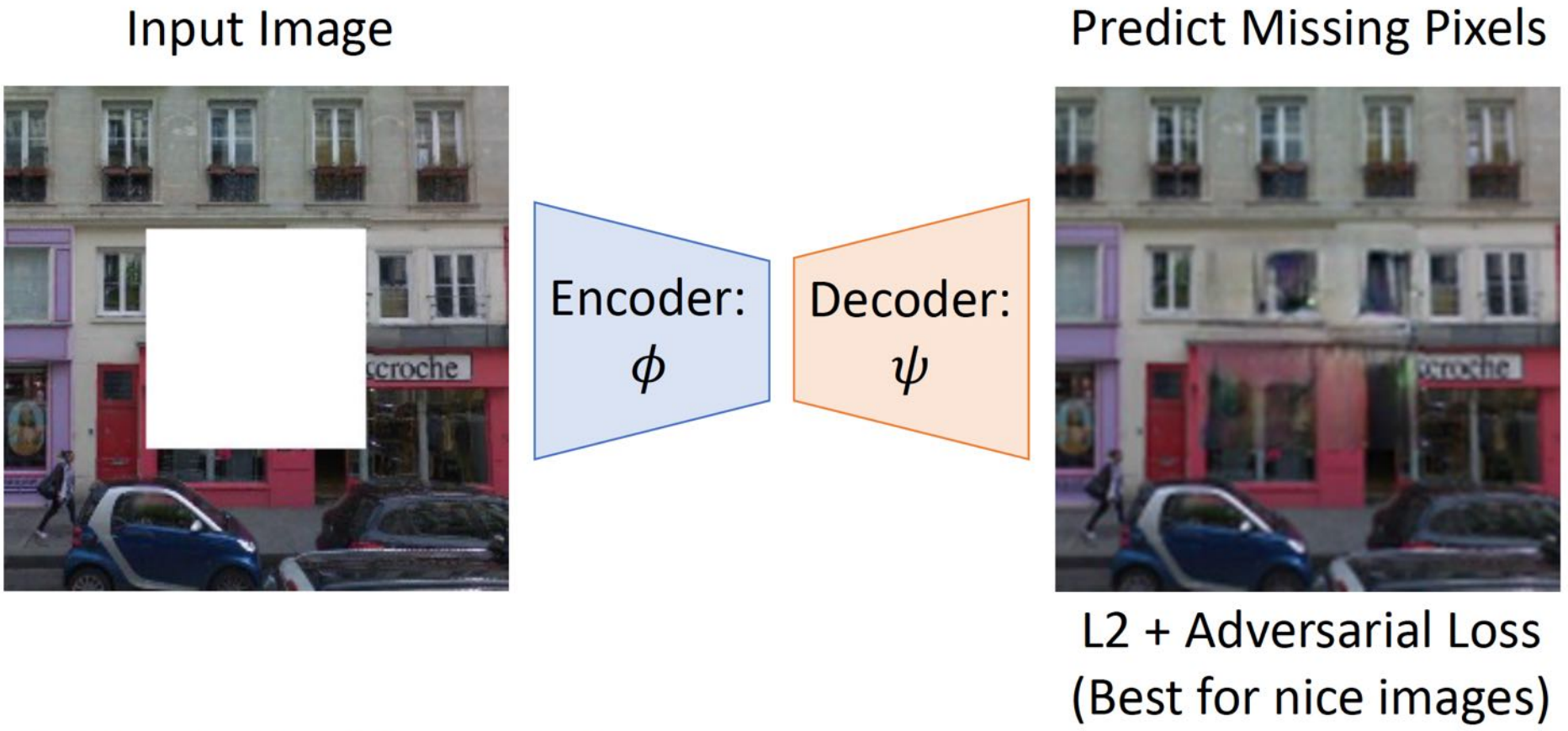
Pretext Task: Inpaiting (Predicting Missing Pixels)



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

slide credit: Justin Johnson

Pretext Task: Inpaiting (Predicting Missing Pixels)



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

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Inpainting Evaluation

Input (context) reconstruction adversarial recon + adv

Source: Pathak et al., 2016

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Transfer of Learned Features to Supervised Learning

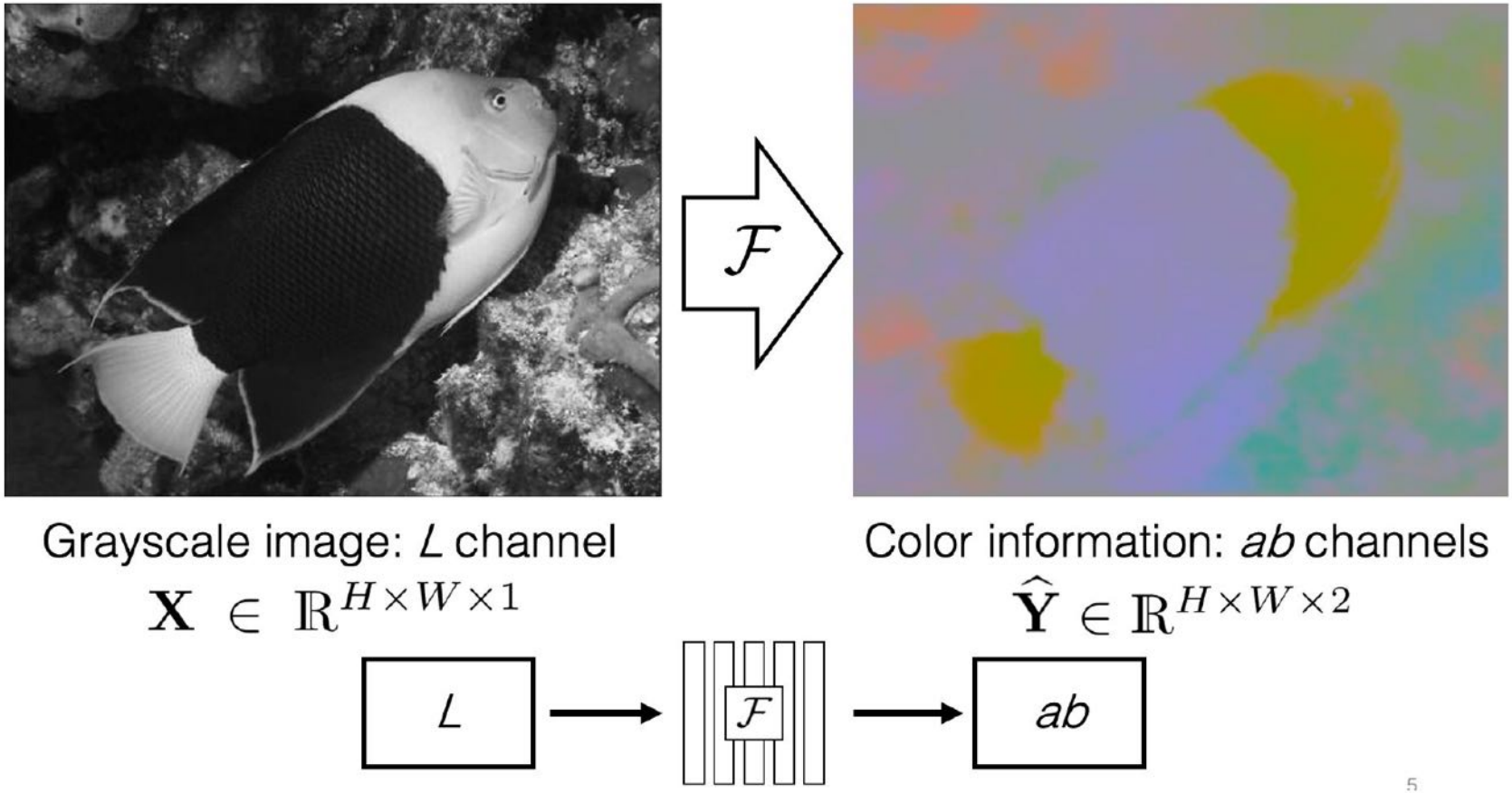
Pretraining Method	Supervision	Pretraining time	Classification	Detection	Segmentation
ImageNet [26]	1000 class labels	3 days	78.2%	56.8%	48.0%
Random Gaussian	initialization	< 1 minute	53.3%	43.4%	19.8%
Autoencoder	-	14 hours	53.8%	41.9%	25.2%
Agrawal *et al.* [1]	egomotion	10 hours	52.9%	41.8%	-
Wang *et al.* [39]	motion	1 week	58.7%	47.4%	-
Doersch *et al.* [7]	relative context	4 weeks	55.3%	46.6%	-
Ours	context	14 hours	56.5%	44.5%	30.0%

Self-supervised learning on ImageNet training set, transfer to classification (Pascal VOC 2007), detection (Pascal VOC 2007), and semantic segmentation (Pascal VOC 2012)

