Self-Supervised Learning and Foundation Models

Mahmoud ALI

Francois Bremond

18/03/2025

INRIA

Inria STARS

Deep Learning for Computer Vision

Winter 2025

[Home | Schedule | Final Project]

Course Overview

This course studies Computer Vision (CV) algorithms together with their visual representations learnt through Deep Learning (DL) techniques. The studied algorithms are intended to solve traditional CV tasks, including classification, object detection and tracking, retrieval, face detection, image/video generation, emotion and action recognition and are illustrated through a panel of applications, such as video retrieval from the web, visual-surveillance, autonomous driving, merchandising, assisted living and robotics. The course discusses state-of-the-art methods from low-level description to high-level representation, and their dependence on the related CV tasks. The focus of the course is on recent, state of the art methods and large scale applications. Cutting-edge topics will be studied, such as Convolutional Neural Networks, Recurrent Neural Networks and Generative Adversarial Networks. You will learn also to build projects in PyTorch/TensorFlow.

Announcements

January 14, 2025: Welcome to Deep Learning for Computer Vision !

Course Information

Course Instructors

Francois Bremond	Tomasz Stanczyk	Snehashis Majhi	Seongro Yoon	Mahmoud Ali

https://www-sop.inria.fr/members/Francois.Bremond/MSclass/deepLearningWinterSchool25/UCA_master/index.html

3iA Côte d'Azur
Interdisciplinary Institute for Artificial Intelligence

Inria STARS

Deep Learning for Computer Vision

Inria

Winter 2025

[Home | Schedule | Final Project]

Lecture	Date	Topic	Instructor	Course Materials	TP/TD
Week 1					
1	Tue 14/01/2025	Introduction. Image Classification. PyTorch Basics.	Francois, Tomasz	slides F. (pdf) video F (mp4)	TP1
Week 2					
2	Tue 21/01/2025	Object Detection	Tomasz	slides T. (pdf) video T. (mp4)	TP2
Week 3					
3	Tue 28/01/2025	Object Tracking	Tomasz	slides T. (pdf) video Tomasz (mp4)	TP3
Week 4					
4	Tue 04/02/2025	Video and Action Classification	Snehashis	slides Snehashis (pdf) video Snehashis (mp4)	TP4.1 TP4.2
Week 5					
5	Tue 04/03/2025	Action Detection and Anticipation	Snehashis	slides Snehashis (pdf) video (mp4)	TP5
Week 6					
6	Tue 11/03/2025	Image and Video Generation	Seongro	slides Seongro (pdf)	
Week 7					
7	Tue 18/03/2025	Foundation Models	Mahmoud		
Week 8					
8	Tue 25/03/2025	Final Project Presentations			

https://www-sop.inria.fr/members/Francois.Bremond/MSclass/deepLearningWinterSchool25/UCA_master/schedule.html

Table of Contents

Table of Contents

Problem

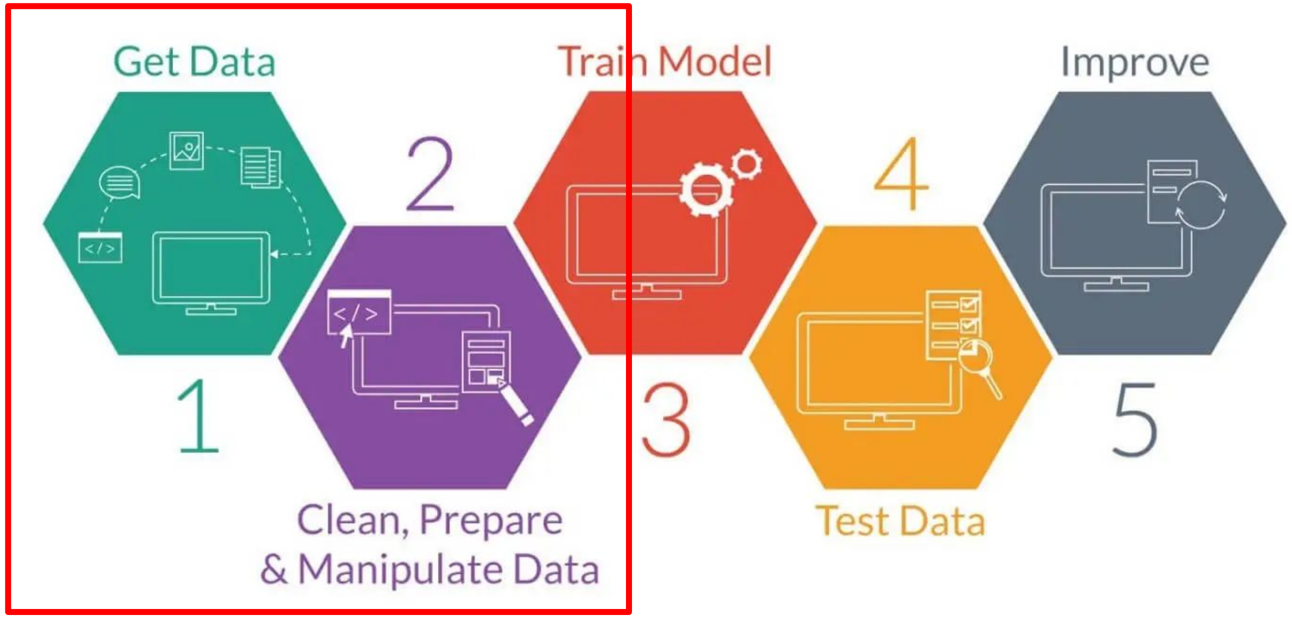
1. **Question:** How can you detect an object, such as a "car," in an image using object detection techniques?

Key Steps to build Model [A-Z]:

1. **Collect Data** – Gather images containing objects of interest.
2. **Label the Data** – Annotate images with bounding boxes or masks.
3. **Train the Model** – Use deep learning techniques (e.g., CNNs) to learn object features.
4. **Test the Model** – Validate your trained model.

Problem

1. **Question:** How can you detect an object, such as a "car," in an image using object detection techniques?



1. High costs, Time consuming, Annotation errors and scalability issues

Problem: Supervised Learning is Expensive!

Assume you want to label 1M images. How much will it cost?

Problem: Supervised Learning is Expensive!

Assume you want to label 1M images. How much will it cost?

(1,000,000 images)	(Small to medium sized dataset)
× (10 seconds/image)	(Fast annotation)
× (1/3600 hours/second)	
× ($15 / hour)	(Low wage paid to annotator)

Problem: Supervised Learning is Expensive!

Assume you want to label 1M images. How much will it cost?

(1,000,000 images) (Small to medium sized dataset)

$\times$ (10 seconds/image) (Fast annotation)

$\times$ (1/3600 hours/second)

$\times$ (\$15 / hour) (Low wage paid to annotator)

= \$41,667

(Other assumptions: one annotator per image, no benefits / payroll tax / crowdsourcing fee for annotators; not accounting for time to set up tasks for annotators, etc. Real costs could easily be 3x this or more)

Problem: Supervised Learning is Expensive!

Assume you want to label 1B images. How much will it cost?

(1,000,000,000 images)

$\times$ (10 seconds/image)

$\times$ (1/3600 hours/second)

$\times$ ($15 / hour)

= $41,666,667

(Large dataset)

(Fast annotation)

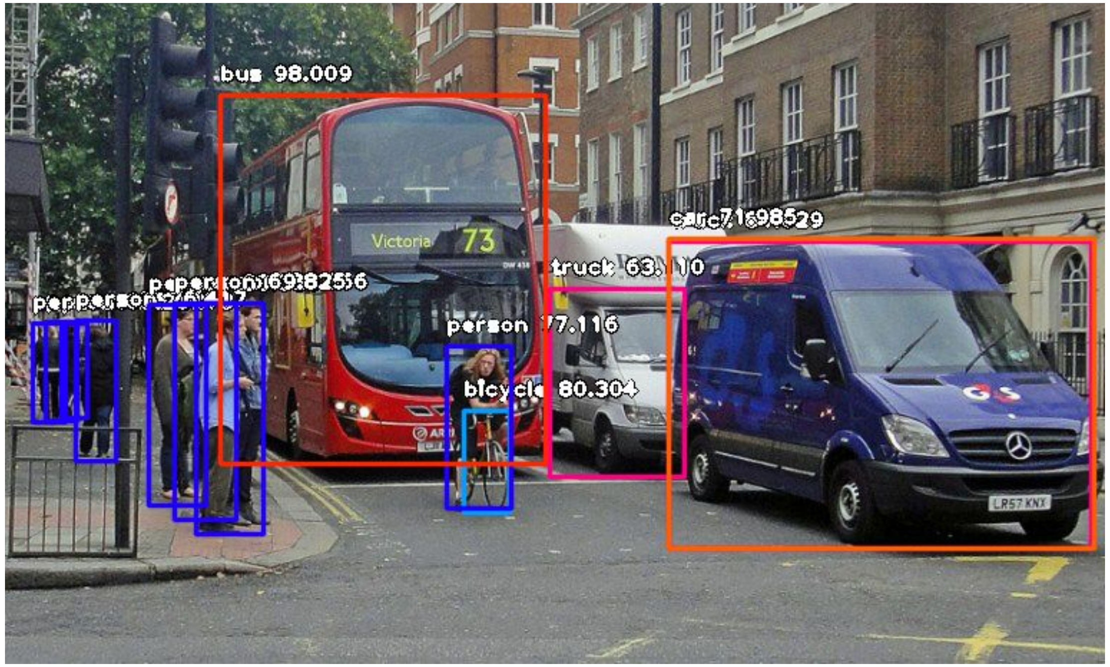
(Low wage paid to annotator)

(Other assumptions: one annotator per image, no benefits / payroll tax / crowdsourcing fee for annotators; not accounting for time to set up tasks for annotators, etc. Real costs could easily be 3x this or more)

Problem

2. Question: How would you detect other objects, such as "person" ?

- What steps would you follow?
- Can the same model be used for both objects?



2. Generalization Challenge

Problem: Supervised Learning is Not How We Learn

Babies don't get supervision for everything they see!

[Baby image](#) is [CC0 public domain](#)

Solution: Self-Supervised Learning

Lets build methods that learn from ”raw” data – :
1. No annotations required.
2. Can Generalize well.

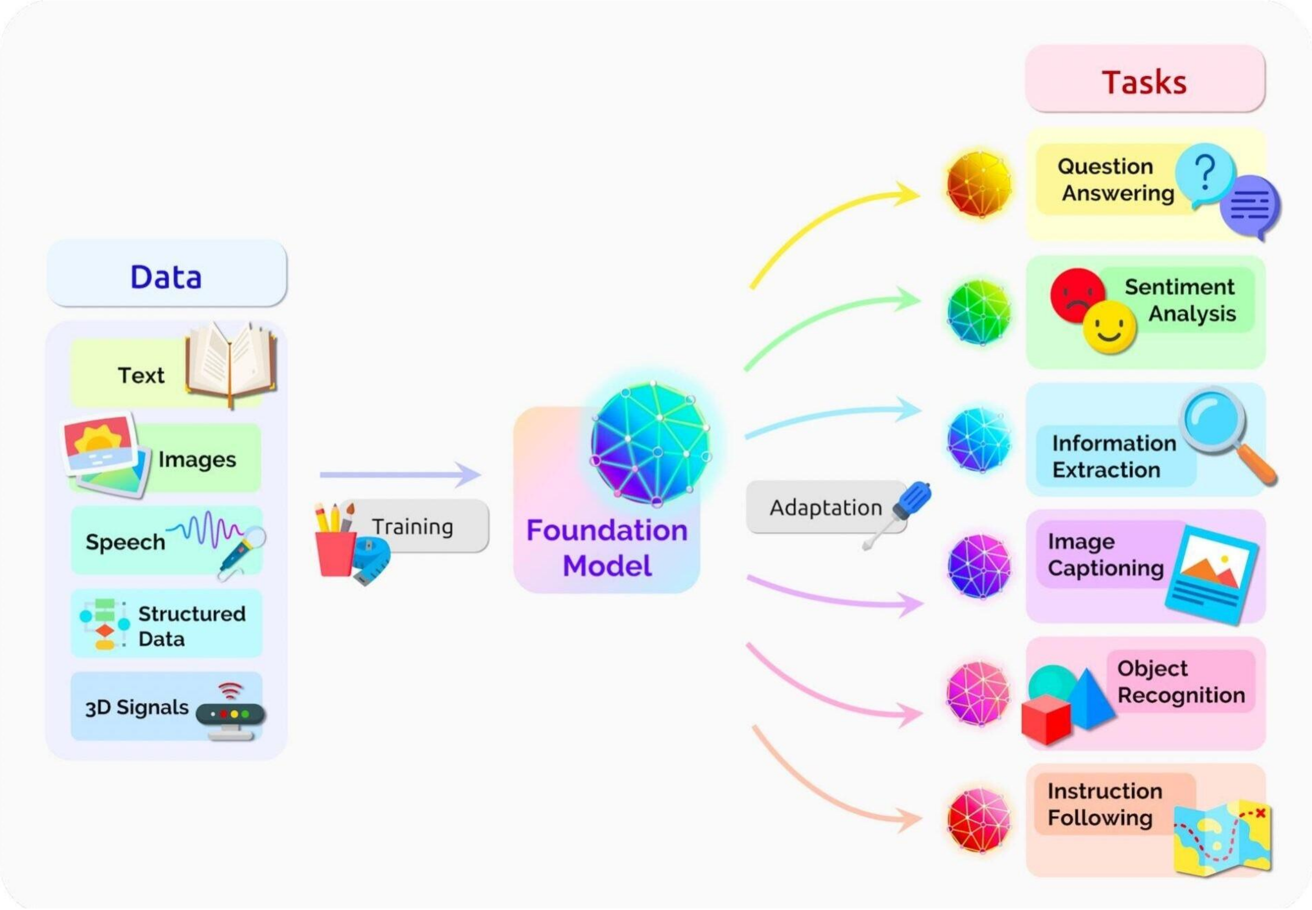
Unsupervised Learning: Model isn’t told what to predict. Older terminology, not used as much today.

Self-Supervised Learning: Model is trained to predict some naturally- occurring signal in the raw data rather than human annotations.

Table of Contents

Introduction

- **Foundation Models for Vision**
  - Is a pre-trained deep neural network that forms the backbone for various downstream tasks.
  - Fixing a foundation model (e.g., trained via self-supervised learning) and only adapting a simple task-specific model is sufficient for many problems



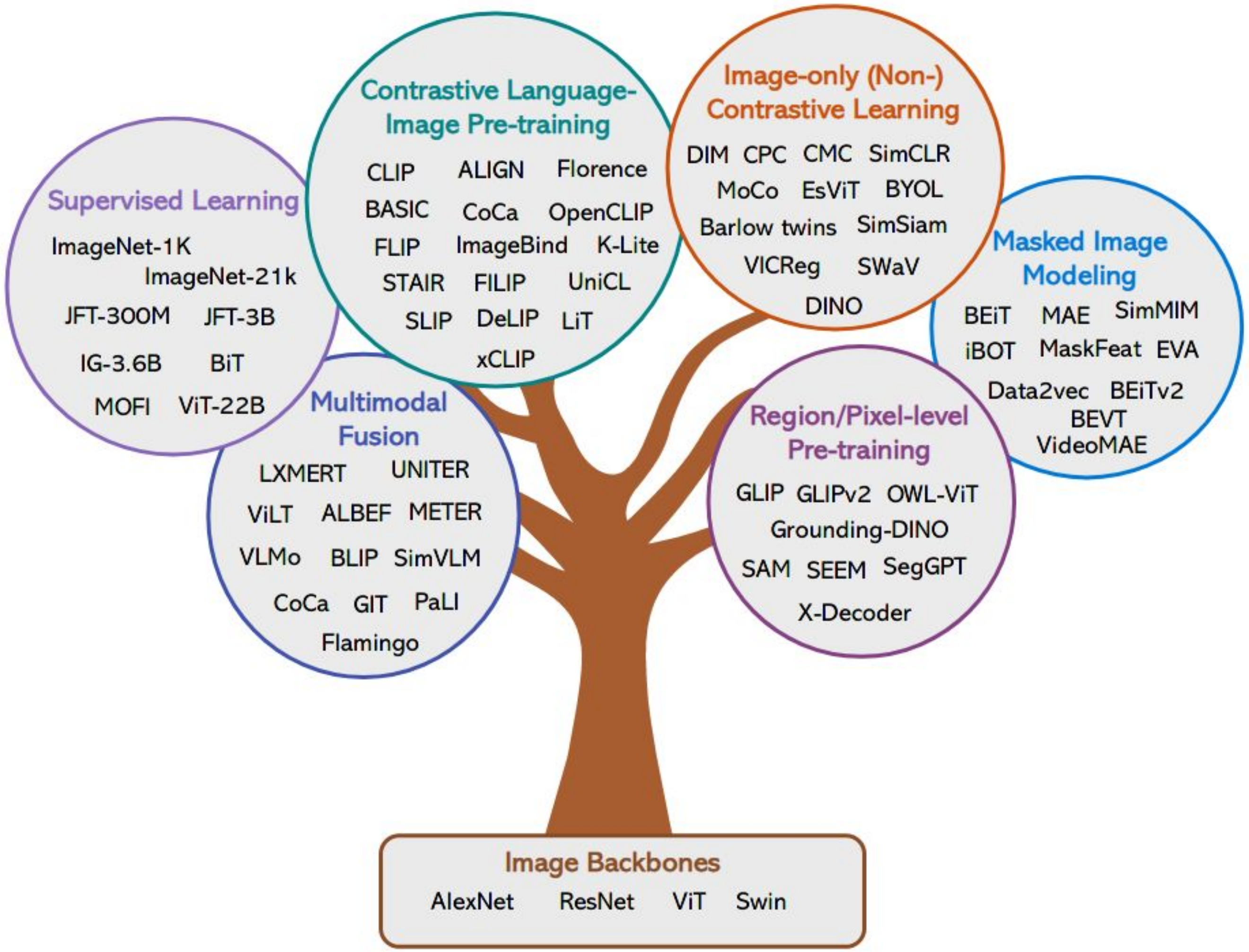
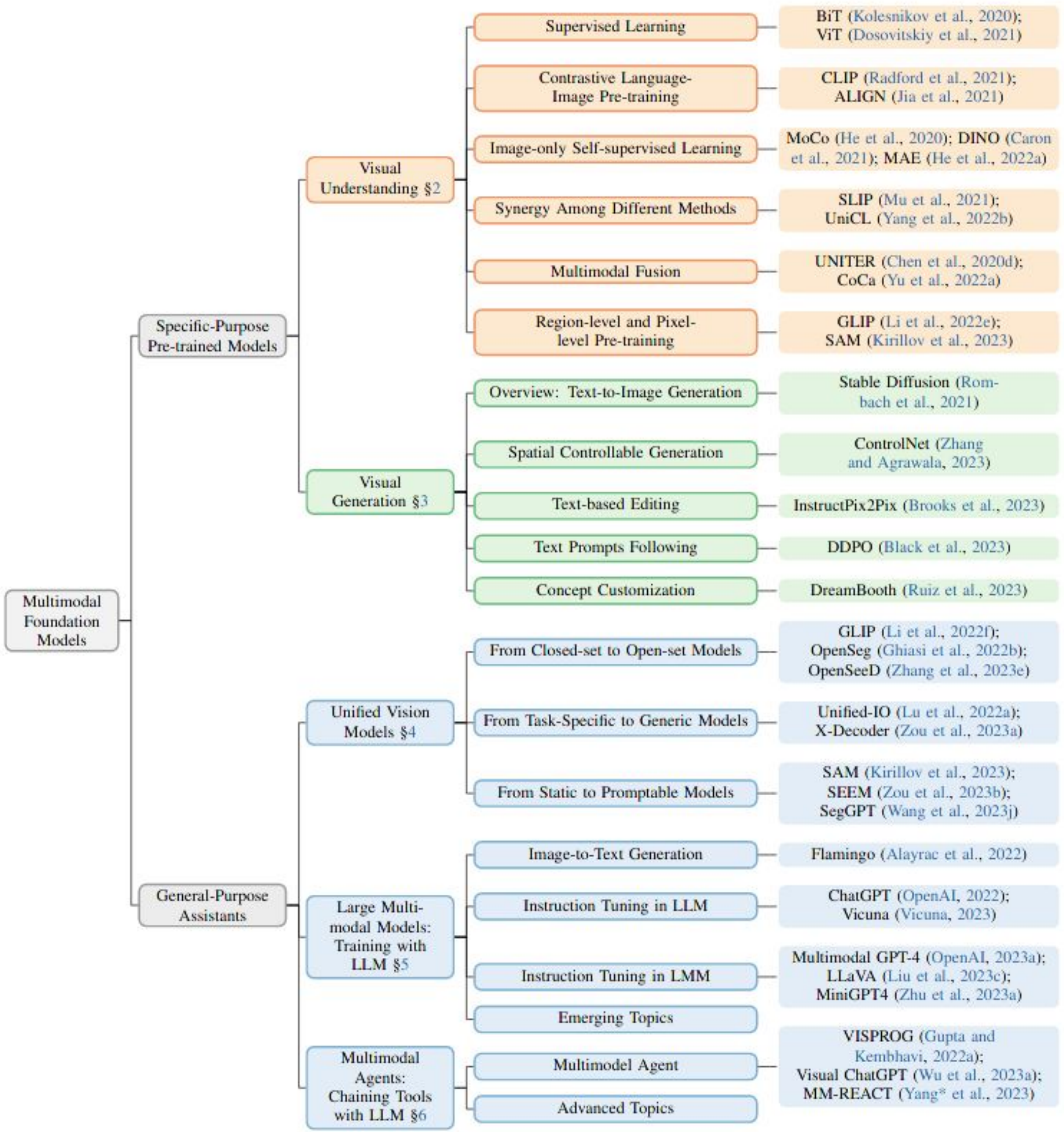


Figure 2.3: An overview of the topics covered in this chapter and representative works in each topic. We start from supervised learning and CLIP, and then move on to image-only self-supervised learning, including contrastive learning, non-contrastive learning, and masked image modeling. Lastly, we discuss pre-training methods that empower multimodal fusion, region-level and pixel-level image understanding.