Source: Pathak et al., 2016

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

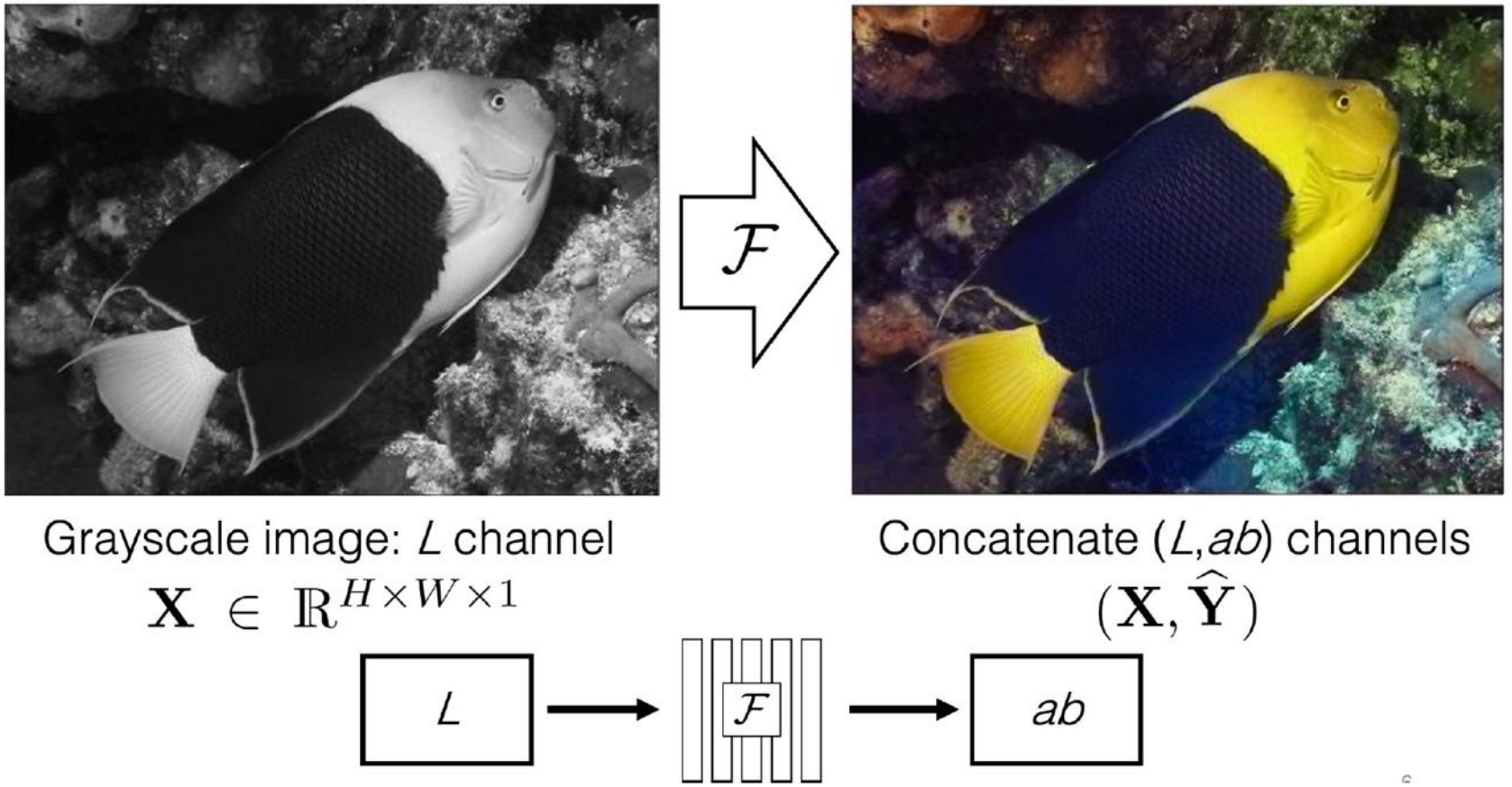
Pretext Task: Image Coloring



Source: Richard Zhang / Phillip Isola

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

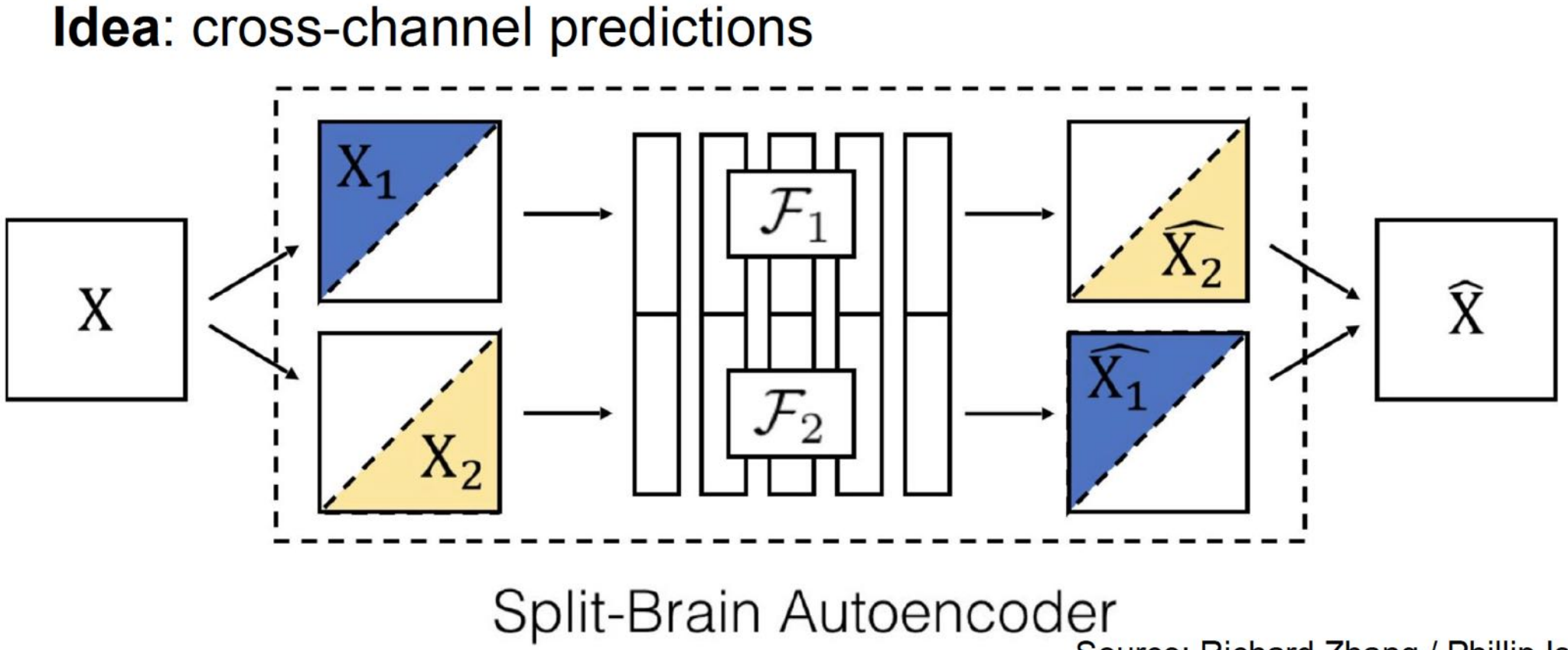
Pretext Task: Image Coloring



Source: Richard Zhang / Phillip Isola

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

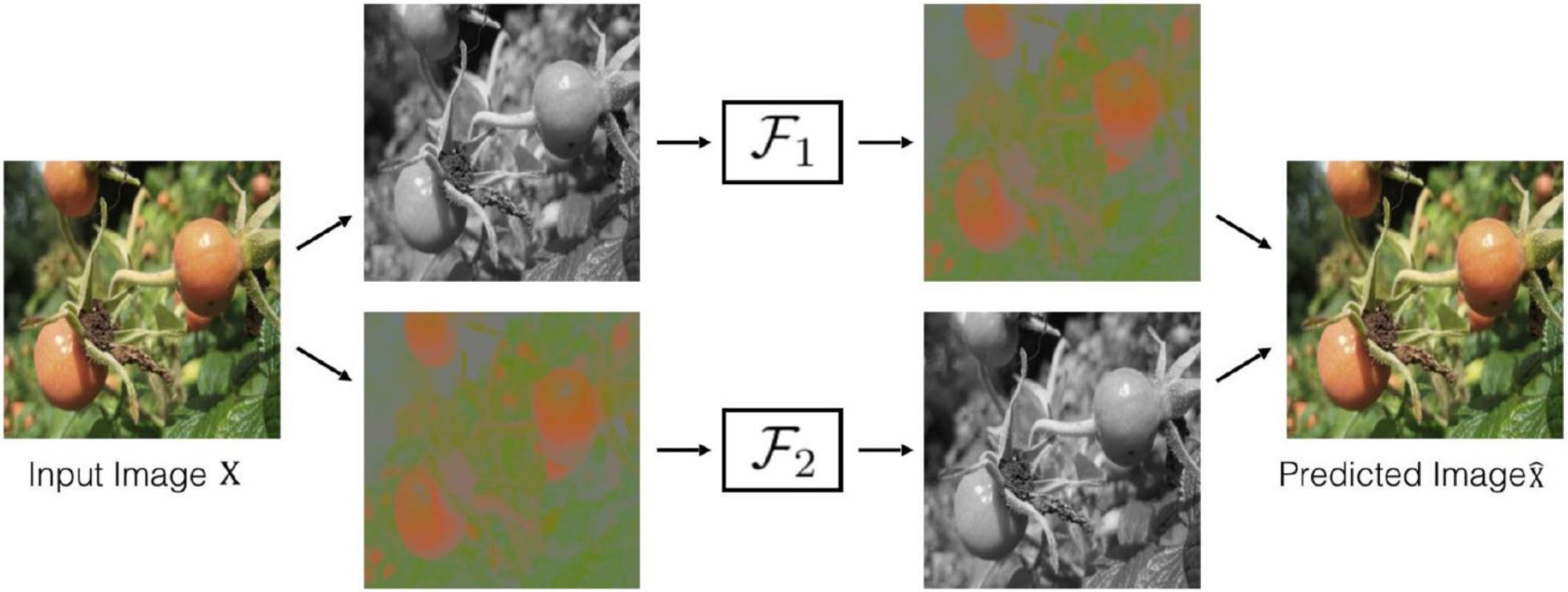
Learning Features from Colorization: Split-Brain Autoencoder



Source: Richard Zhang / Phillip Isola

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Learning Features from Colorization: Split-Brain Autoencoder



Source: Richard Zhang / Phillip Isola

Transfer of Learned Features to Supervised Learning

Zhang, Isola, Efros

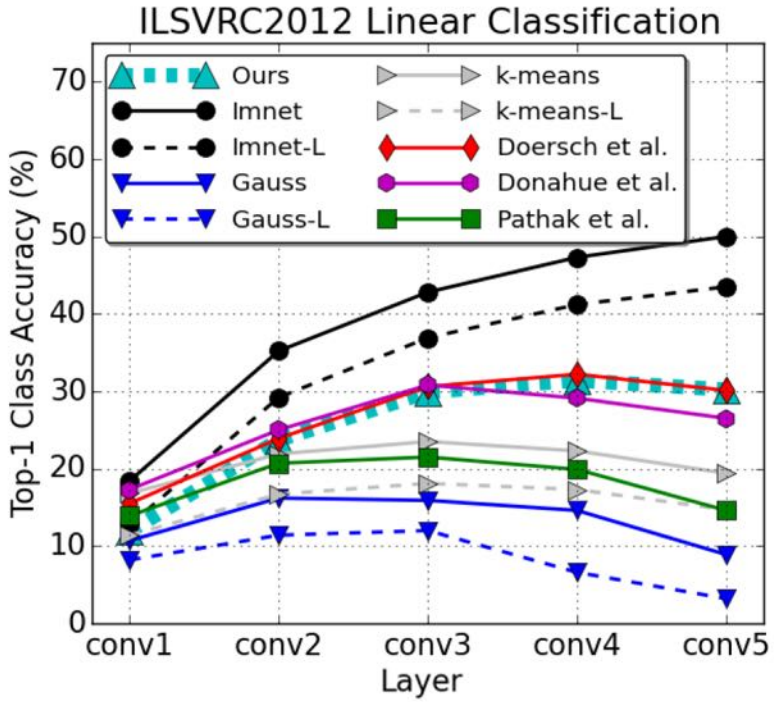


Fig. 7. ImageNet Linear Classification

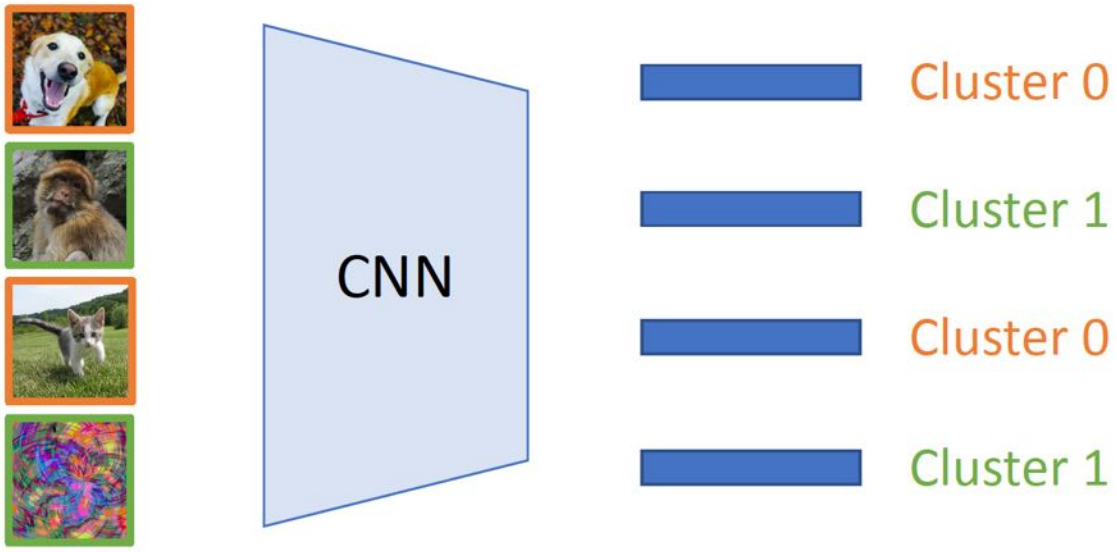
Dataset and Task Generalization on PASCAL [37]								
		Class. (%mAP)			Det. (%mAP)		Seg. (%mIU)	
fine-tune layers	**[Ref]**	**fc8**	**fc6-8**	**all**	**[Ref]**	**all**	**[Ref]**	**all**
ImageNet [38]	-	76.8	78.9	79.9	[36]	56.8	[42]	48.0
Gaussian	[10]	–	–	53.3	[10]	43.4	[10]	19.8
Autoencoder	[16]	24.8	16.0	53.8	[10]	41.9	[10]	25.2
k-means [36]	[16]	32.0	39.2	56.6	[36]	45.6	[16]	32.6
Agrawal et al. [8]	[16]	31.2	31.0	54.2	[36]	43.9	–	–
Wang & Gupta [15]	–	28.1	52.2	58.7	[36]	47.4	–	–
*Doersch et al. [14]	[16]	44.7	55.1	**65.3**	[36]	**51.1**	–	–
*Pathak et al. [10]	[10]	–	–	56.5	[10]	44.5	[10]	29.7
*Donahue et al. [16]	–	38.2	50.2	58.6	[16]	46.2	[16]	34.9
Ours (gray)	–	**52.4**	**61.5**	**65.9**	–	46.1	–	35.0
Ours (color)	–	**52.4**	**61.5**	**65.6**	–	46.9	–	**35.6**

Table 2. PASCAL Tests

slide credit: Justin Johnson

Deep Clustering

(1) Randomly initialize a CNN



(2) Run many images through CNN, get their final-layer features

(3) Cluster the features with K-Means; record cluster for each feature

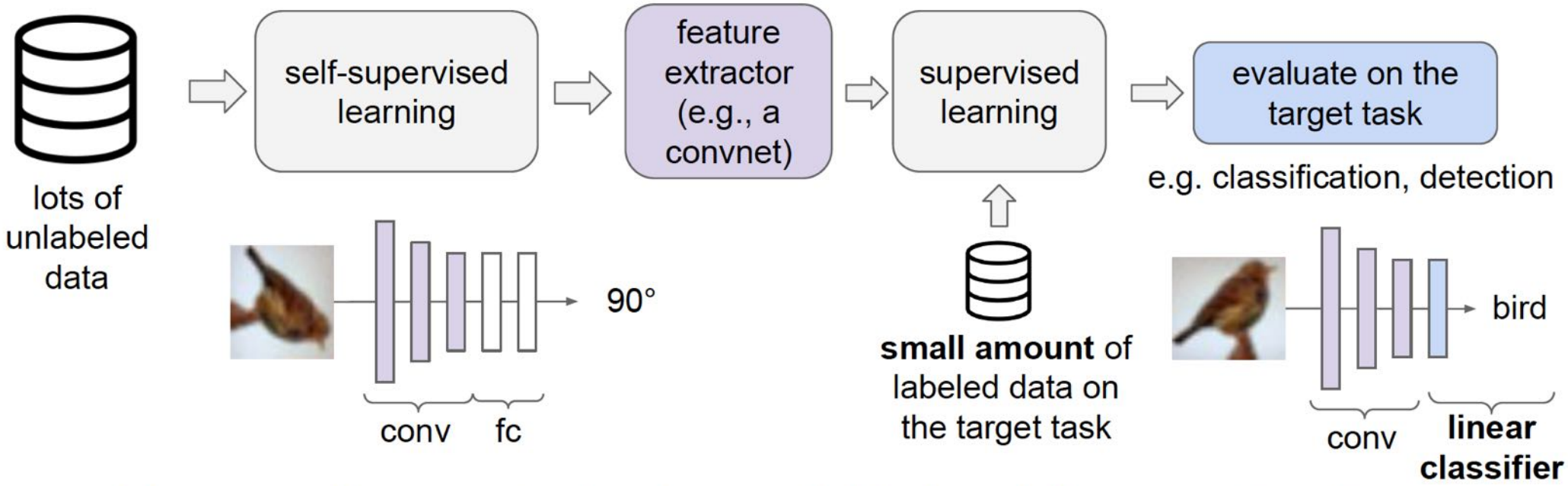
(4) Use cluster assignments as pseudo-labels for each image; train the CNN to predict cluster assignments

(5) Repeat: GOTO (2)

Caron et al, "Deep Clustering for Unsupervised Learning of Visual Features", ECCV 2018
Caron et al, "Unsupervised Pre-Training of Image Features on Non-Curated Data", ICCV 2019
Yan et al, "ClusterFit: Improving Generalization of Visual Representations", CVPR 2020
Caron et al, "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments", NeurIPS 2020

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

How to Evaluate a Self-Supervised Learning Method?



1. Learn good feature extractors from self-supervised pretext tasks, e.g., predicting image rotations

2. Attach a shallow network on the feature extractor; train the shallow network on the target task with small amount of labeled data

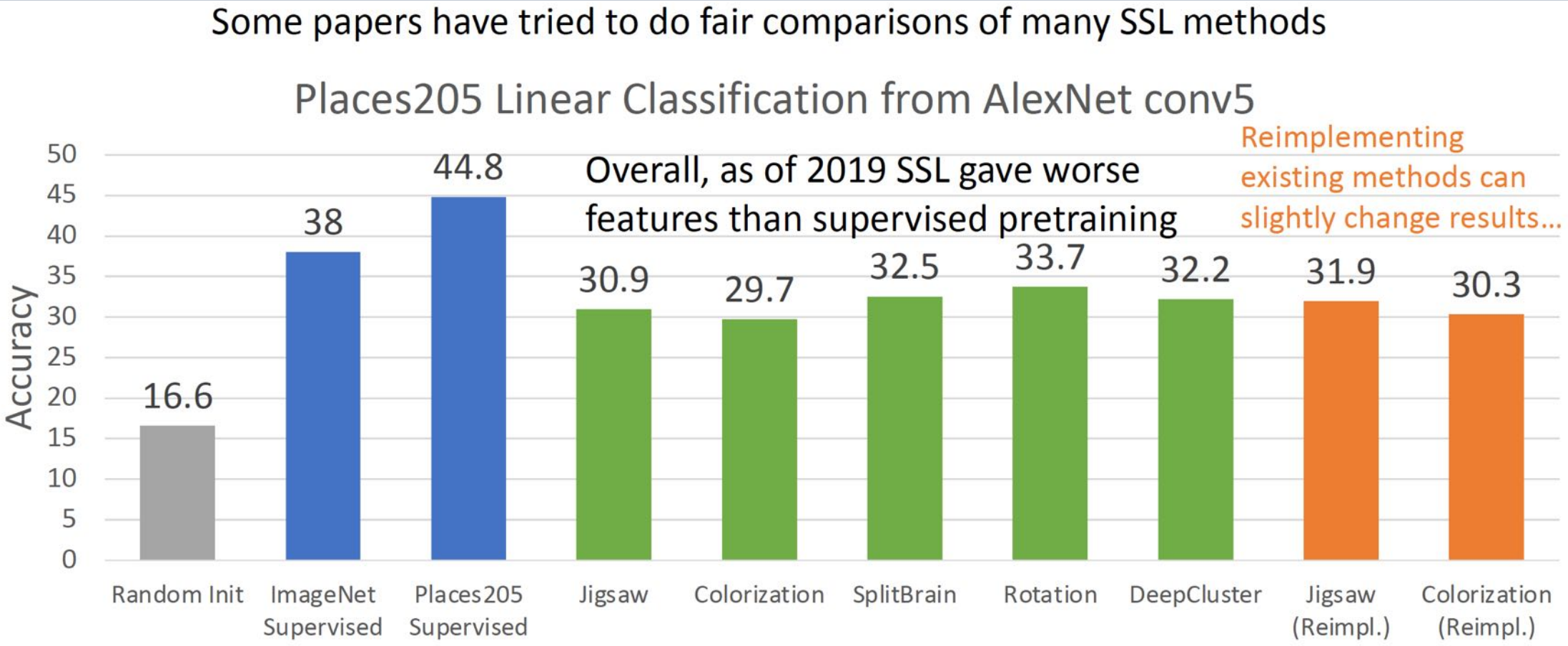
How to Evaluate a Self-Supervised Learning Method?