Multimodal Foundation Models
Specific-Purpose Pre-trained Models
Visual Understanding §2
Supervised Learning
BiT (Kolesnikov et al., 2020); ViT (Dosovitskiy et al., 2021)
Contrastive Language-Image Pre-training
CLIP (Radford et al., 2021); ALIGN (Jia et al., 2021)
Image-only Self-supervised Learning
MoCo (He et al., 2020); DINO (Caron et al., 2021); MAE (He et al., 2022a)
Synergy Among Different Methods
SLIP (Mu et al., 2021); UniCL (Yang et al., 2022b)
Multimodal Fusion
UNITER (Chen et al., 2020d); CoCa (Yu et al., 2022a)
Region-level and Pixel-level Pre-training
GLIP (Li et al., 2022e); SAM (Kirillov et al., 2023)
Visual Generation §3
Overview: Text-to-Image Generation
Stable Diffusion (Rombach et al., 2021)
Spatial Controllable Generation
ControlNet (Zhang and Agrawala, 2023)
Text-based Editing
InstructPix2Pix (Brooks et al., 2023)
Text Prompts Following
DDPO (Black et al., 2023)
Concept Customization
DreamBooth (Ruiz et al., 2023)
General-Purpose Assistants
Unified Vision Models §4
From Closed-set to Open-set Models
GLIP (Li et al., 2022f); OpenSeg (Ghiasi et al., 2022b); OpenSeeD (Zhang et al., 2023e)
From Task-Specific to Generic Models
Unified-IO (Lu et al., 2022a); X-Decoder (Zou et al., 2023a)
From Static to Promptable Models
SAM (Kirillov et al., 2023); SEEM (Zou et al., 2023b); SegGPT (Wang et al., 2023j)
Large Multimodal Models: Training with LLM §5
Image-to-Text Generation
Flamingo (Alayrac et al., 2022)
Instruction Tuning in LLM
ChatGPT (OpenAI, 2022); Vicuna (Vicuna, 2023)
Instruction Tuning in LMM
Multimodal GPT-4 (OpenAI, 2023a); LLaVA (Liu et al., 2023c); MiniGPT4 (Zhu et al., 2023a)
Emerging Topics
Multimodal Agents: Chaining Tools with LLM §6
Multimodel Agent
VISPROG (Gupta and Kembhavi, 2022a); Visual ChatGPT (Wu et al., 2023a); MM-REACT (Yang* et al., 2023)
Advanced Topics

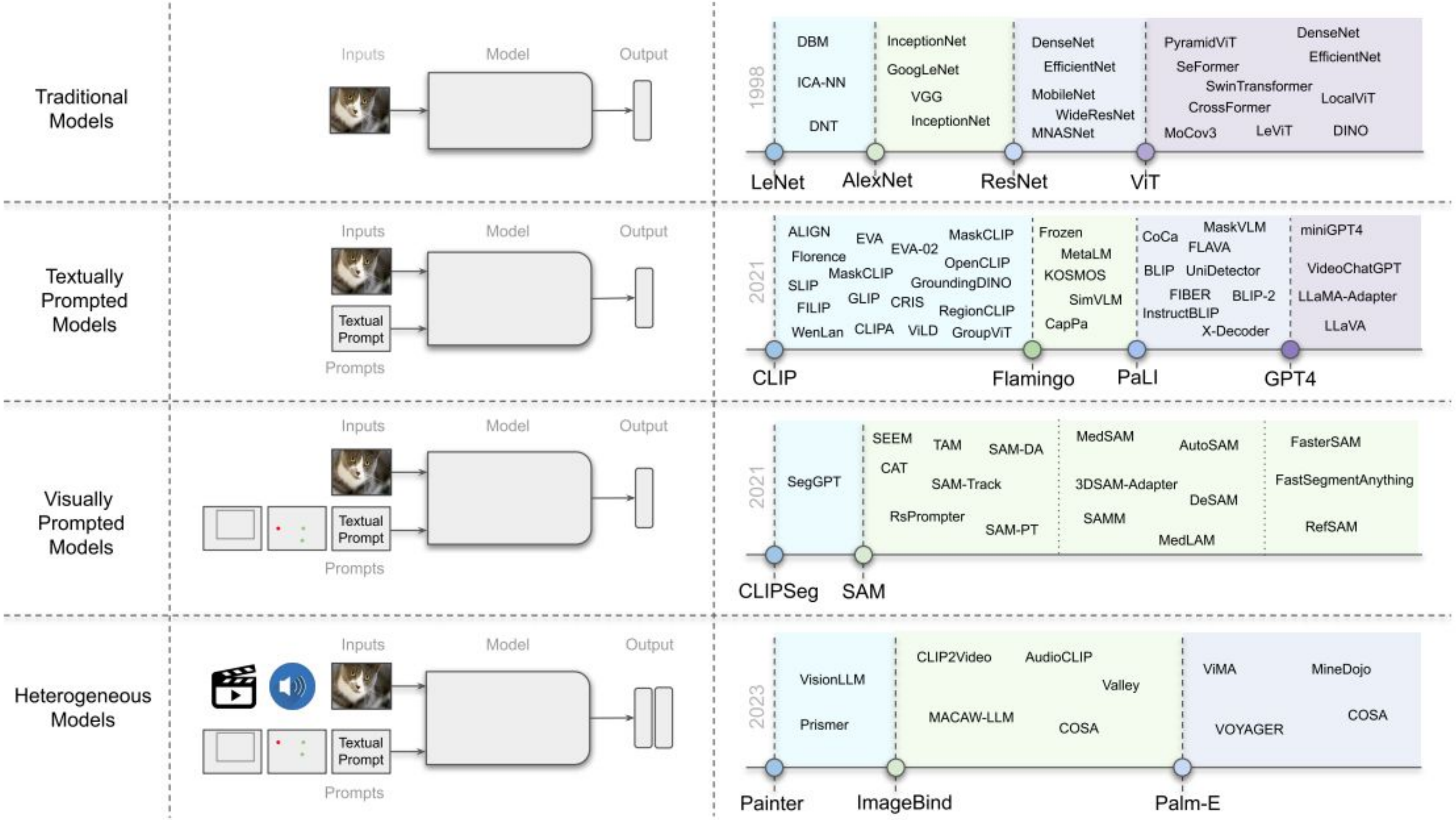
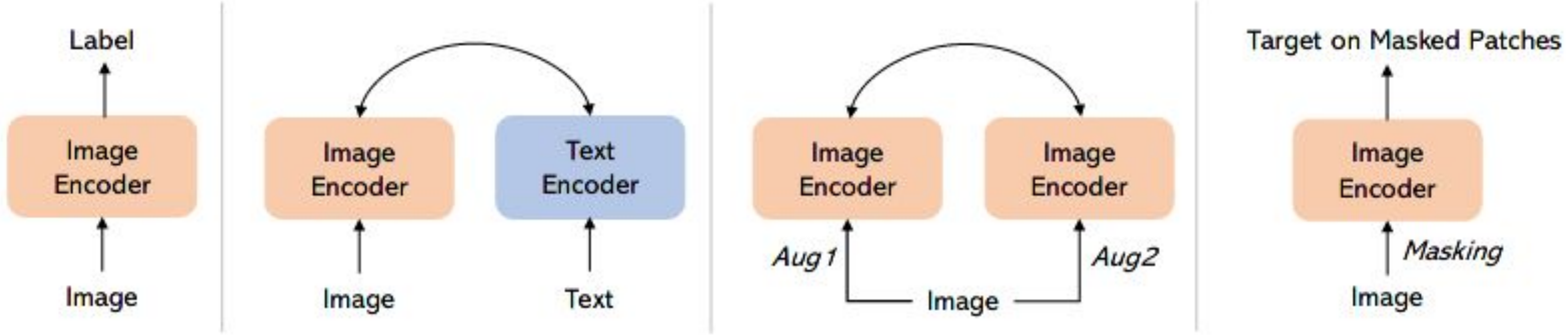


Fig. 1. Overview of the evolution of foundational models in computer vision. (left) We show the progression of models in computer vision. (right) We show the evolution of these models with major milestones reported in the literature shown with dotted lines.

https://github.com/awaisrauf/Awesome-CV-Foundational-Models



(a) Supervised Learning (b) CLIP (c) Image-only (non-)contrastive learning (d) Masked image modeling

Figure 2.2: A high-level overview of different approaches to learn general image representations, including supervised learning (Krizhevsky et al., 2012), contrastive language-image pre-training (Radford et al., 2021; Jia et al., 2021), and image-only self-supervised learning, including contrastive learning (Chen et al., 2020a; He et al., 2020), non-contrastive learning (Grill et al., 2020; Chen and He, 2021), and masked image modeling (Bao et al., 2022; He et al., 2022a).

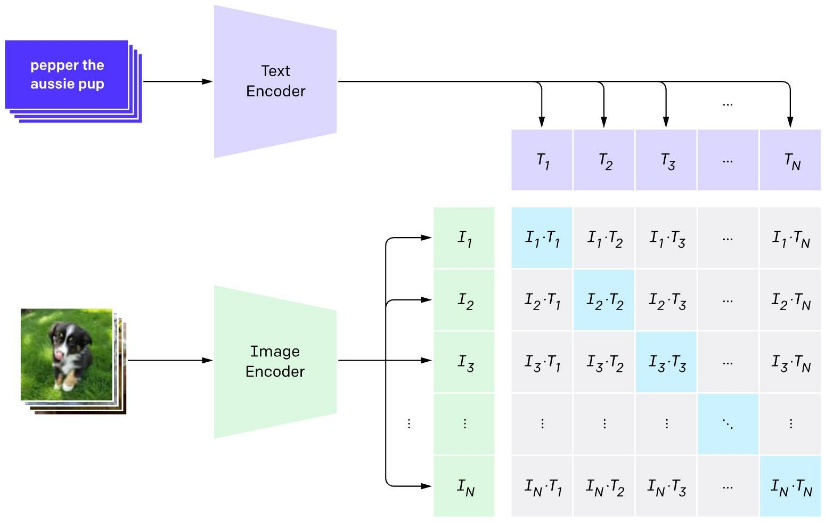
Self-Supervised Representation Learning

- Scalable: train huge models on unlimited data and not worry about overfitting

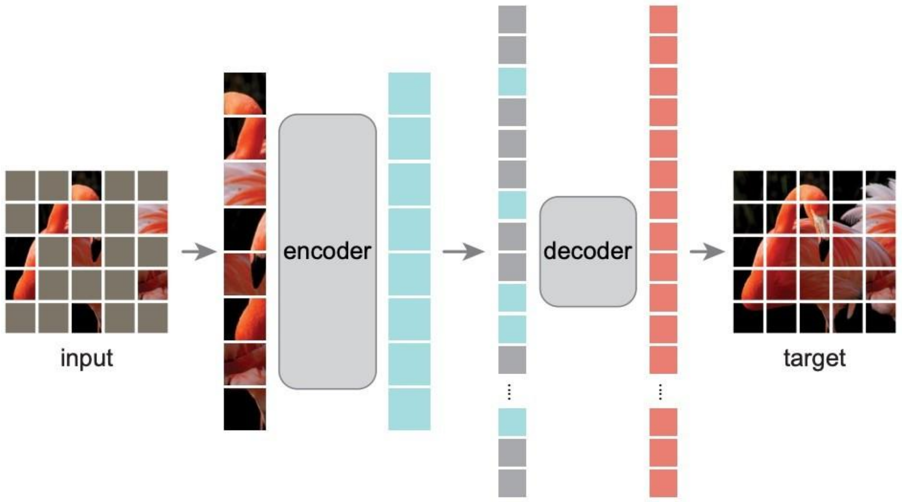
Language

[Devlin et al, NAACL 2019] [Brown et al, NeurIPS 2020]

- **Foundation Models for Vision**
  - Fixing a foundation model (e.g., trained via self-supervised learning) and only adapting a simple task-specific model is sufficient for many problems
  - This lecture will cover following foundation models for vision
    - Discriminative and generative models (e.g., self-supervised models, CLIP)



CLIP [Radford et al., '21]



MAE [He et al., '21]

- **Foundation Models for Vision**
  - Fixing a foundation model (e.g., trained via self-supervised learning) and only adapting a simple task-specific model is sufficient for many problems
  - This lecture will cover following foundation models for vision
    - Discriminative and generative models (e.g., self-supervised models, CLIP)
    - Vision-specific models (e.g., Segment Anything (SAM),

Segment Anything [Meta AI, '22]

- **Foundation Models for Vision**
  - Fixing a foundation model (e.g., trained via self-supervised learning) and only adapting a simple task-specific model is sufficient for many problems
  - This lecture will cover following foundation models for vision
    - Discriminative and generative models (e.g., self-supervised models, CLIP)
    - Vision-specific models (e.g., Segment Anything (SAM)

  - In specific, this lecture will answer (or at least hint) to the following questions:
    - How to train foundation models?
    - What are the zero-shot capabilities of foundation models?
    - How to exploit foundation models on specific tasks?

Discriminative Visual Foundation Models: Overview

We are interested in visual representations that extract high-level semantics which can be applied to various **downstream tasks** such as

- Supervised learning (e.g., classification, detection)
- Unsupervised learning (e.g., clustering, metric learning)
- Modular component for multimodal understanding (e.g., image-text retrieval, visual question answering)

Scaling model and data size is key recipe in training foundation models:

- The loss function must be designed to be scalable and stable
- The data should be curated to remove bias or noisy label
- Computation efficiency to lower the training cost

Table of Contents

Self-supervised learning: Overview

Introduce self-supervised learning (SSL) methods:

- **Discriminative Visual Foundation Models**
  - Invariance based methods such as contrastive learning (CLIP, DINO)

- **Generative Visual Foundation Models**
  - Masked image modeling (MIM)

Recall: Supervised vs Unsupervised Learning

Supervised Learning

Data: (x, y)

x is data, y is label

Goal: Learn a *function* to map x -> y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.

Unsupervised Learning

Data: x

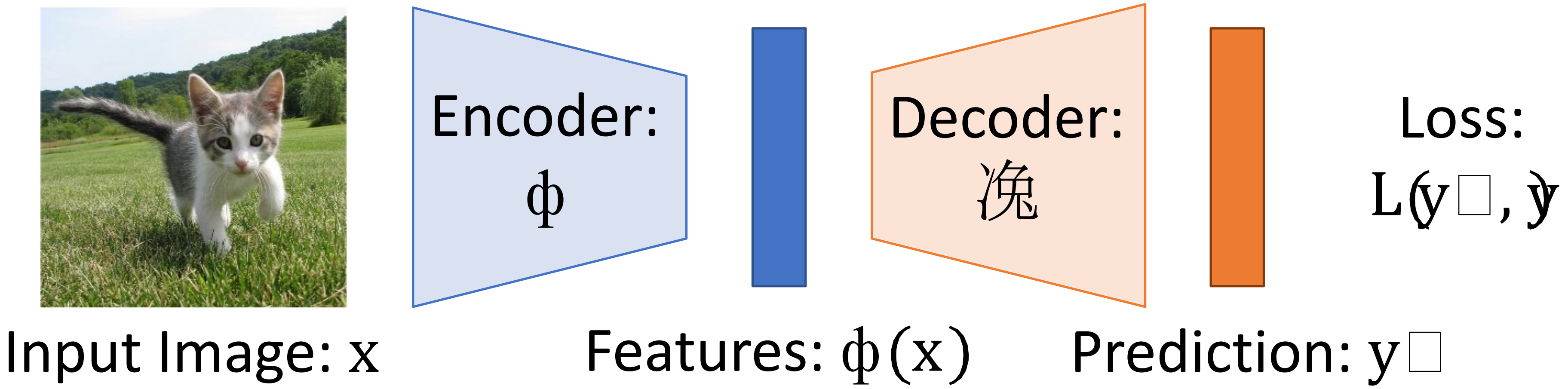
Just data, no labels!

Goal: Learn some underlying hidden *structure* of the data

Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.

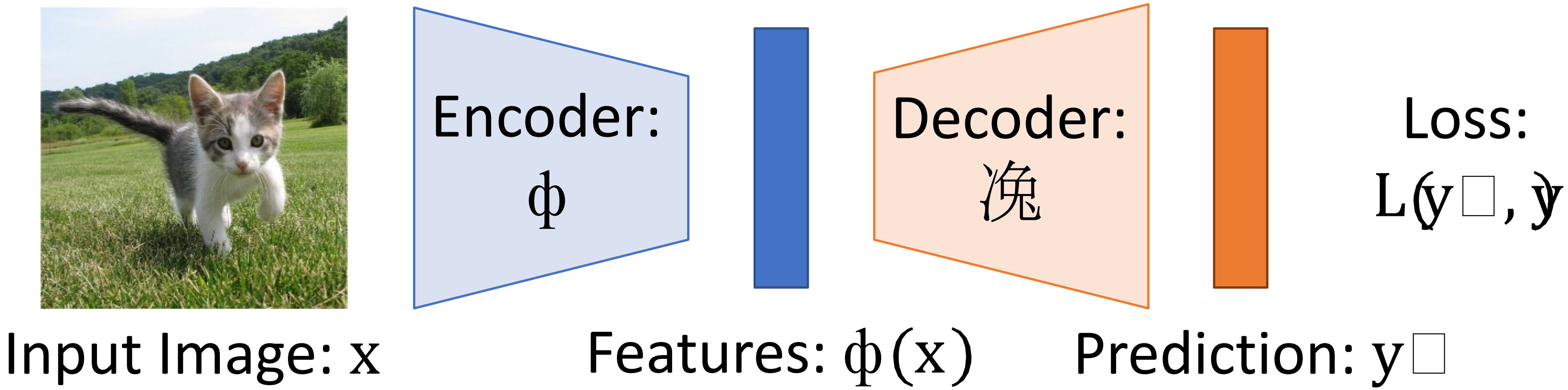
Self-Supervised Learning: Pretext then Transfer

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

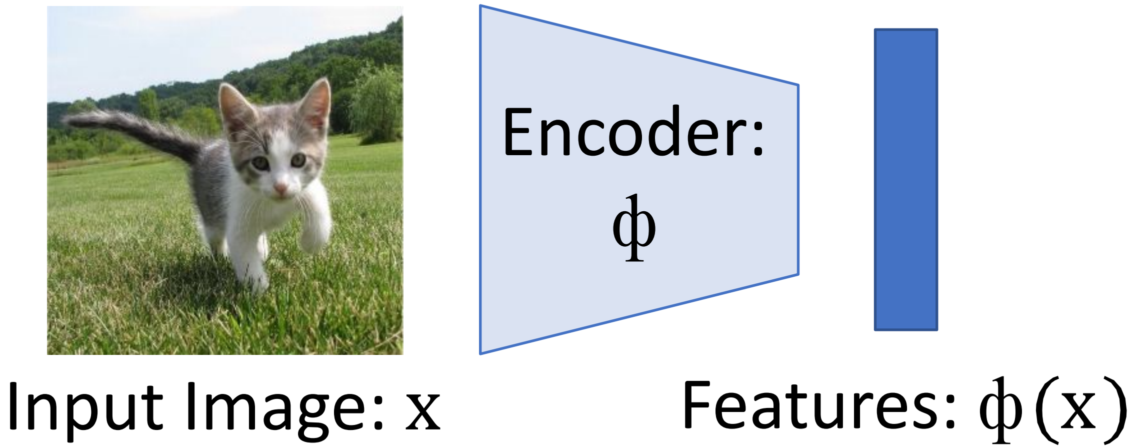


Self-Supervised Learning: Pretext then Transfer

Step 1: <u>Pretrain</u> a network on a <u>pretext task</u> that doesn't require supervision



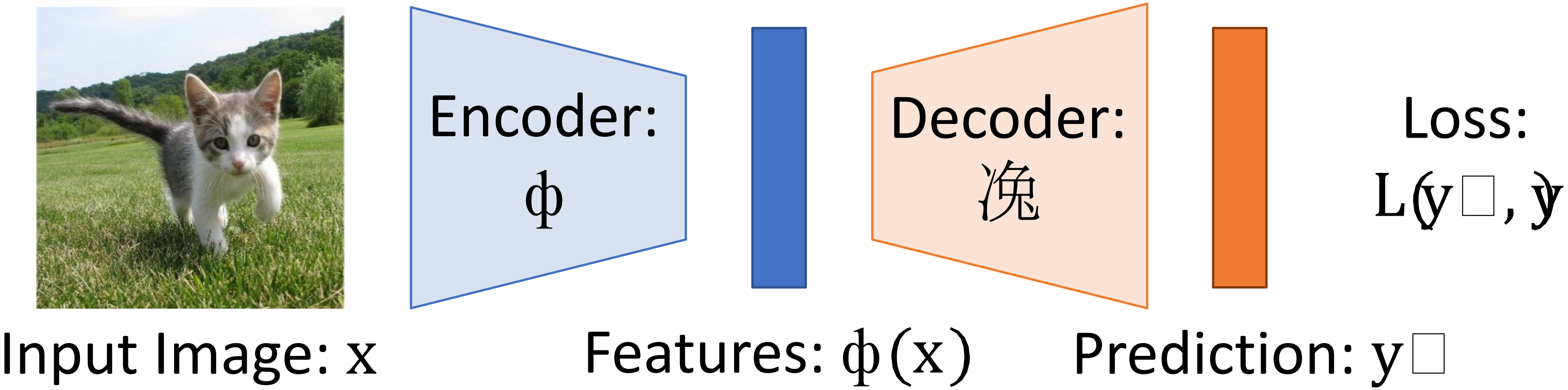
Step 2: Transfer encoder to <u>downstream tasks</u> via linear classifiers, KNN, finetuning



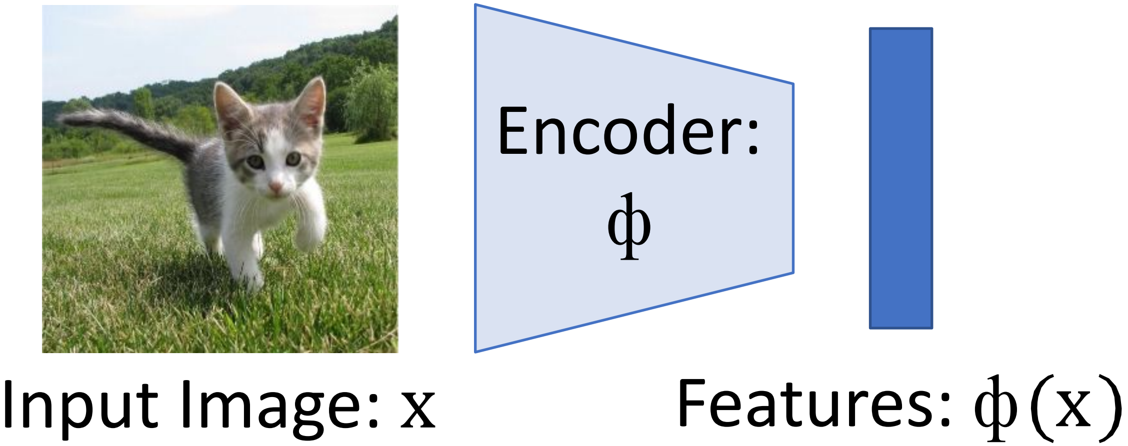
Downstream tasks: Image classification, object detection, semantic segmentation

Self-Supervised Learning: Pretext then Transfer

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



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



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

Self-Supervised Learning: Pretext Tasks

Discriminative: Predict something about the input signal

- Context prediction
- Rotation
- Clustering
- Contrastive

Generative: Predict part of the input signal

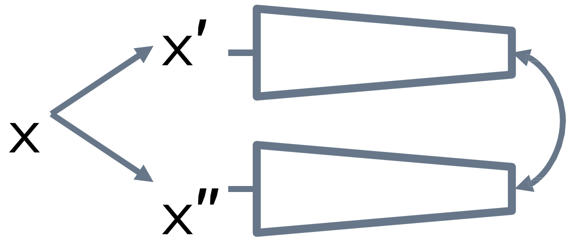
- Autoencoders (sparse, denoising, masked)
- Autoregressive
- GANs
- Colorization
- Inpainting

Multimodal: Use some additional signal in addition to RGB images

- Video
- 3D
- Sound
- Language

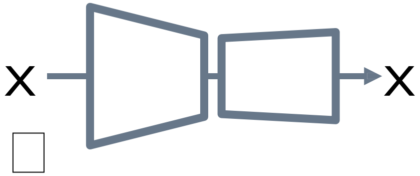
Self-Supervised Paradigms in Vision

- Contrastive / Siamese



  - Compare data points in the latent *representation* space
  - Computer vision: SimCLR, MoCo, BYOL, DINO, …, with *augmentations*