- Ideally
  - evaluate how well self-supervised learning performs on down-stream tasks…
  - however, ill-defined
    - what are the down-stream tasks
    - how much labeled data exists for them
    - how is supervised data used (lots of design choices: how is fine-tuning done, number of epochs, etc.)

- Today - two proxies are generally used for evaluation
  - linear probe:
    - after self-supervised learning — only a linear classifier is trained with labeled data
  - k-NN
    - after self-supervised learning — the learned representation is used for a k-nearest neighbor classifier

slide credit: Justin Johnson

Comparison of SSL Methods…



Goyal et al, "Scaling and Benchmarking Self-Supervised Visual Representation Learning", ICCV 2019

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Summary: Pretext Tasks based on Image Transformations

- Pretext tasks focus on “visual common sense”, e.g., predict rotations, inpainting, rearrangement, and colorization.
- The models are forced learn good features about natural images, e.g., semantic representation of an object category, in order to solve the pretext tasks.
- We don’t care about the performance of these pretext tasks, but rather how useful the learned features are for downstream tasks (classification, detection, segmentation).
- Problems: 1) coming up with individual pretext tasks is tedious, and 2) the learned representations may not be general.

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Pretext Tasks from Image Transformations

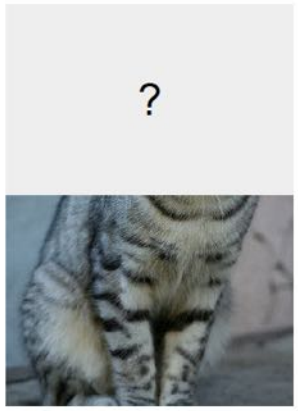
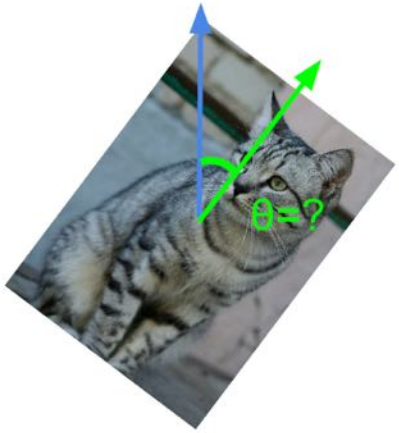


image completion



rotation prediction

"jigsaw puzzle"

colorization

Learned representations may be tied to a specific pretext task!

Can we come up with a more general pretext task?

Overview of Today's Lecture

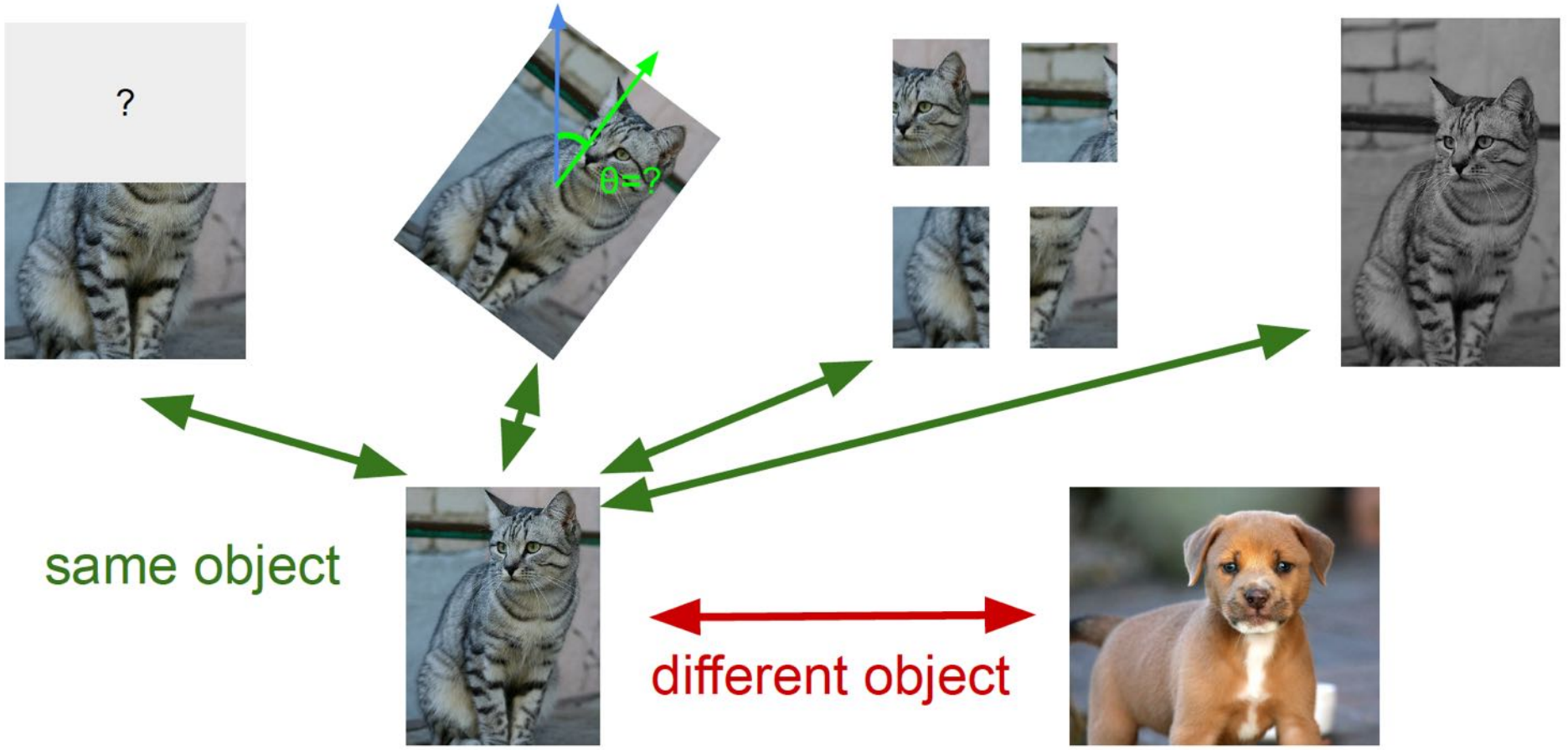
- U-Net Architecture for Semantic Segmentation

- Self-Supervised Learning - Part 1 (today):
  - Motivation of Self-Supervised Learning
    - setting of pretext tasks — how to evaluate
  - Pretext tasks from image transformations
    - rotation, inpainting, rearrangement, coloring
  - Contrastive representation learning
    - intuition and formulation
    - instance contrastive learning: SimCLR & MOCO
    - sequence contrastive learning: CPC

- Self-Supervised Learning - Part 2 (next time):
  - Teacher-Student "feature reconstruction"
    - motivation, setting
    - methods: BYOL, DINO

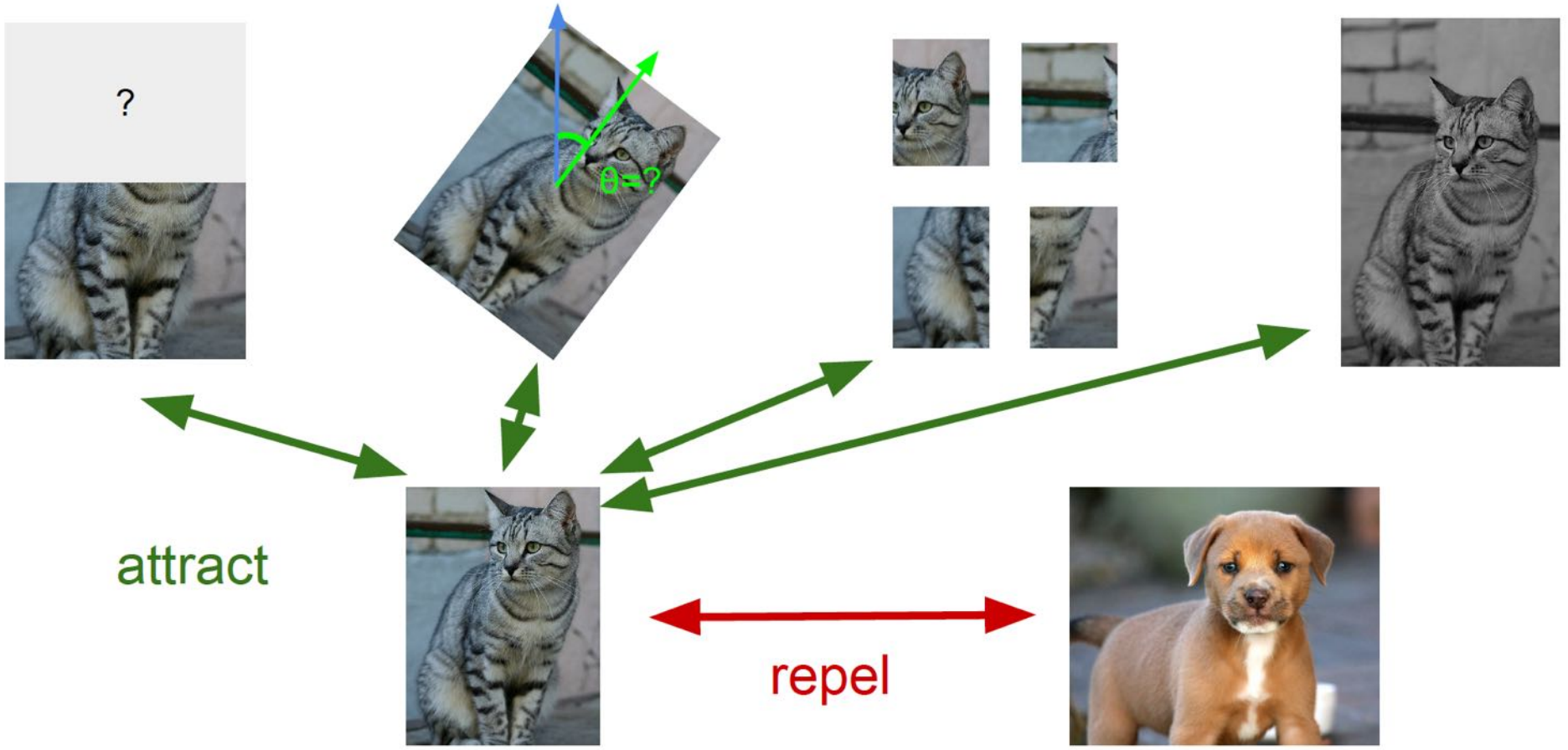
slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

A more General Pretext Task?



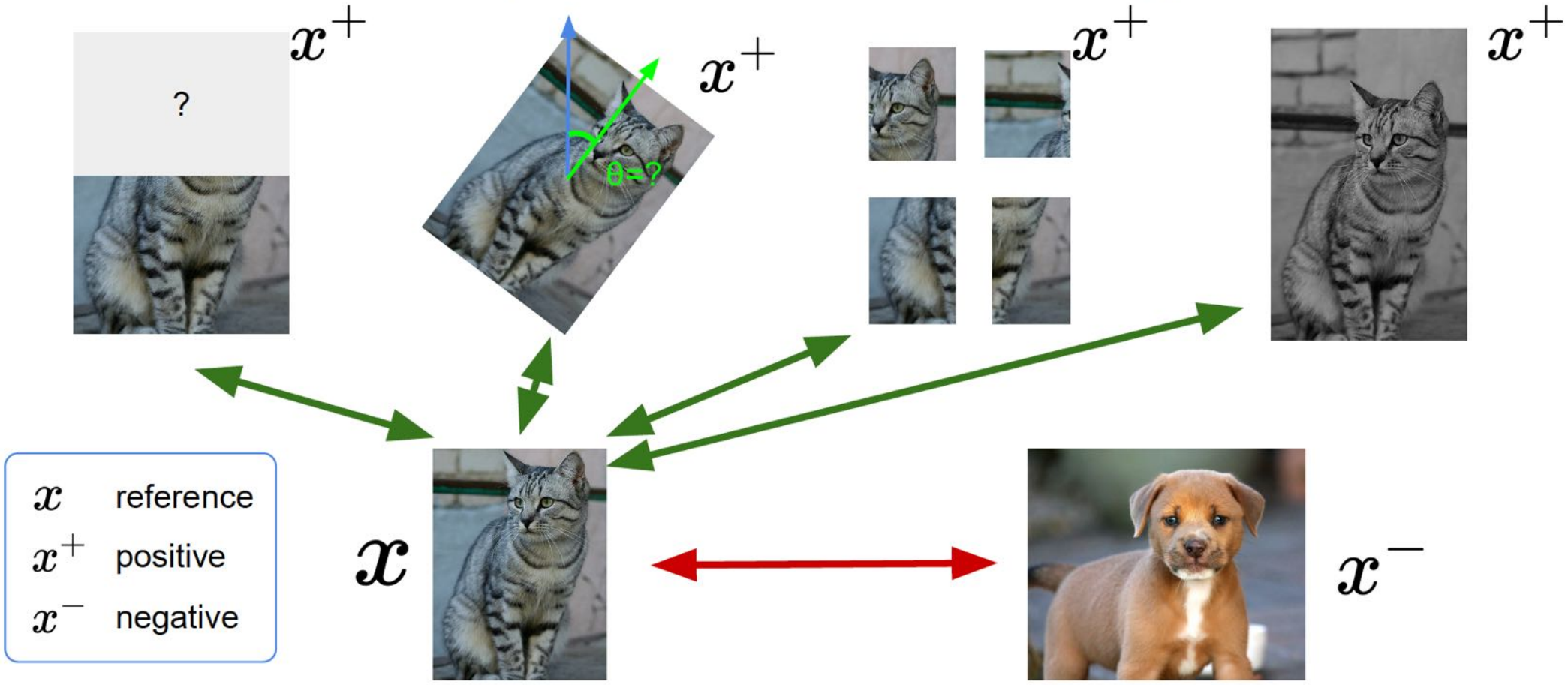
slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Contrastive Representation Learning



slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Contrastive Representation Learning



slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

A Formulation of Contrastive Learning

What we want:

$$\text{score}(f(x), f(x^+)) >> \text{score}(f(x), f(x^-))$$

x: reference sample; x^+ positive sample; x^- negative sample

Given a chosen score function, we aim to learn an **encoder function** f that yields high score for positive pairs (x, x^+) and low scores for negative pairs (x, x^-).