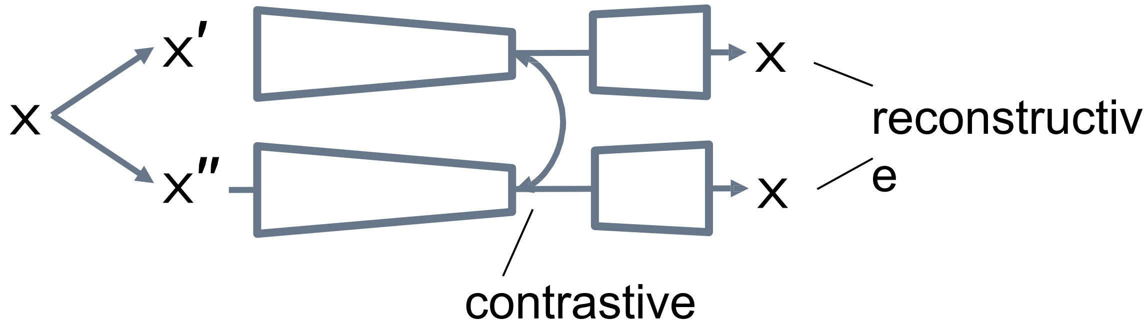
- Reconstructive / Auto-Encoding



  - Reconstruct *corrupted* data points
  - *Grounded* in the input space
  - Paradigm of BERT & GPT in NLP
  - Computer Vision: MAE

Self-Supervised Paradigms in Vision

- “Contrastive + Reconstructive” is also possible



- Multi-tasking makes representations more *versatile*: iBOT, MAGE
- But the pipeline is *less clean* to understand scientifically

Table of Contents

Contrastive Learning

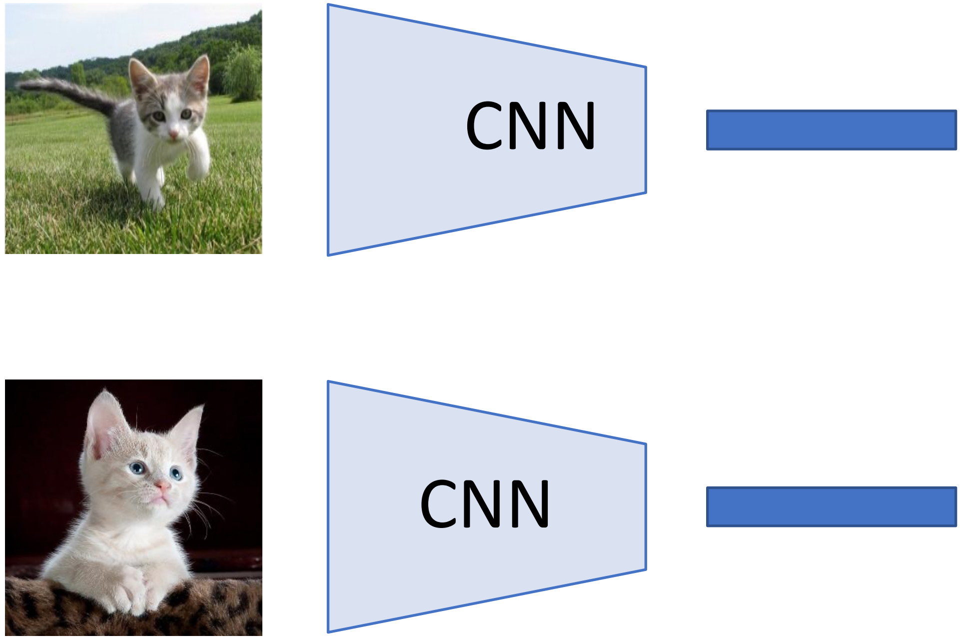
Assume we don't have labels for images, but we know whether some pairs of images are similar or dissimilar

Hadsell et al, "Dimensionality Reduction by Learning and Invariant Mapping", CVPR 2006

Contrastive Learning

Assume we don't have labels for images, but we know whether some pairs of images are similar or dissimilar

Similar images should have similar features

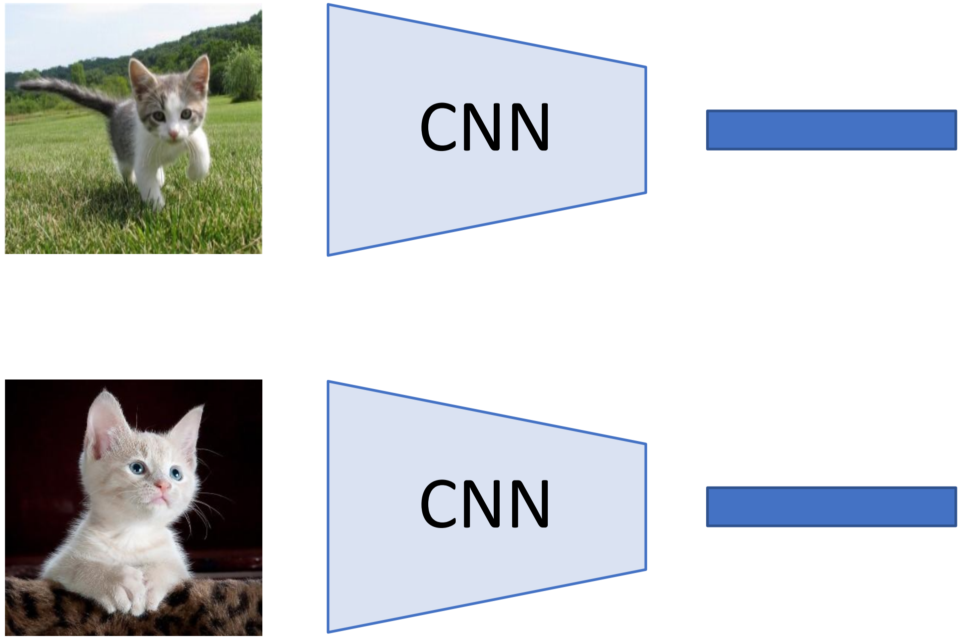


Hadsell et al, "Dimensionality Reduction by Learning and Invariant Mapping", CVPR 2006

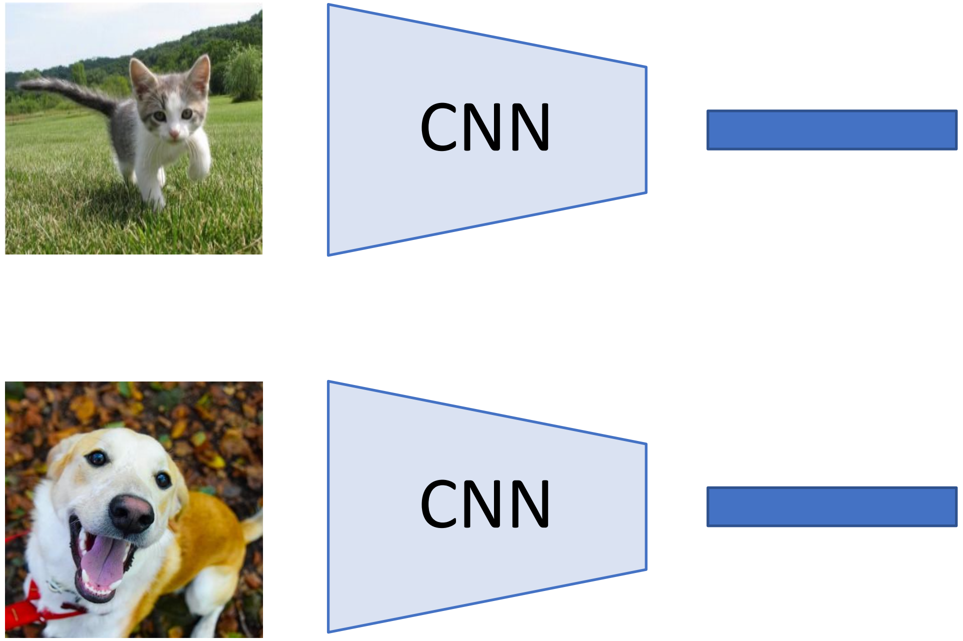
Contrastive Learning

Assume we don't have labels for images, but we know whether some pairs of images are similar or dissimilar

Similar images should have similar features



Dissimilar images should have dissimilar features



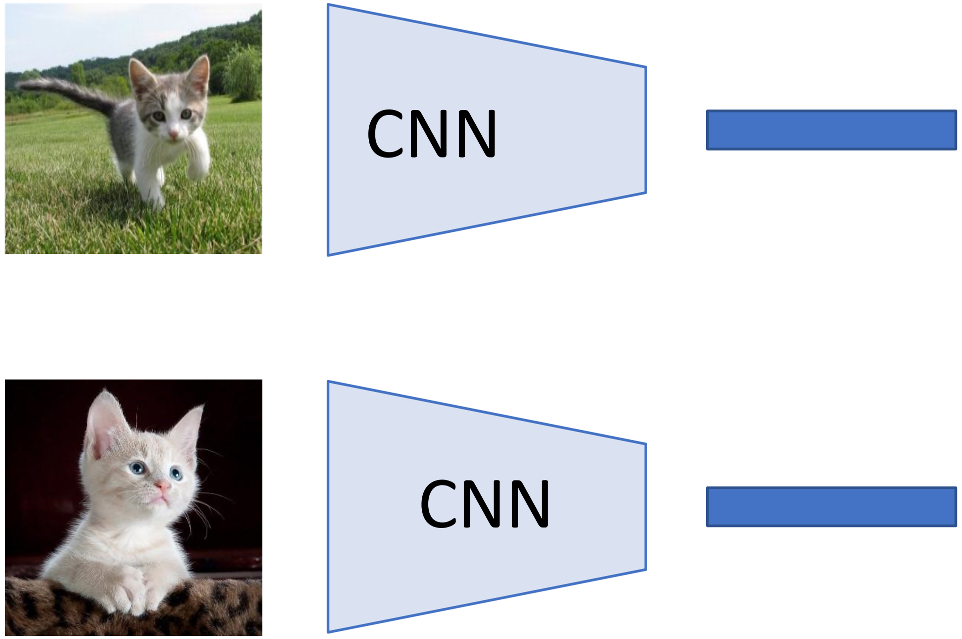
Hadsell et al, "Dimensionality Reduction by Learning and Invariant Mapping", CVPR 2006

Contrastive Learning

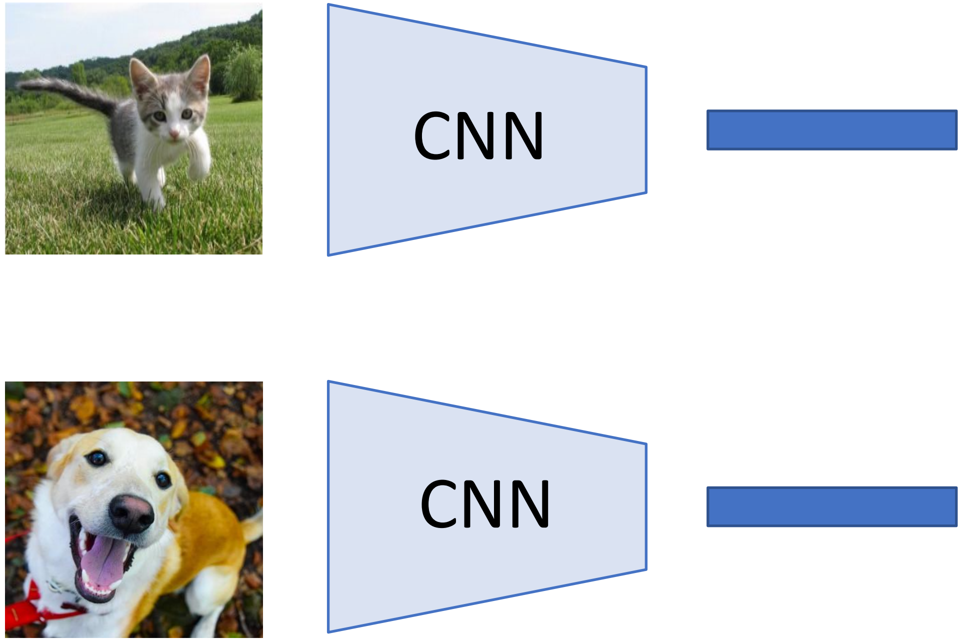
Assume we don't have labels for images, but we know whether some pairs of images are similar or dissimilar

Let $d = \|\phi(x_1) - \phi(x_2)\|_2$ be the Euclidean distance between features for two images

Similar images should have similar features



Dissimilar images should have dissimilar features



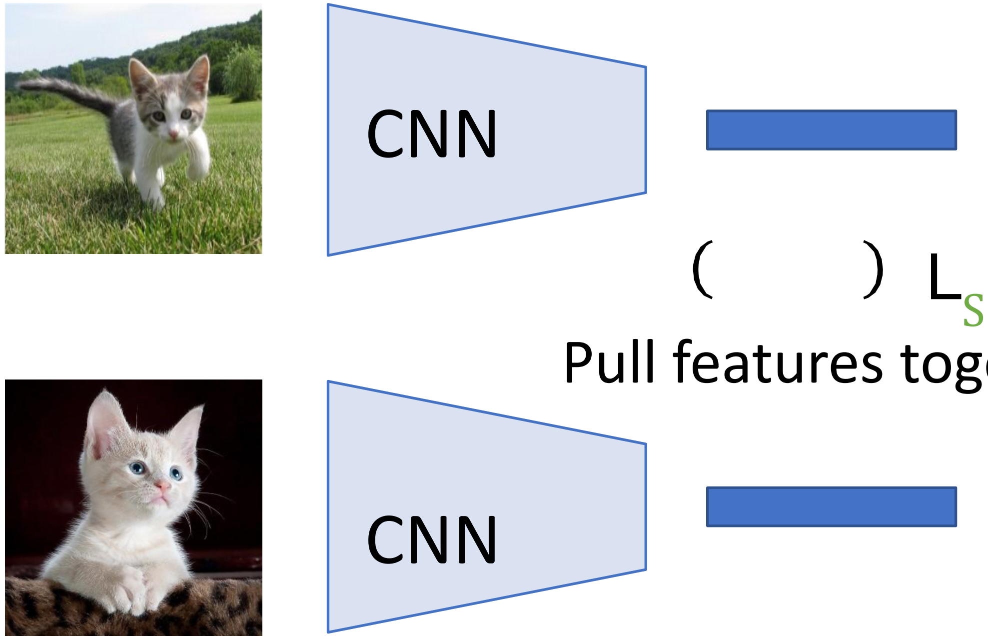
Hadsell et al, "Dimensionality Reduction by Learning and Invariant Mapping", CVPR 2006

Contrastive Learning

Assume we don't have labels for images, but we know whether some pairs of images are similar or dissimilar

Let $d = ||\phi(x_1) - \phi(x_2)||_2$ be the Euclidean distance between features for two images

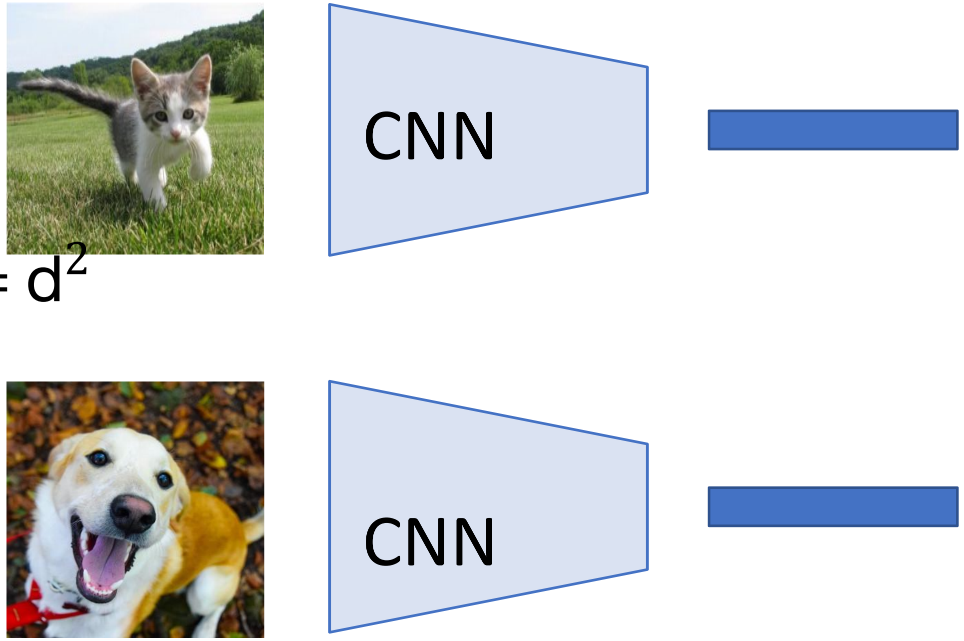
Similar images should have similar features



$L_S(x_1, x_2) = d^2$

Pull features together

Dissimilar images should have dissimilar features



Hadsell et al, "Dimensionality Reduction by Learning and Invariant Mapping", CVPR 2006

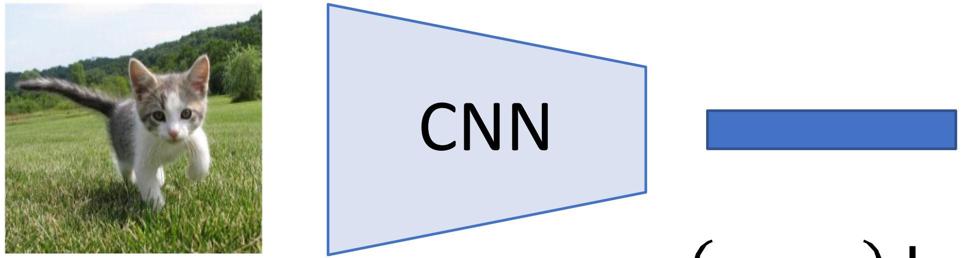
Contrastive Learning

Assume we don't have labels for images, but we know whether some pairs of images are similar or dissimilar

Let $d = \|\phi(x_1) - \phi(x_2)\|_2$ be the Euclidean distance between features for two images

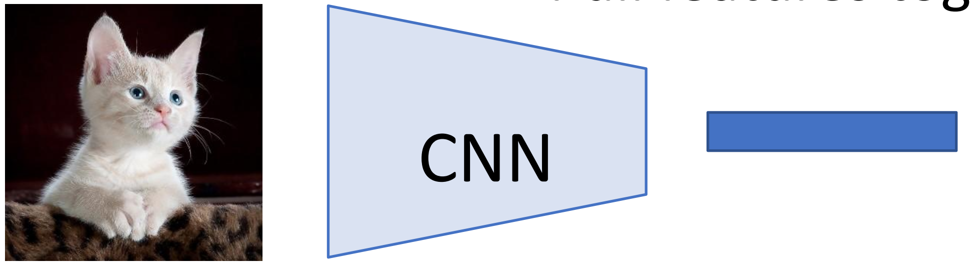
Similar images should have similar features

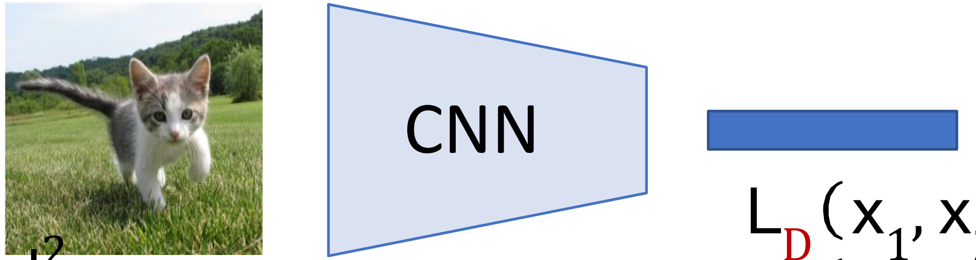
Dissimilar images should have dissimilar features



$L_S(x_1, x_2) = d^2$

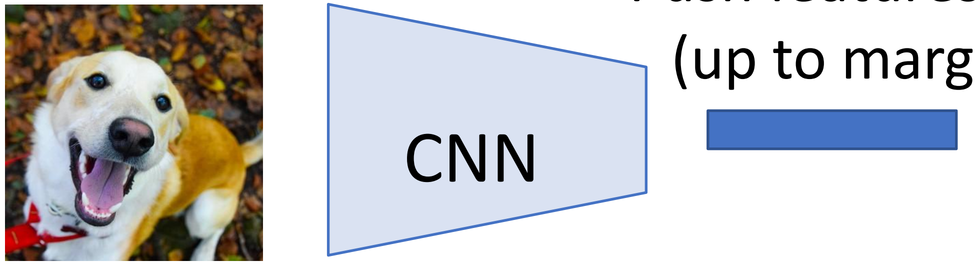
Pull features together





$L_D(x_1, x_2) = \max(0, m - d)^2$

Push features apart (up to margin m)



Hadsell et al, "Dimensionality Reduction by Learning and Invariant Mapping", CVPR 2006

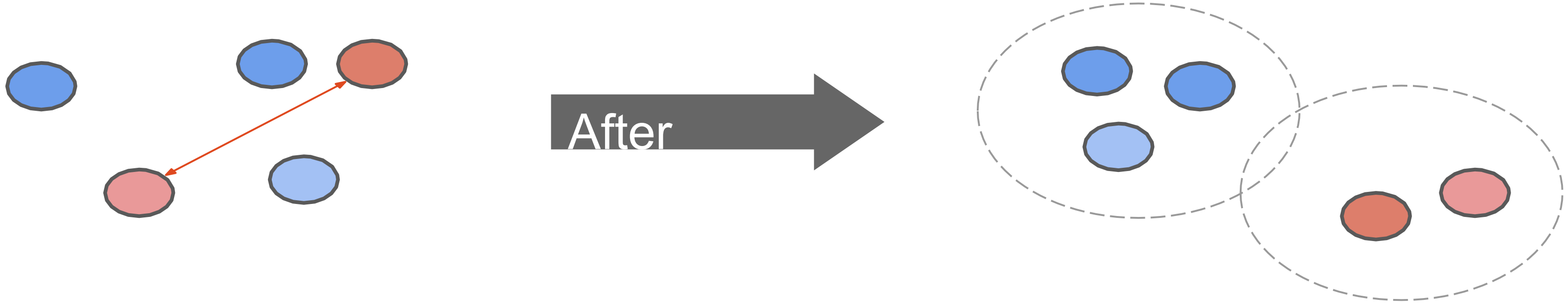
SSL via Invariance

Core idea of invariance-based learning:

- **Invariance:** Representations of related samples should be similar
- **Contrast** (optional)**:** Representations of unrelated samples should be dissimilar

Positive pair $f($  $) \approx f($  $)$

Negative pair $f($  $) \neq f($  $)$



SSL via Invariance

Core idea of invariance-based learning:

- **Invariance:** Representations of related samples should be similar
- **Contrast** (optional)**:** Representations of unrelated samples should be dissimilar

Positive pair $f($  $) \approx f($  $)$

Negative pair $f($  $) \neq f($  $)$

- **Q)** How to construct positive/negative pairs in the unsupervised setting?
- **A)** Positive samples are constructed from
  - Similar samples (e.g., in the same cluster)
  - Same instance of different data augmentation
  - Additional structures (e.g., multi-view images, video) (negative samples = not positive samples)

Problem: Where to get positive and negative pairs?

Contrastive Learning Loss: Inter-Sample Classification

Given both similar (“positive”) and dissimilar (“negative”) candidates, to identify which ones are similar to the anchor data point is a *classification* task.