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

A Formulation of Contrastive Learning

Loss function given 1 positive sample and N - 1 negative samples:

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+))}{\exp(s(f(x), f(x^+)) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))} \right]$$

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

A Formulation of Contrastive Learning

Loss function given 1 positive sample and N - 1 negative samples:

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+))}{\exp(s(f(x), f(x^+)) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))} \right]$$

x

x^+

x

x_1^-

x_2^-

x_3^-

...

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

A Formulation of Contrastive Learning

Loss function given 1 positive sample and N - 1 negative samples:

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+))}{\exp(s(f(x), f(x^+)) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))} \right]$$

score for the positive pair

score for the N-1 negative pairs

This seems familiar …

Cross entropy loss for a N-way softmax classifier!

I.e., learn to find the positive sample from the N samples

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

A Formulation of Contrastive Learning

Loss function given 1 positive sample and N - 1 negative samples:

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+))}{\exp(s(f(x), f(x^+)) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))} \right]$$

Commonly known as the InfoNCE loss (van den Oord et al., 2018)

A *lower bound* on the mutual information between $f(x)$ and $f(x^+)$

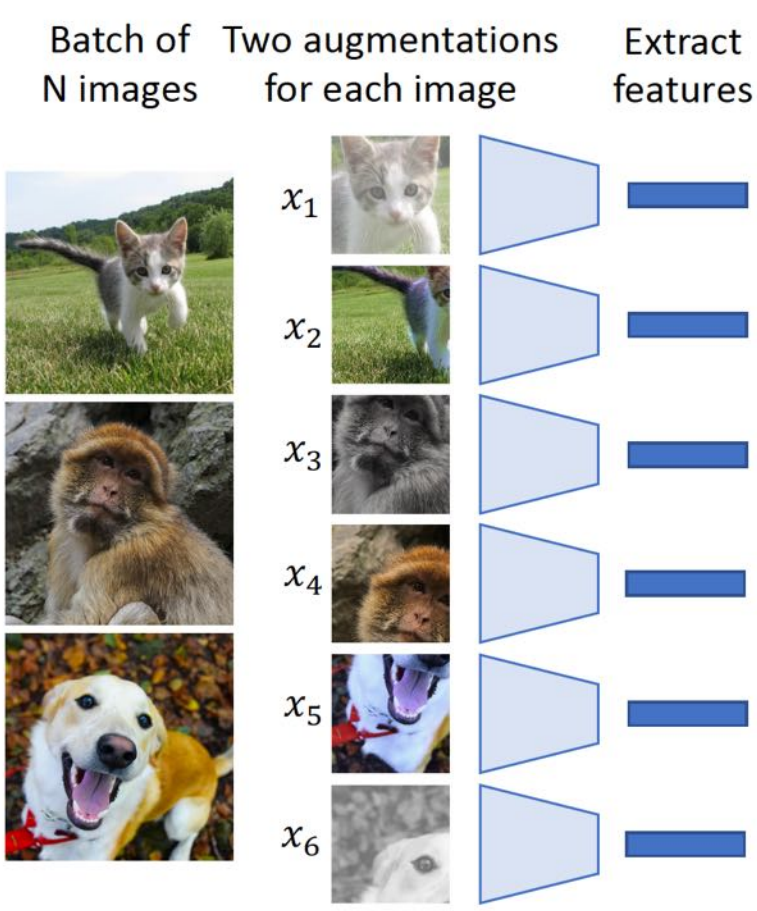
$$MI[f(x), f(x^+)] \geq \log(N) - \mathcal{L}$$

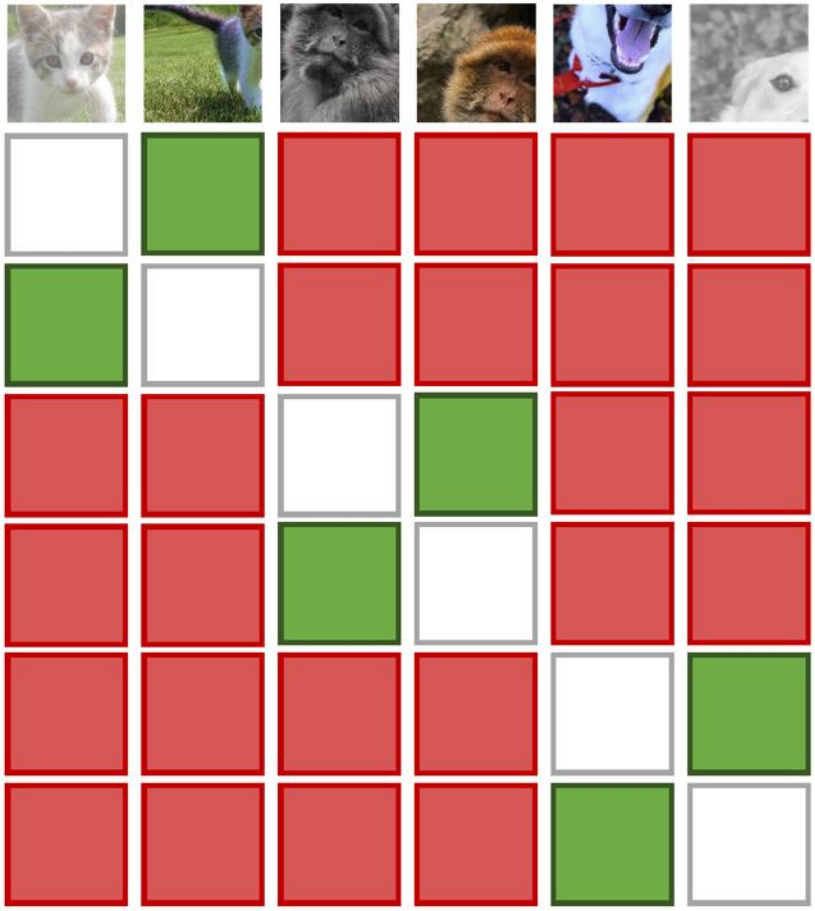
The larger the negative sample size (N), the tighter the bound

Detailed derivation: Poole et al., 2019

slide credit: Justin Johnson

Contrastive Learning with Data Augmentation





Each image tries to predict which of the *other* 2N-1 images came from the same original image

Similarity between x_i and x_j:

$$s_{i,j} = \frac{\phi(x_i)^T \phi(x_j)}{\|\phi(x_i)\| \cdot \|\phi(x_i)\|}$$

If (x_i, x_j) is a positive pair, then loss for x_i is:

$$L_i = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{\substack{k=1 \\ k \neq i}}^{2N} \exp(s_{i,k}/\tau)}$$

(τ is a *temperature*)

Interpretation: Cross-entropy loss over the other 2N-1 elements in the batch!

Hadsell et al, "Dimensionality Reduction by Learning and Invariant Mapping", CVPR 2006
Wu et al, "Unsupervised Feature Learning by Non-Parametric Instance-Level Discrimination", CVPR 2018
Van den Oord et al, "Representation Learning with Contrastive Predictive Coding", NeurIPS 2018

Hjelm et al, "Learning deep representations by mutual information estimation and maximization", ICLR 2019
Bachman et al, "Learning Representations by Maximizing Mutual Information Across Views", NeurIPS 2019
Henaff et al, "Data-Efficient Image Recognition with Contrastive Predictive Coding", ICML 2020

Tian et al, "Contrastive Multiview Coding", ECCV 2020
He et al, "Momentum Contrast for Unsupervised Visual Representation Learning", CVPR 2020
Chen et al, "A Simple Framework for Contrastive Learning of Visual Representations", ICML 2020

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

SimCLR: a Simple Framework for Contrastive Learning

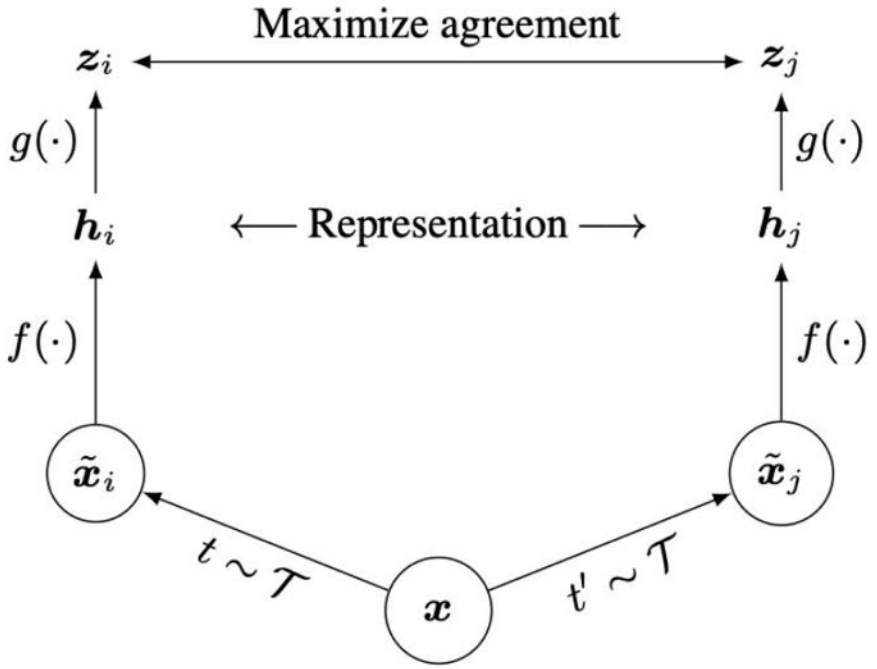
Cosine similarity as the score function:

$$s(u, v) = \frac{u^T v}{||u||||v||}$$

Use a projection network ***g(·)*** to project features to a space where contrastive learning is applied

Generate positive samples through data augmentation:

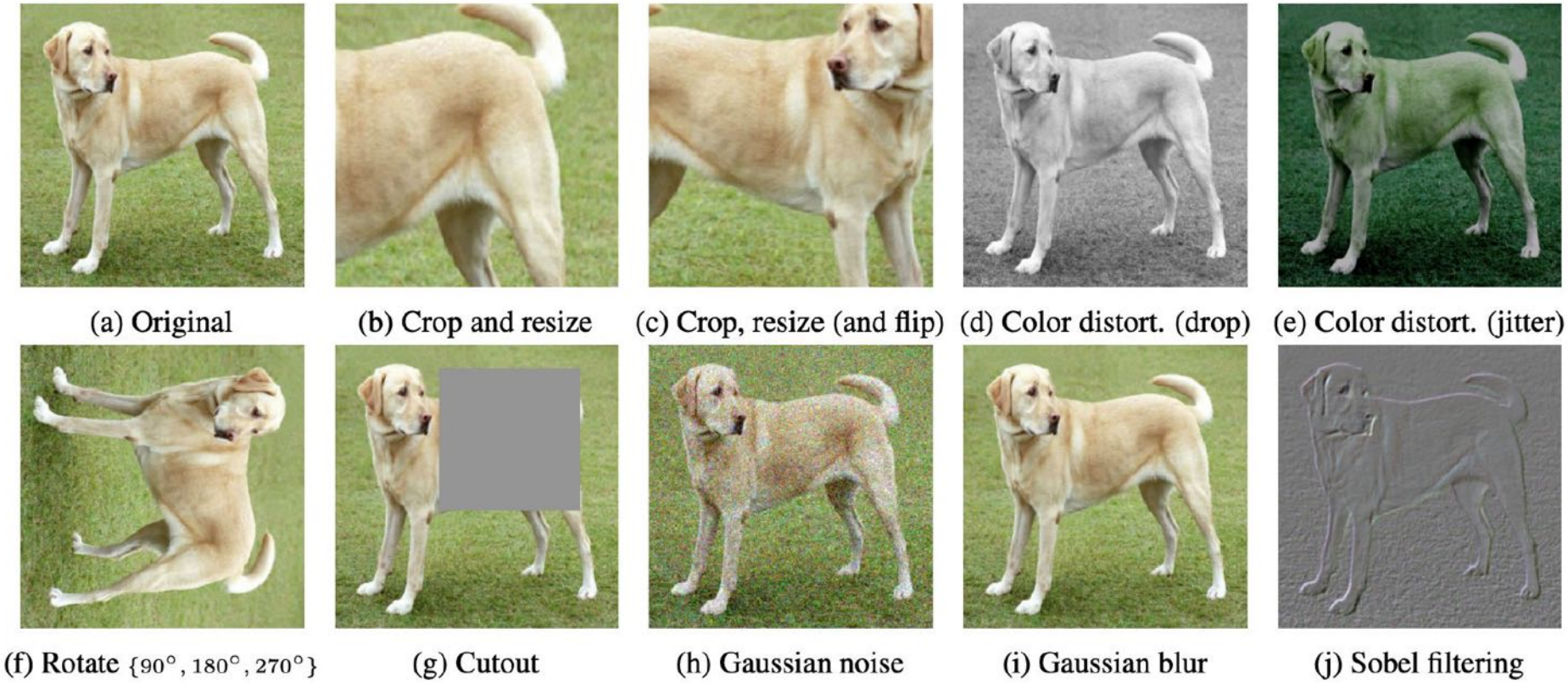
- random cropping, random color distortion, and random blur.