There are ways to construct a set of data point candidates:

1. The original input and its distorted version
2. Data that captures the same target from different views

3.1.1 Data Augmentation

Techniques: Data Augmentation

Data augmentation setup is critical for learning good embedding.

It introduces the non-essential variations into examples without modifying semantic meanings and thus encourages the model to learn the essential part within the representation.

- **Image augmentation**
- Text augmentation

Techniques: Image Augmentation

- Basic Image Augmentation
  - Random crop
  - color distortion
  - Gaussian blur
  - color jittering
  - random flip/rotation
  - etc.
- Augmentation Strategies
- Image Mixture

Techniques: Image Augmentation

- Basic Image Augmentation
- Augmentation Strategies
  - AutoAugment (Cubuk, et al. 2018): inspired by NAS
  - RandAugment (Cubuk et al. 2019): reduces NAS search space in AutoAugment.
  - PBA (Population based augmentation; Ho et al. 2019): evolutionary algorithm
  - UDA (Unsupervised Data Augmentation; Xie et al. 2019): minimize the KL divergence between the predicted distribution over an unlabelled example and its unlabelled augmented version.
- Image Mixture

Techniques: Image Augmentation

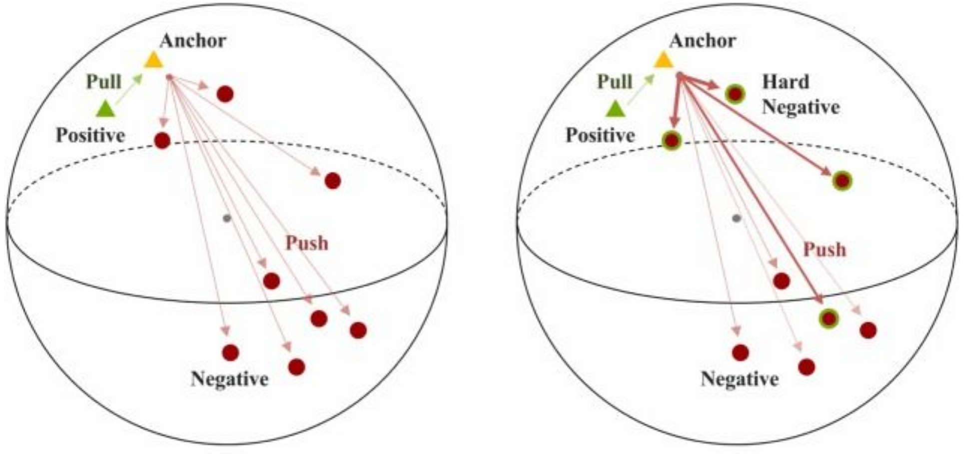
- Basic Image Augmentation
- Augmentation Strategies
- Image Mixture
  - Mixup (Zhang et al 2018): weighted pixel-wise combination of two images.
  - Cutmix (Yun et al 2019): mix in a local region of one image into the other.
  - MoCHi ("Mixing of Contrastive Hard Negatives"; Kalantidis et al 2020): mixture of hard negative samples.

Hard Negative Mining

Hard negative samples are different to learn. They should have different labels from the anchor sample, but the embedding features may be very close.

Hard negative mining is important for contrastive learning.

Challenging negative samples encourages the model to learn better representations that can distinguish hard negatives from true positives.



(a) Negative (b) Hard Negative

Hard Negative Mining

Explicit hard negative mining

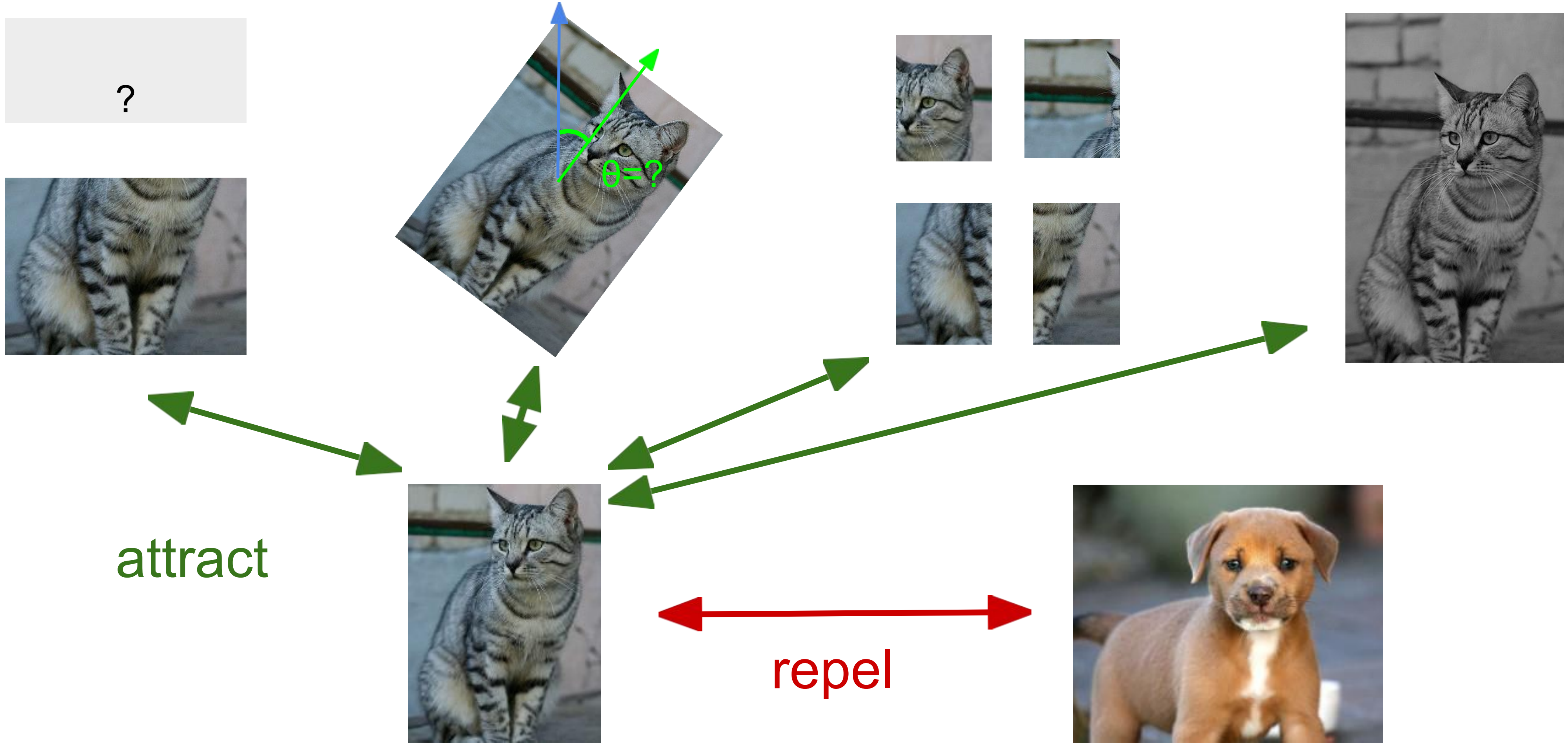
- **MoCHi** (Kalantidis et al. 2020): mine hard negative by sorting them according to similarity to the query in descending order.
- Extract task-specific hard negative samples from labelled datasets.
    - e.g. "`contradiction`" sentence pairs from NLI datasets. (Most sentence embedding papers)
- Keyword based retrieval
    - e.g. BM25 search results (Karpukhin et al. 2020)
- Upweight the negative sample probability to be proportional to its similarity to the anchor sample (Robinson et al. 2021)

Hard Negative Mining

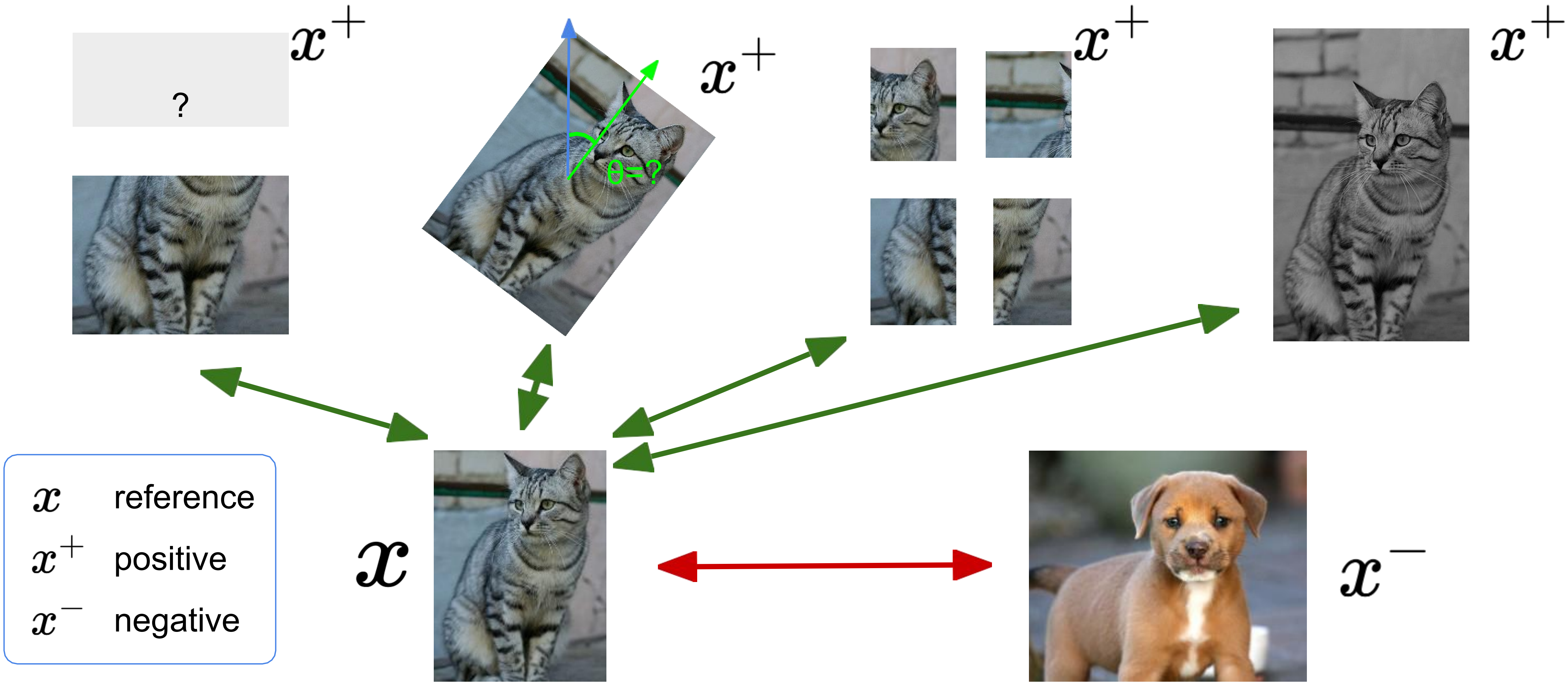
Implicit hard negative mining

- In-batch negative samples
- Memory bank (Wu et al. 2018, He et al. 2019) → Increase batch size
- Large batch size via various training parallelism

Contrastive Representation Learning



Contrastive Representation Learning



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^-).

A formulation of contrastive learning

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

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

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^-

...