Source: Chen et al., 2020

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

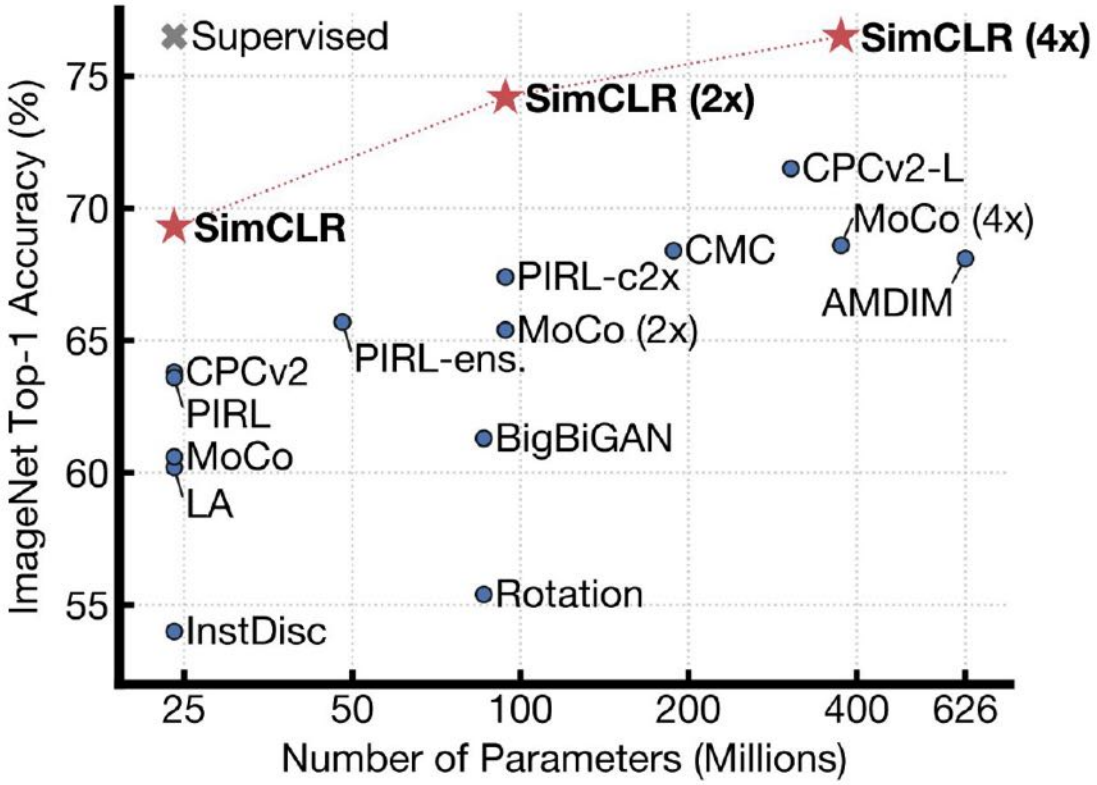
SimCLR: Generating Positive Samples by Data Augmentation



Source: Chen et al., 2020

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Transfer Using 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

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Semi-Supervised Learning based on SimCLR Features

Method	Architecture	Label fraction 1%	Label fraction 10%
		Top 5	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.

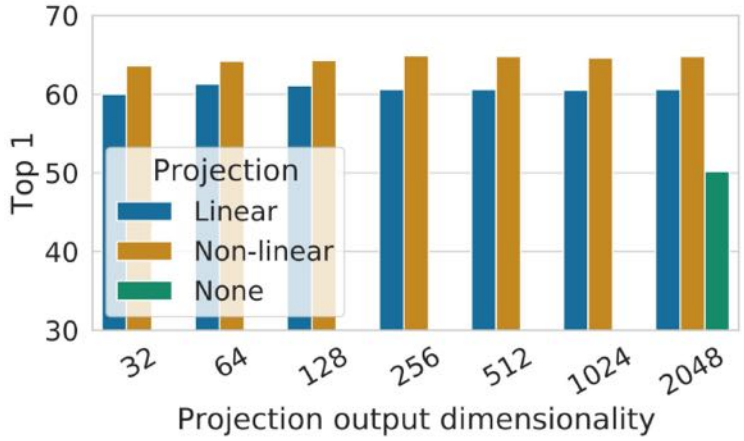
Train feature encoder on **ImageNet** (entire training set) using SimCLR.

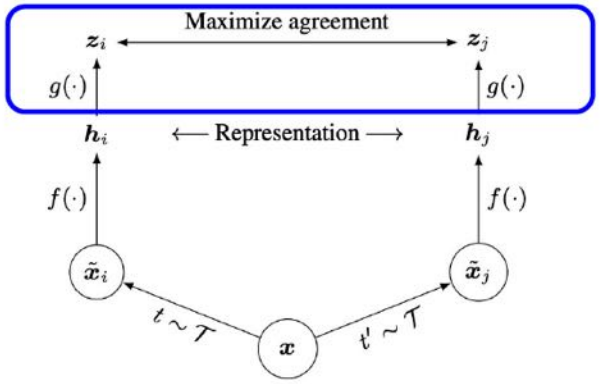
Finetune the encoder with 1% / 10% of labeled data on ImageNet.

Source: Chen et al., 2020

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

SimCLR Design Choices: Projection Head





Linear / non-linear projection heads improve representation learning.

A possible explanation:

- contrastive learning objective may discard useful information for downstream tasks
- representation space ***z*** is trained to be invariant to data transformation.
- by leveraging the projection head **g(·)**, more information can be preserved in the **h** representation space

Source: Chen et al., 2020

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

SimCLR Design Choices: Large Batch Size

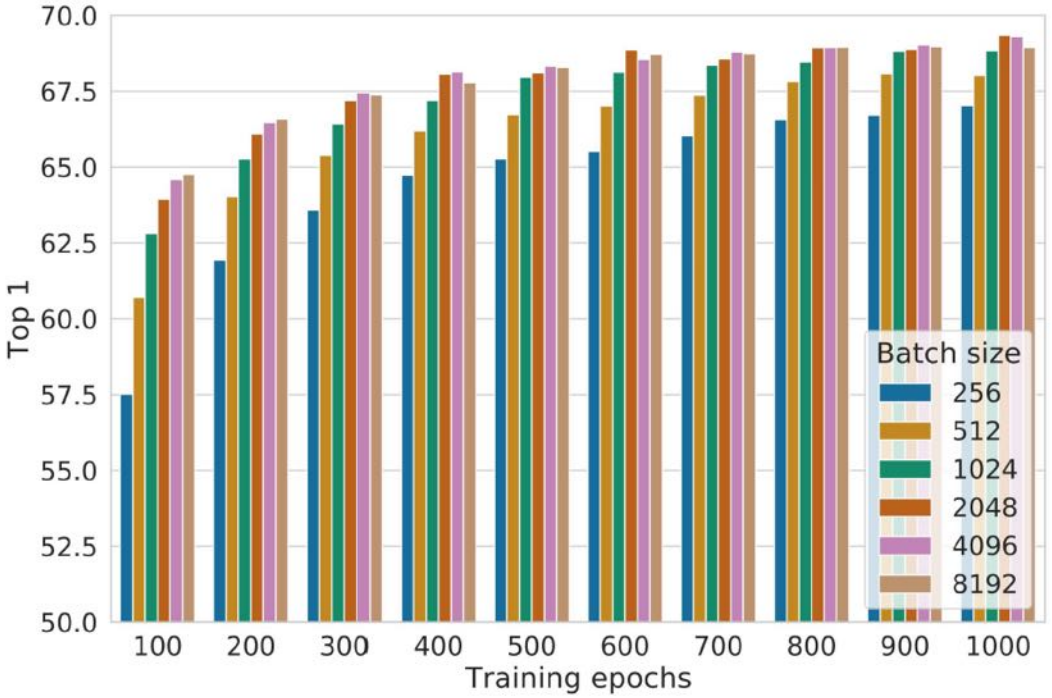


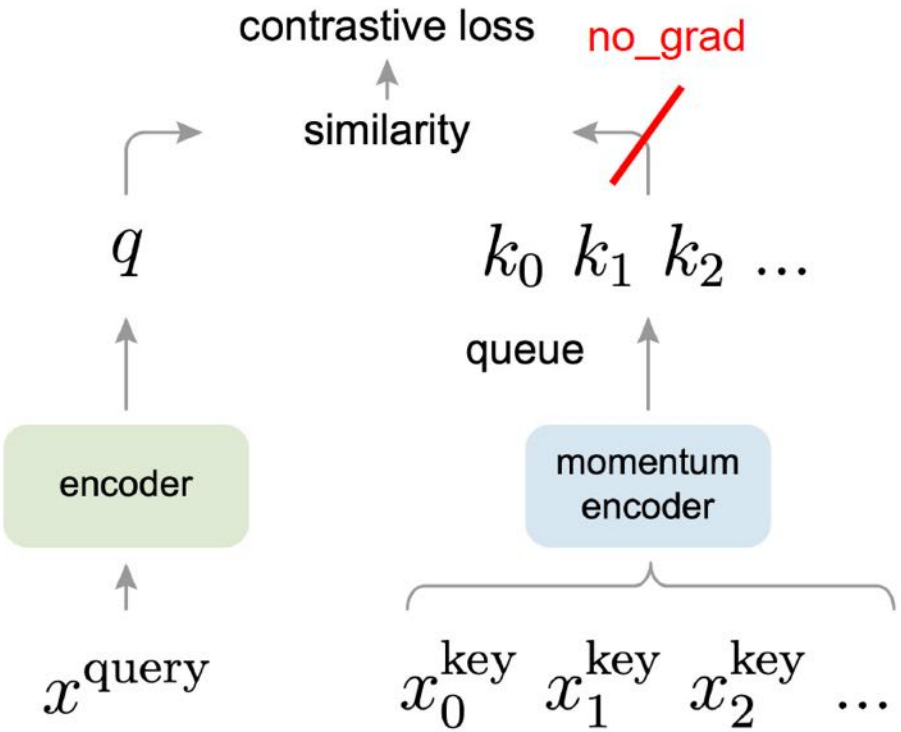
Figure 9. Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch.[10]

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

Momentum Contrastive Learning (MoCo)



Key differences to SimCLR:

- Keep a running queue of keys (negative samples).
- Compute gradients and update the encoder only through the queries.
- Decouple min-batch size with the number of keys: can support a large number of negative samples.
- The key encoder is slowly progressing through the momentum update rules:

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

Source: He et al., 2020

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

MoCo V2…

Improved Baselines with Momentum Contrastive Learning

Xinlei Chen Haoqi Fan Ross Girshick Kaiming He
Facebook AI Research (FAIR)

A hybrid of ideas from SimCLR and MoCo:

- **From SimCLR**: non-linear projection head and strong data augmentation.
- **From MoCo**: momentum-updated queues that allow training on a large number of negative samples (no TPU required!).

Source: Chen et al., 2020

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

SimCLR vs. MoCo V1 vs. MoCo V2

case	unsup. pre-train MLP	aug+	cos	epochs	ImageNet acc.	VOC detection AP_{50}	AP	AP_{75}
supervised					76.5	81.3	53.5	58.8
MoCo v1				200	60.6	81.5	55.9	62.6
(a)	✓			200	66.2	82.0	56.4	62.6
(b)		✓		200	63.4	82.2	56.8	63.2
(c)	✓	✓		200	67.3	**82.5**	57.2	63.9
(d)	✓	✓	✓	200	67.5	82.4	57.0	63.6
(e)	✓	✓	✓	**800**	**71.1**	**82.5**	**57.4**	**64.0**

Table 1. **Ablation of MoCo baselines**, evaluated by ResNet-50 for (i) ImageNet linear classification, and (ii) fine-tuning VOC object detection (mean of 5 trials). "**MLP**": with an MLP head; "**aug+**": with extra blur augmentation; "**cos**": cosine learning rate schedule.

Key takeaways:

- Non-linear projection head and strong data augmentation are crucial for contrastive learning.

Source: Chen et al., 2020

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

SimCLR vs. MoCo V1 vs. MoCo V2

case	unsup. pre-train MLP	aug+	cos	epochs	batch	ImageNet acc.
MoCo v1 [6]				200	256	60.6
SimCLR [2]	✓	✓	✓	200	256	61.9
SimCLR [2]	✓	✓	✓	200	8192	66.6
MoCo v2	✓	✓	✓	200	256	**67.5**
*results of **longer** unsupervised training follow:*						
SimCLR [2]	✓	✓	✓	1000	4096	69.3
MoCo v2	✓	✓	✓	800	256	**71.1**

Table 2. **MoCo *vs*. SimCLR**: ImageNet linear classifier accuracy (**ResNet-50, 1-crop 224×224**), trained on features from unsupervised pre-training. "aug+" in SimCLR includes blur and stronger color distortion. SimCLR ablations are from Fig. 9 in [2] (we thank the authors for providing the numerical results).

Key takeaways:

- Non-linear projection head and strong data augmentation are crucial for contrastive learning.
- Decoupling mini-batch size with negative sample size allows MoCo-V2 to outperform SimCLR with smaller batch size (256 vs. 8192).

Source: Chen et al., 2020

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

SimCLR vs. MoCo V1 vs. MoCo V2

mechanism	batch	memory / GPU	time / 200-ep.
MoCo	256	**5.0G**	**53 hrs**
end-to-end	256	7.4G	65 hrs
end-to-end	4096	93.0G†	n/a

Table 3. **Memory and time cost** in 8 V100 16G GPUs, implemented in PyTorch. †: based on our estimation.

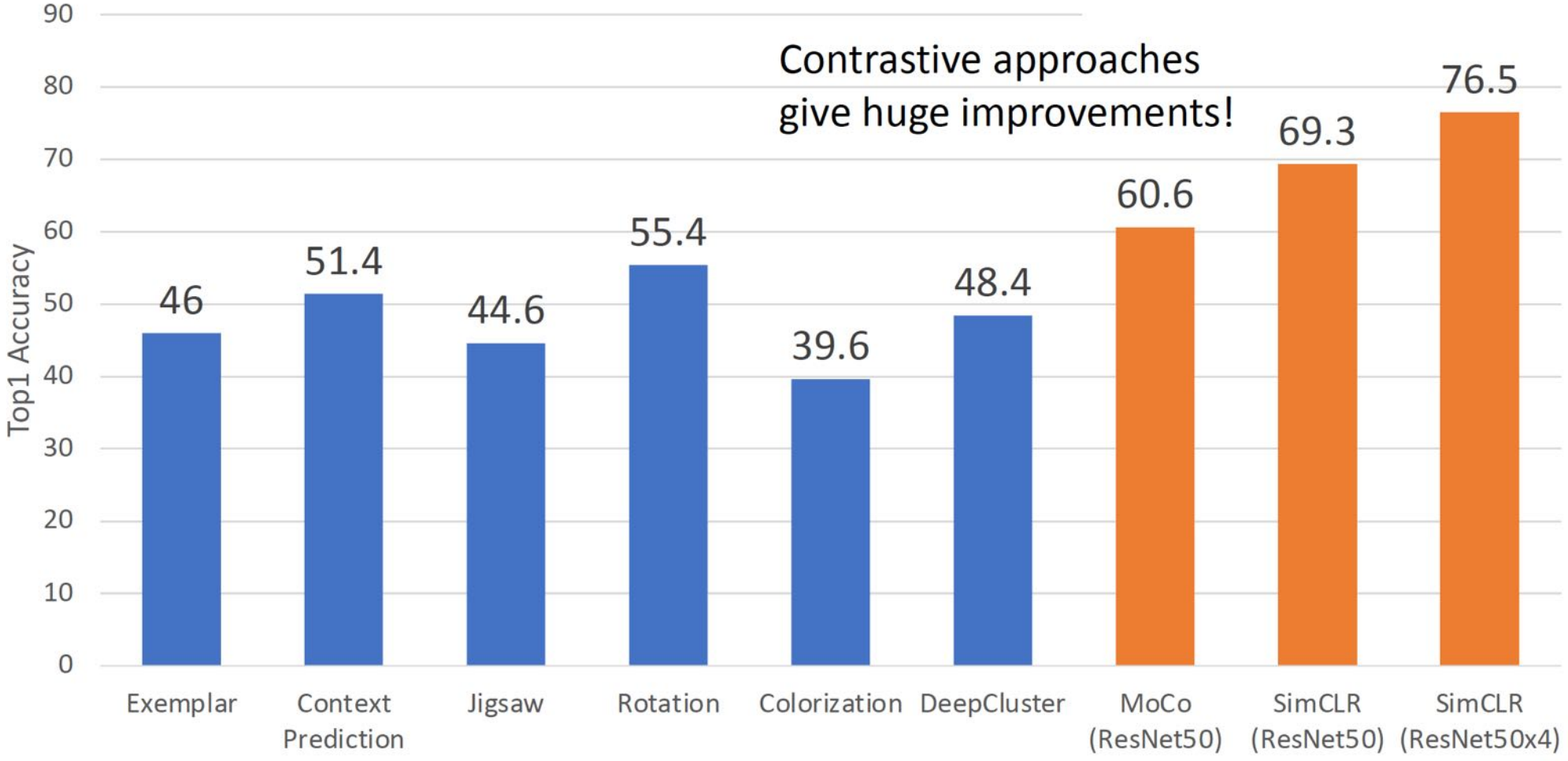
Key takeaways:

- Non-linear projection head and strong data augmentation are crucial for contrastive learning.
- Decoupling mini-batch size with negative sample size allows MoCo-V2 to outperform SimCLR with smaller batch size (256 vs. 8192).
- … all with much smaller memory footprint! ("end-to-end" means SimCLR here)

Source: Chen et al., 2020

slide credit: Justin Johnson

ImageNet Linear Classification from SSL Features

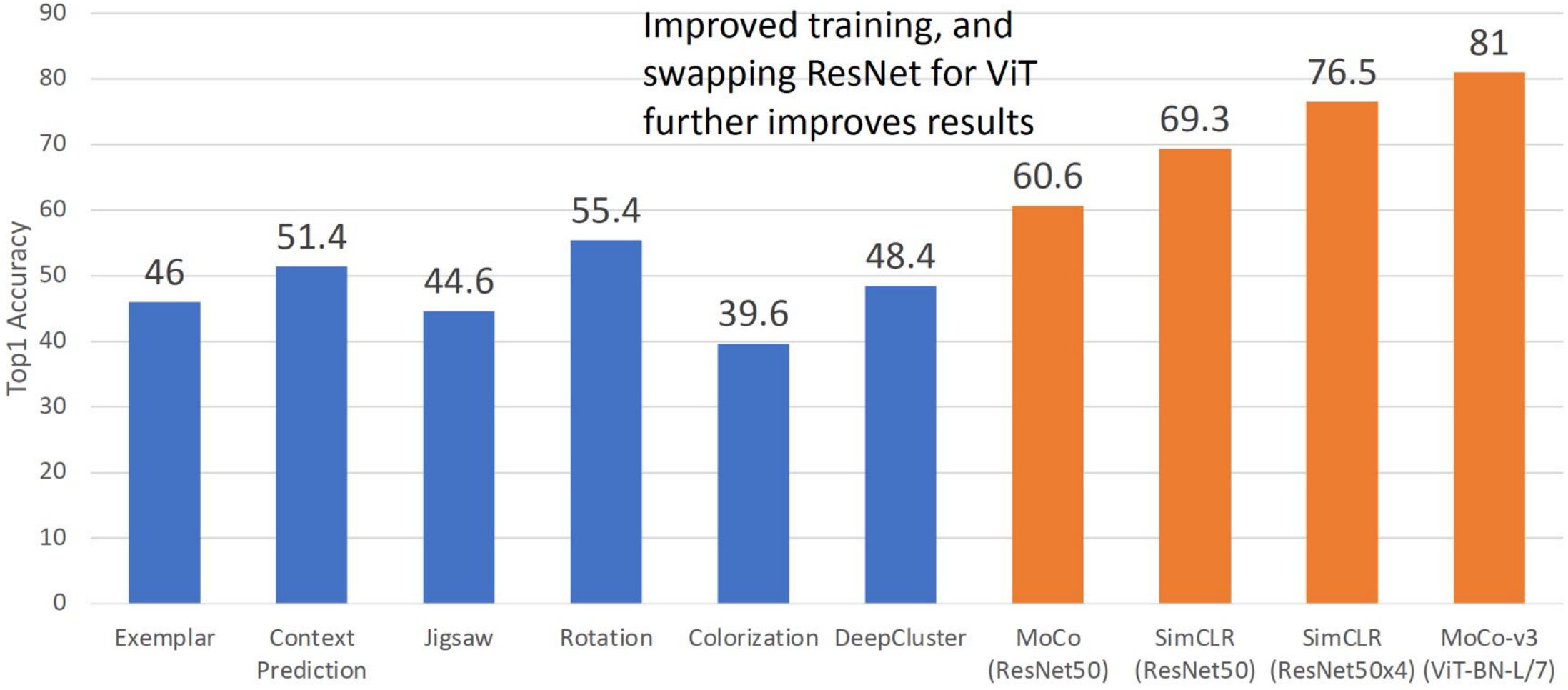


He et al, "Momentum Contrast for Unsupervised Visual Representation Learning", CVPR 2020
Chen et al, "A Simple Framework for Contrastive Learning of Visual Representations", ICML 2020
Chen et al, "An Empirical Study of Training Self-Supervised Vision Transformers", ICCV 2021

(Lots of caveats here … different architectures, etc)

slide credit: Justin Johnson

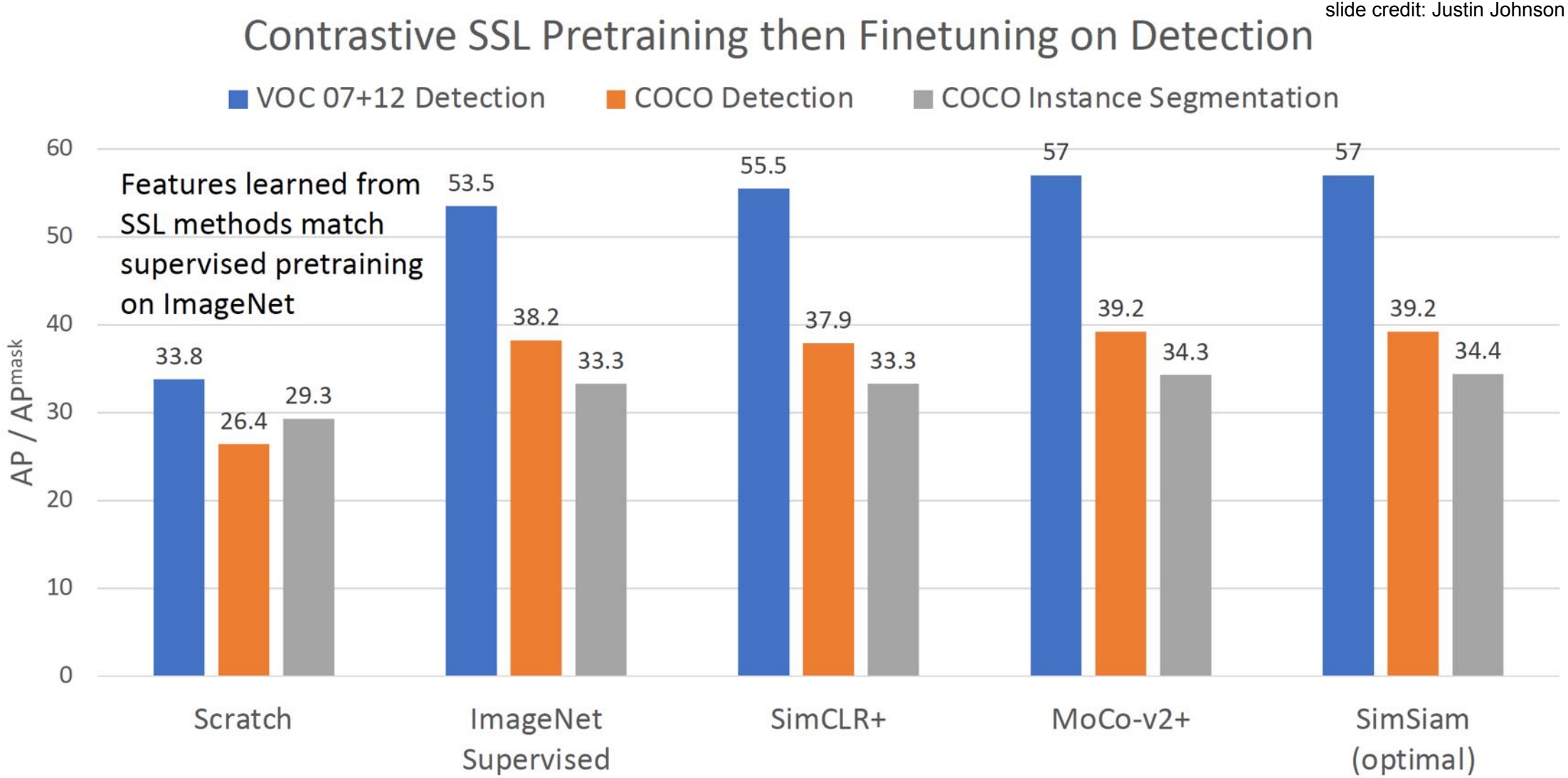
ImageNet Linear Classification from SSL Features



He et al, "Momentum Contrast for Unsupervised Visual Representation Learning", CVPR 2020
Chen et al, "A Simple Framework for Contrastive Learning of Visual Representations", ICML 2020
Chen et al, "An Empirical Study of Training Self-Supervised Vision Transformers", ICCV 2021

(Lots of caveats here … different architectures, etc)

slide credit: Justin Johnson

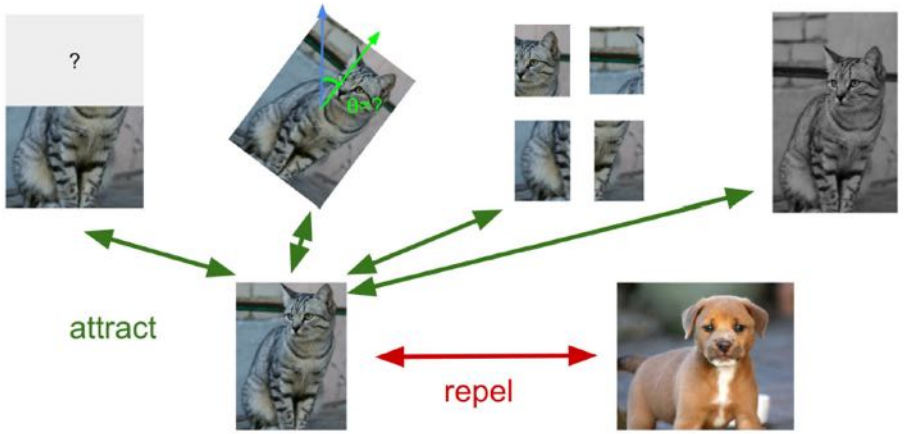


He et al, "Momentum Contrast for Unsupervised Visual Representation Learning", CVPR 2020
Chen et al, "A Simple Framework for Contrastive Learning of Visual Representations", ICML 2020

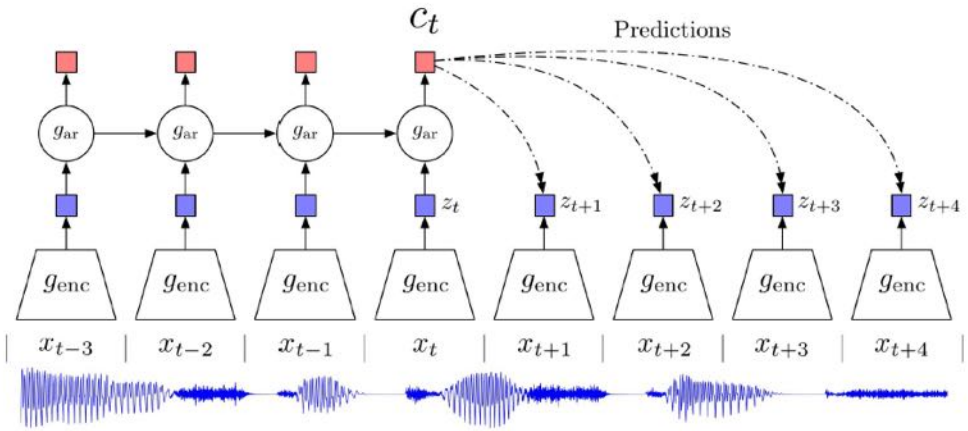
Chen et al, "Improved Baselines with Momentum Contrastive Learning", arXiv 2020
Chen and He, "Exploring simple Siamese representation learning", CVPR 2021

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Instance vs. Sequence Contrastive Learning



Instance-level contrastive learning:
contrastive learning based on
positive & negative instances.
Examples: SimCLR, MoCo

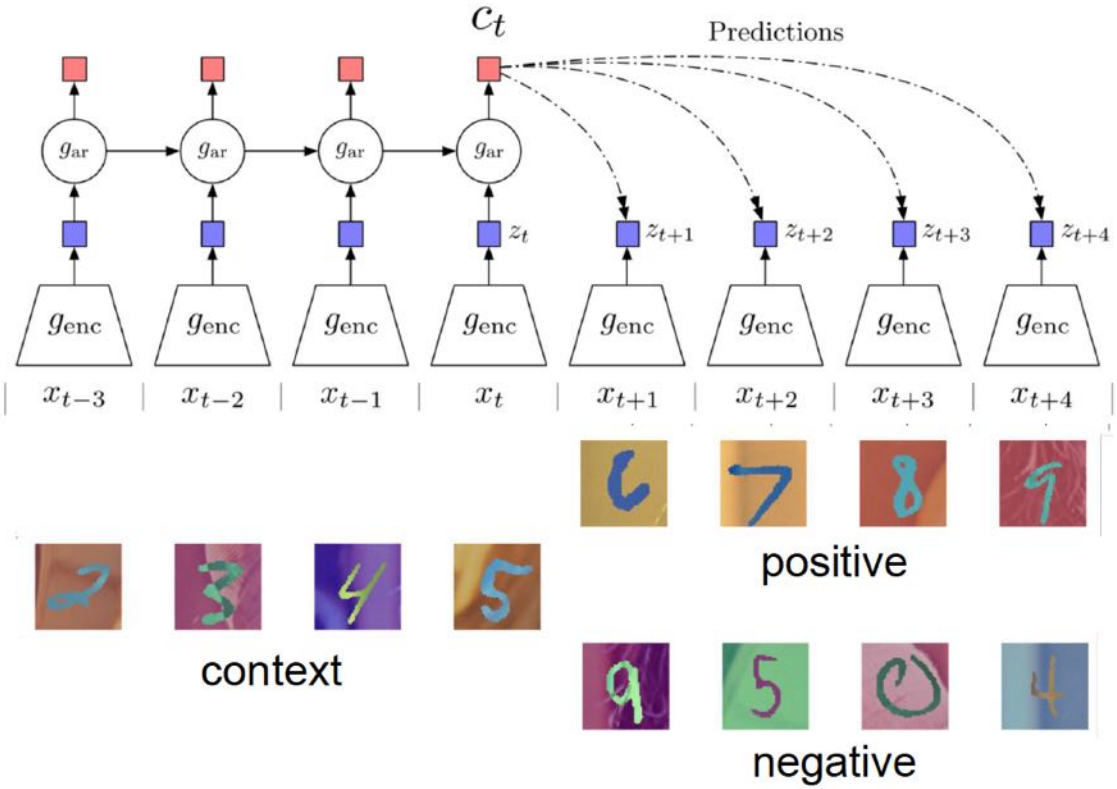


Source: van den Oord et al., 2018

Sequence-level contrastive learning:
contrastive learning based on
sequential / temporal orders.
Example: **Contrastive Predictive Coding (CPC)**

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Contrastive Predictive Coding (CPC)



Contrastive: contrast between "right" and "wrong" sequences using contrastive learning.

Predictive: the model has to predict future patterns given the current context.

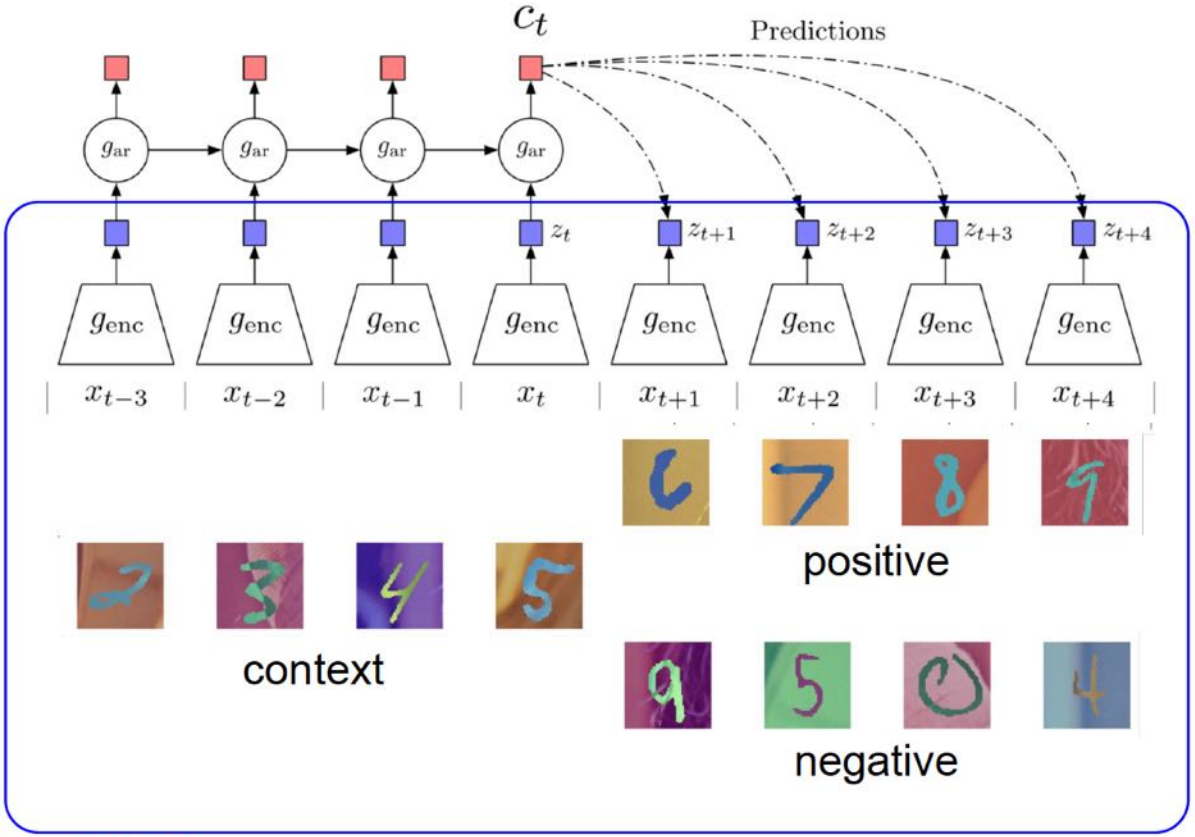
Coding: the model learns useful feature vectors, or "code", for downstream tasks, similar to other self-supervised methods.

Figure source

Source: van den Oord et al., 2018,

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Contrastive Predictive Coding (CPC)



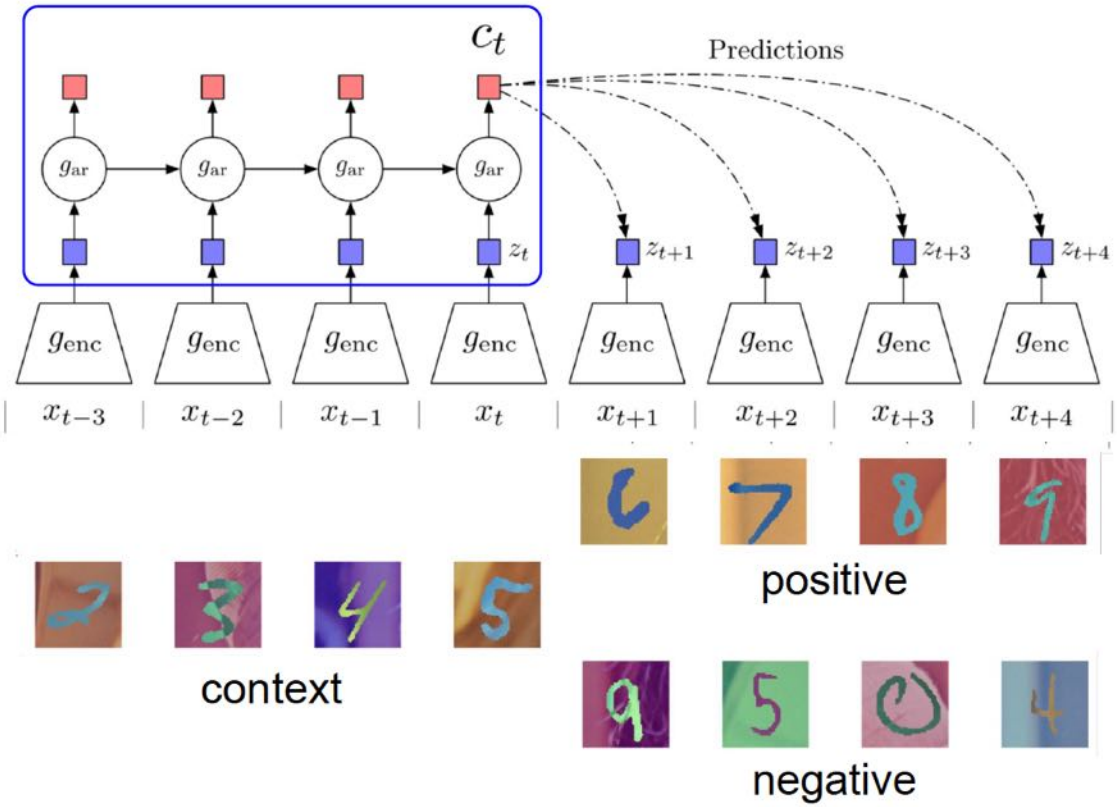
1. Encode all samples in a sequence into vectors $z_t = g_{enc}(x_t)$

Figure source

Source: van den Oord et al., 2018,

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Contrastive Predictive Coding (CPC)



1. Encode all samples in a sequence into vectors $z_t = g_{enc}(x_t)$

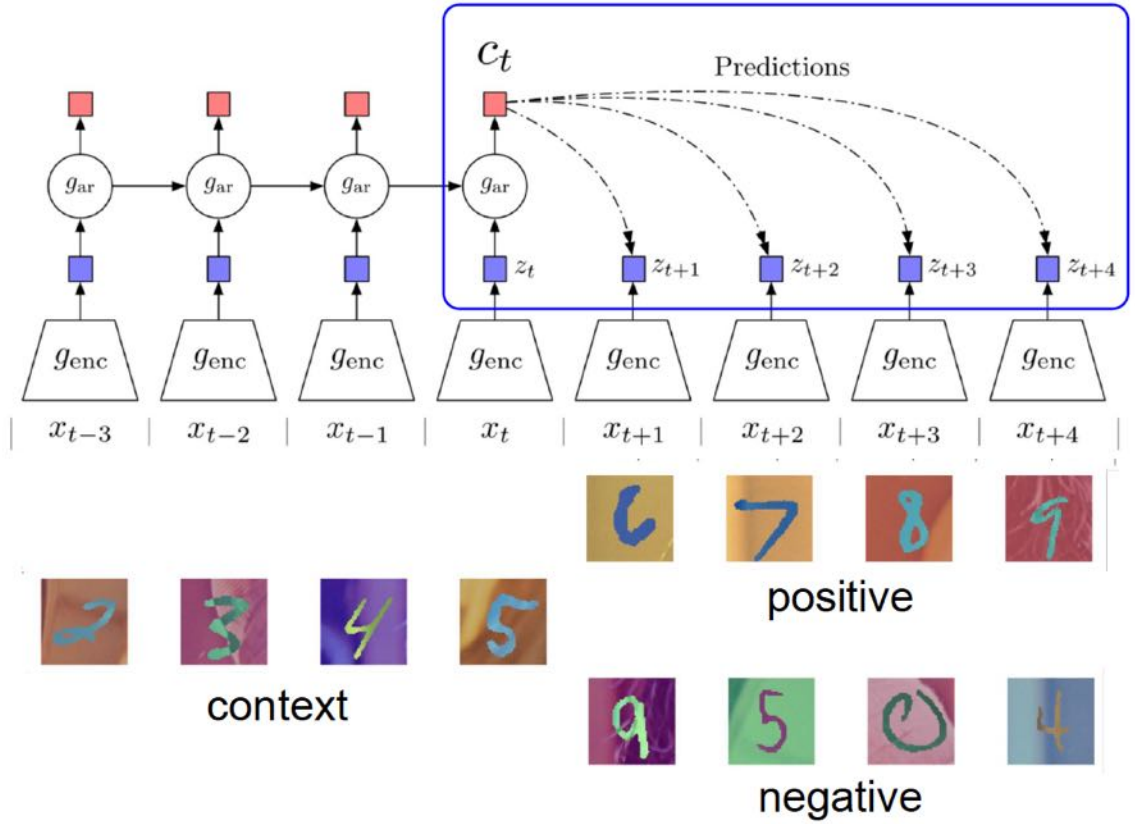
2. Summarize context (e.g., half of a sequence) into a context code c_t using an auto-regressive model (g_{ar}). The original paper uses GRU-RNN here.

Figure source

Source: van den Oord et al., 2018,

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Contrastive Predictive Coding (CPC)



1. Encode all samples in a sequence into vectors $z_t = g_{enc}(x_t)$

2. Summarize context (e.g., half of a sequence) into a context code c_t using an auto-regressive model (g_{ar})

3. Compute InfoNCE loss between the context c_t and future code z_{t+k} using the following time-dependent score function:

$$s_k(z_{t+k}, c_t) = z_{t+k}^T W_k c_t$$

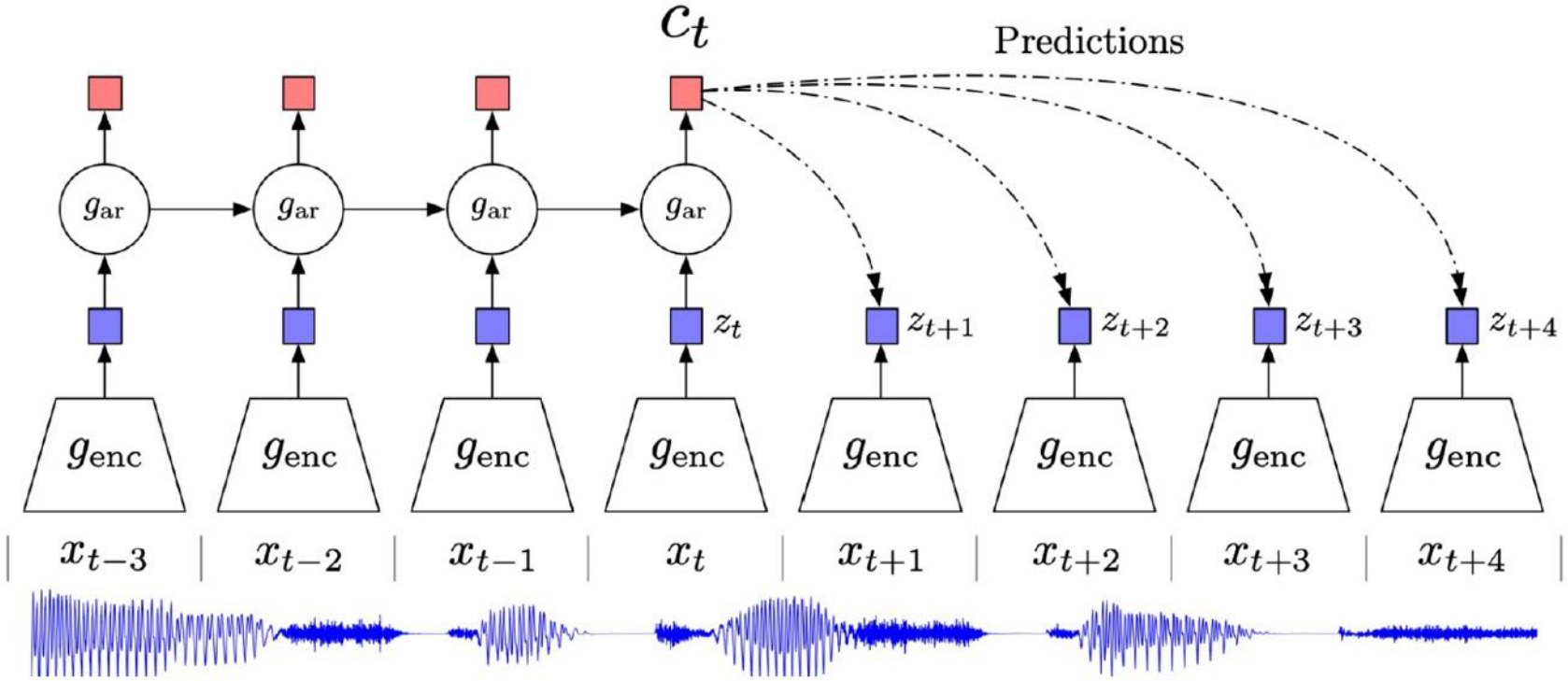
, where W_k is a trainable matrix.

Figure source

Source: van den Oord et al., 2018,

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

CPC Example: Modeling Audio Sequences



Source: van den Oord et al., 2018,

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

CPC Example: Modeling Audio Sequences

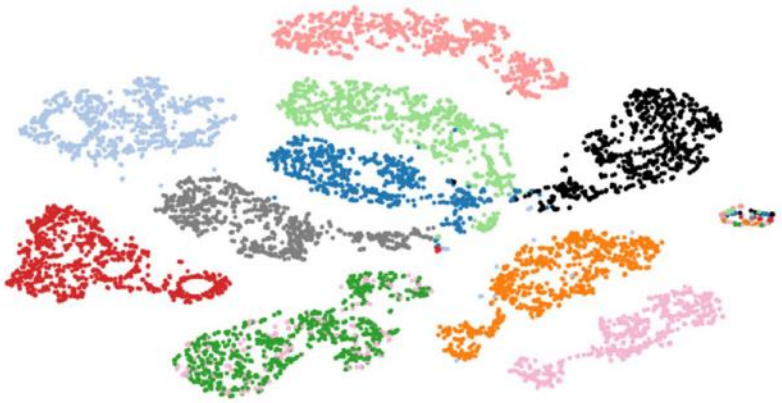


Figure 2: t-SNE visualization of audio (speech) representations for a subset of 10 speakers (out of 251). Every color represents a different speaker.

Method	ACC
Phone classification	
Random initialization	27.6
MFCC features	39.7
CPC	64.6
Supervised	74.6
Speaker classification	
Random initialization	1.87
MFCC features	17.6
CPC	97.4
Supervised	98.5

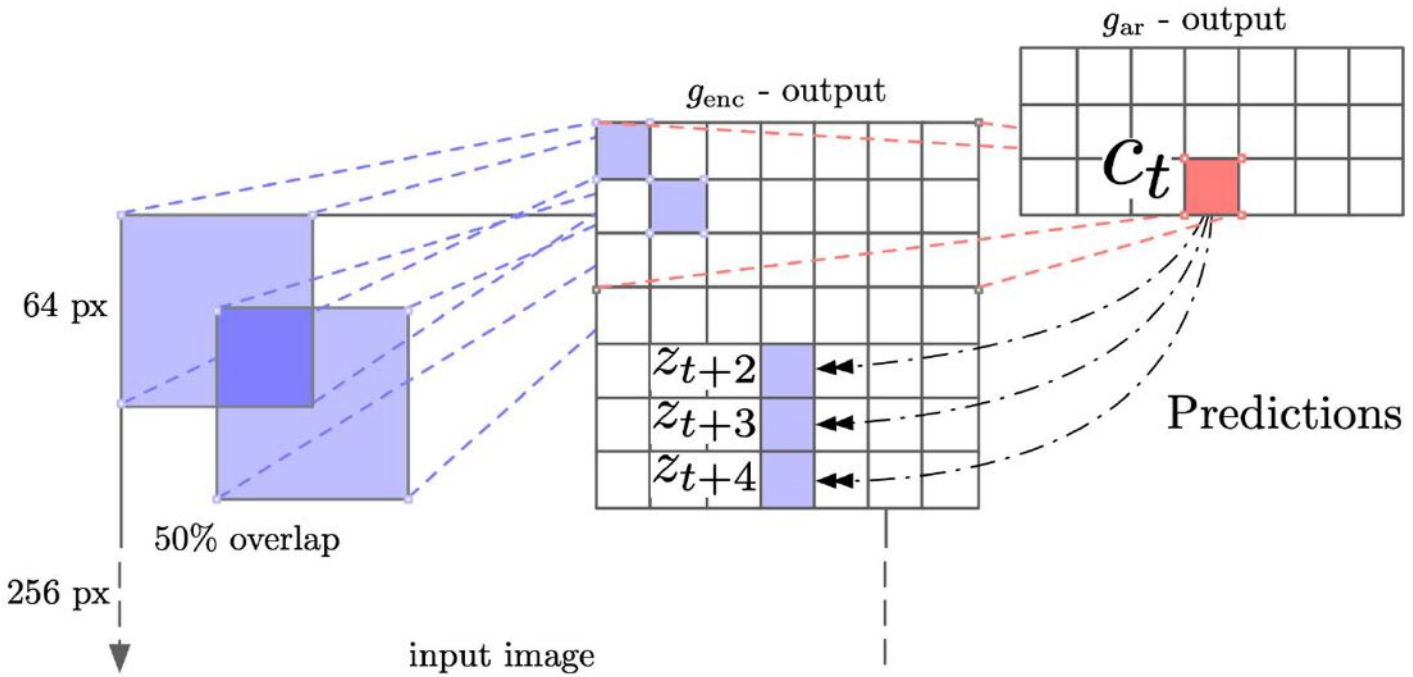
Linear classification on trained representations (LibriSpeech dataset)

Source: van den Oord et al., 2018,

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

CPC Example: Modeling Visual Context

Idea: split image into patches, model rows of patches from top to bottom as a sequence. I.e., use top rows as context to predict bottom rows.



Source: van den Oord et al., 2018,

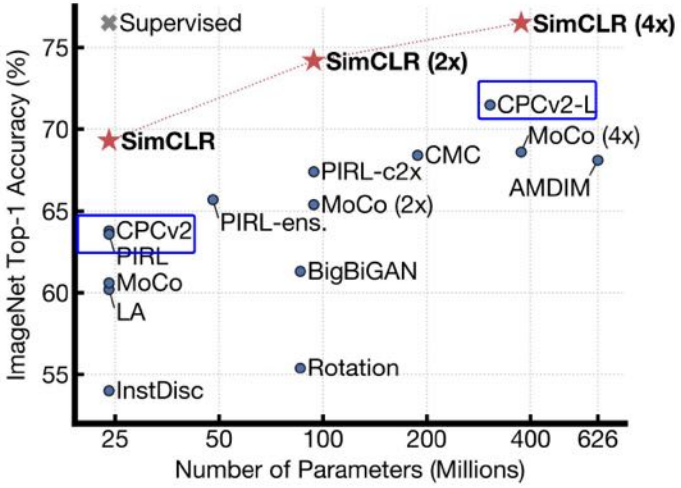
slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

CPC Example: Modeling Visual Context

Method	Top-1 ACC
Using AlexNet conv5	
Video [28]	29.8
Relative Position [11]	30.4
BiGan [35]	34.8
Colorization [10]	35.2
Jigsaw [29] *	38.1
Using ResNet-V2	
Motion Segmentation [36]	27.6
Exemplar [36]	31.5
Relative Position [36]	36.2
Colorization [36]	39.6
CPC	**48.7**

Table 3: ImageNet top-1 unsupervised classification results. *Jigsaw is not directly comparable to the other AlexNet results because of architectural differences.

- Compares favorably with other pretext task-based self-supervised learning method.
- Doesn't do as well compared to newer instance-based contrastive learning methods on image feature learning.



Source: van den Oord et al., 2018,

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Summary: Contrastive Representation Learning

A general formulation for contrastive learning:

$$\text{score}(f(x), f(x^+)) >> \text{score}(f(x), f(x^-))$$

InfoNCE loss: N-way classification among positive and negative samples

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+))}{\exp(s(f(x), f(x^+)) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))} \right]$$

Commonly known as the InfoNCE loss (van den Oord et al., 2018)

A *lower bound* on the mutual information between *f(x)* and *f(x⁺)*

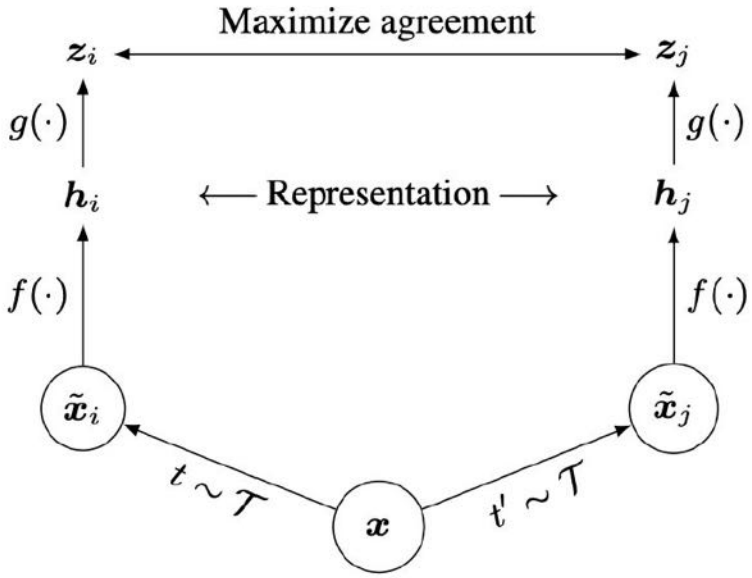
$$MI[f(x), f(x^+)] \geq \log(N) - \mathcal{L}$$

slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Summary: Contrastive Representation Learning

SimCLR: a simple framework for contrastive representation learning

- **Key ideas**: non-linear projection head to allow flexible representation learning
- Simple to implement, effective in learning visual representation
- Requires large training batch size to be effective; large memory footprint

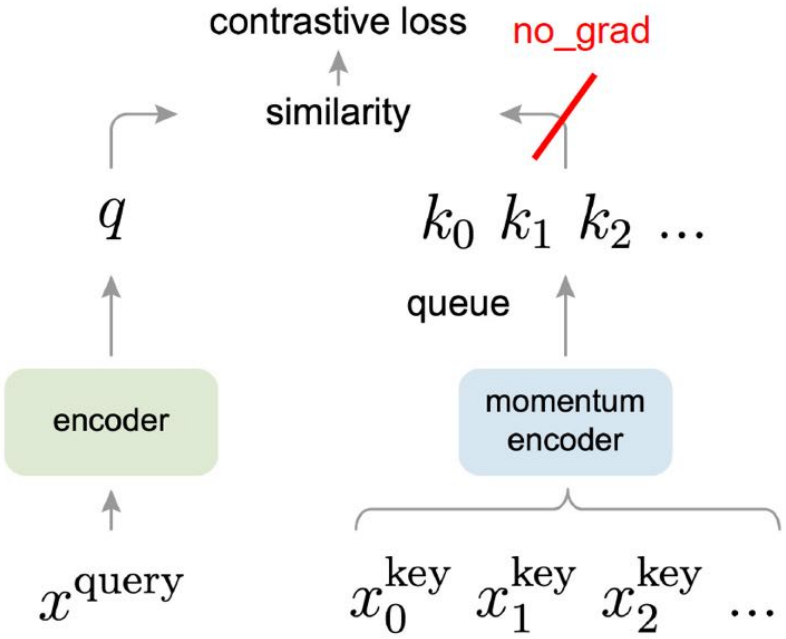


slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Summary: Contrastive Representation Learning

MoCo (v1, v2): contrastive learning using momentum sample encoder

- Decouples negative sample size from minibatch size; allows large batch training without TPU
- MoCo-v2 combines the key ideas from SimCLR, i.e., nonlinear projection head, strong data augmentation, with momentum contrastive learning

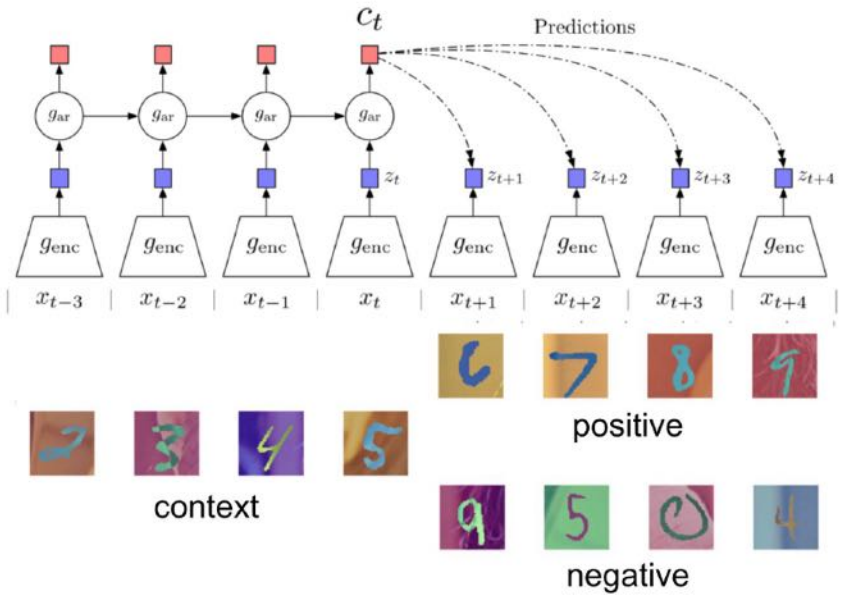


slide credit: Fei-Fei Li, Jiajun Wu, Ruohan Gao

Summary: Contrastive Representation Learning

CPC: sequence-level contrastive learning

- Contrast “right” sequence with “wrong” sequence.
- InfoNCE loss with a time-dependent score function.
- Can be applied to a variety of learning problems, but not as effective in learning image representations compared to instance-level methods.



Overview of Today's Lecture

- U-Net Architecture for Semantic Segmentation

- Self-Supervised Learning - Part 1 (today):
  - Motivation of Self-Supervised Learning
    - setting of pretext tasks — how to evaluate
  - Pretext tasks from image transformations
    - rotation, inpainting, rearrangement, coloring
  - Contrastive representation learning
    - intuition and formulation
    - instance contrastive learning: SimCLR & MOCO
    - sequence contrastive learning: CPC

- Self-Supervised Learning - Part 2 (next time):
  - Teacher-Student "feature reconstruction"
    - motivation, setting
    - methods: BYOL, DINO