A formulation of contrastive learning

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

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+)))}{\underbrace{\exp(s(f(x), f(x^+)))}_{\text{score for the positive pair}} + \underbrace{\sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-)))}_{\text{score for the N-1 negative pairs}}} \right]$$

This seems familiar …

A formulation of contrastive learning

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

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+)))}{\underbrace{\exp(s(f(x), f(x^+)))}_{\text{score for the positive pair}} + \underbrace{\sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-)))}_{\text{score for the N-1 negative pairs}}} \right]$$

This seems familiar …

Cross entropy loss for a N-way softmax classifier!
I.e., learn to find the positive sample from the N samples

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*$^+$*)*

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

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

Detailed derivation: Poole et al., 2019

3.1.2 Losses

Contrastive Learning Loss: Inter-Sample Classification

Common loss functions:

- Contrastive loss (Chopra et al. 2005)
- Triplet loss (Schroff et al. 2015; FaceNet)
- Lifted structured loss (Song et al. 2015)
- Multi-class n-pair loss (Sohn 2016)
- Noise contrastive estimation (“NCE”; Gutmann & Hyvarinen 2010)
- InfoNCE (van den Oord, et al. 2018)
- Soft-nearest neighbors loss (Salakhutdinov & Hinton 2007, Frosst et al. 2019)

Contrastive Learning Loss: Inter-Sample Classification

Contrastive loss (Chopra et al. 2005): Works with labelled dataset.

Encodes data into an embedding vector:

- Examples from the same class have similar embeddings.
- Samples from different classes have different ones.

Given two labeled data pairs $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$:

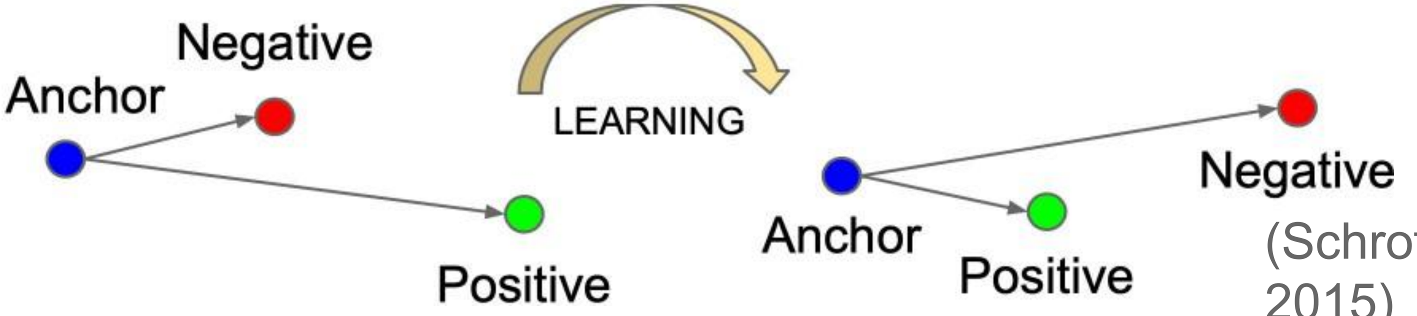
$$\mathcal{L}_{\text{cont}}(\mathbf{x}_i, \mathbf{x}_j, \theta) = \mathbb{1}[y_i = y_j] \underbrace{\boxed{\|f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}_j)\|_2^2}}_{\text{minimize}} + \mathbb{1}[y_i \neq y_j] \max(0, \epsilon - \underbrace{\boxed{\|f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}_j)\|_2}}_{\text{maximize}})^2$$

Contrastive Learning Loss: Inter-Sample Classification

Triplet loss (Schroff et al. 2015): learns to minimize the distance between the anchor x and positive x+ and maximize the distance between the anchor x and negative x- at the same time.

Given a triplet input $(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-)$

$$\mathcal{L}_{\text{triplet}}(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-) = \sum_{\mathbf{x} \in \mathcal{X}} \max\left(0, \|f(\mathbf{x}) - f(\mathbf{x}^+)\|_2^2 - \|f(\mathbf{x}) - f(\mathbf{x}^-)\|_2^2 + \epsilon\right)$$



(Schroff et al. 2015)

Contrastive Learning Loss: Inter-Sample Classification

N-pair loss (Sohn 2016) generalizes triplet loss to include comparison with multiple negative samples.

Given one positive and N-1 negative samples $\{\mathbf{x}, \mathbf{x}^+, \mathbf{x}_1^-, \ldots, \mathbf{x}_{N-1}^-\}$

$$\mathcal{L}_{\text{N-pair}}(\mathbf{x}, \mathbf{x}^+, \{\mathbf{x}_i^-\}_{i=1}^{N-1}) = \log\left(1 + \sum_{i=1}^{N-1} \exp(f(\mathbf{x})^\top f(\mathbf{x}_i^-) - f(\mathbf{x})^\top f(\mathbf{x}^+))\right)$$

$$= -\log \frac{\exp(f(\mathbf{x})^\top f(\mathbf{x}^+))}{\exp(f(\mathbf{x})^\top f(\mathbf{x}^+)) + \sum_{i=1}^{N-1} \exp(f(\mathbf{x})^\top f(\mathbf{x}_i^-))}$$

Contrastive Learning Loss: Inter-Sample Classification

Lifted structured loss (Song et al. 2015): utilizes all the pairwise edges within one training batch for better computational efficiency.

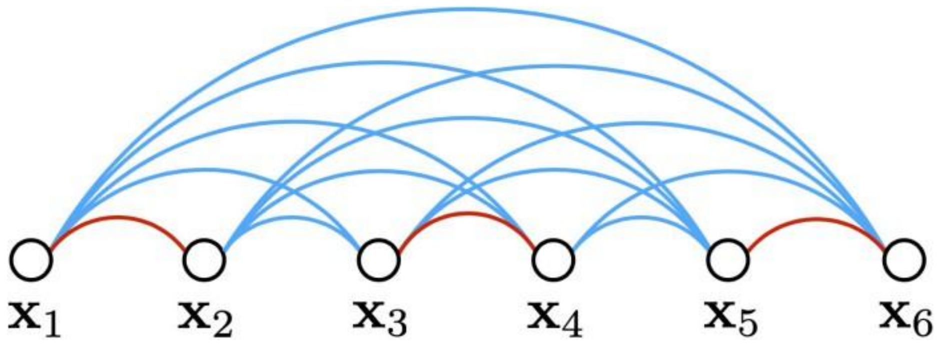
$$\mathcal{L}^{(ij)}_{\text{struct}} = D_{ij} + \log \Big(\sum_{(i,k)\in\mathcal{N}} \exp(\epsilon - D_{ik}) + \sum_{(j,l)\in\mathcal{N}} \exp(\epsilon - D_{jl}) \Big)$$

where $D_{ij} = \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|_2$

$(i, j) \in \mathcal{P}$

$\mathcal{P}$ set of positive pairs

$\mathcal{N}$ set of negative pairs



(Song et al. 2015)

Contrastive Learning Loss: Inter-Sample Classification

Noise Contrastive Estimation (NCE) (Gutmann & Hyvarinen 2010) runs logistic regression to tell apart the target data from noise.

Given target sample distribution p and noise distribution q,

$$\mathcal{L}_{\text{NCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[\log\sigma(\ell_\theta(\mathbf{x}_i)) + \log(1-\sigma(\ell_\theta(\tilde{\mathbf{x}}_i)))\right]$$

just cross entropy

where logit $\ell_\theta(\mathbf{u}) = \log\frac{p_\theta(\mathbf{u})}{q(\mathbf{u})} = \log p_\theta(\mathbf{u}) - \log q(\mathbf{u})$

sigmoid $\sigma(\ell) = \frac{1}{1+\exp(-\ell)} = \frac{p_\theta}{p_\theta + q}$

Contrastive Learning Loss: Inter-Sample Classification

InfoNCE (van den Oord, et al. 2018):
uses categorical cross-entropy loss to identify the positive sample amongst a set of unrelated noise samples.

Given a context vector c, the positive sample should be drawn from the conditional distribution p(x|c), while N−1 negative samples are drawn from the proposal distribution p(x), independent from the context c.

The probability of detecting the positive sample correctly is:

$$p(C = \texttt{pos}|X, \mathbf{c}) = \frac{f(\mathbf{x}_{\texttt{pos}}, \mathbf{c})}{\sum_{j=1}^{N} f(\mathbf{x}_j, \mathbf{c})}$$

where the density function is

$$f(\mathbf{x}, \mathbf{c}) \propto \frac{p(\mathbf{x}|\mathbf{c})}{p(\mathbf{x})}$$

Contrastive Learning Loss: Inter-Sample Classification

Soft-Nearest Neighbors Loss (Frosst et al. 2019) extends the loss function to include multiple positive samples given known labels.

Given a batch of samples $\{\mathbf{x}_i, y_i)\}_{i=1}^{B}$,

$$\mathcal{L}_{\text{snn}} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\sum_{i\neq j, y_i=y_j, j=1,\ldots,B} \exp(-f(\mathbf{x}_i, \mathbf{x}_j)/\tau)}{\sum_{i\neq k, k=1,\ldots,B} \exp(-f(\mathbf{x}_i, \mathbf{x}_k)/\tau)}$$

temperature term

3.1.3 Framework

SSL via Invariance

- **Instantiations of invariance-based approach**
  - Many classes of self-supervised learning can be viewed as invariance-based

  - **Contrastive learning**
    - Attract similar samples and dispel dissimilar samples
    - E.g., MoCo, SimCLR, CLIP
    -
  - **Clustering & pseudo-labeling**
    - Cluster data into K groups, and assume they are pseudo-labels
    - Distill pseudo-labels to the self-supervised classifier (strengthen the similarity)
    - E.g., DeepCluster, SwAV, DINO

  - **Consistency regularization**
    - Attract similar samples
    - E.g., MixMatch, UDA, BYOL

SimCLR: A Simple Framework for Contrastive Learning

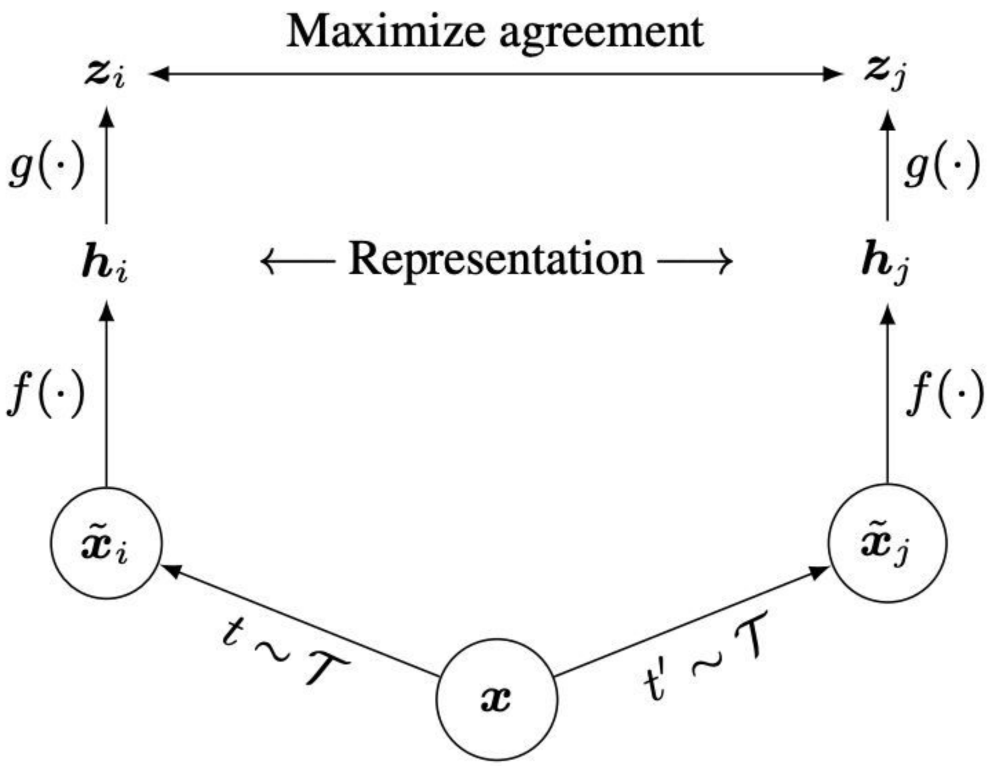
- A **simple** framework for contrastive learning without requiring specialized architectures or a memory bank

- Cosine similarity as the score function:

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

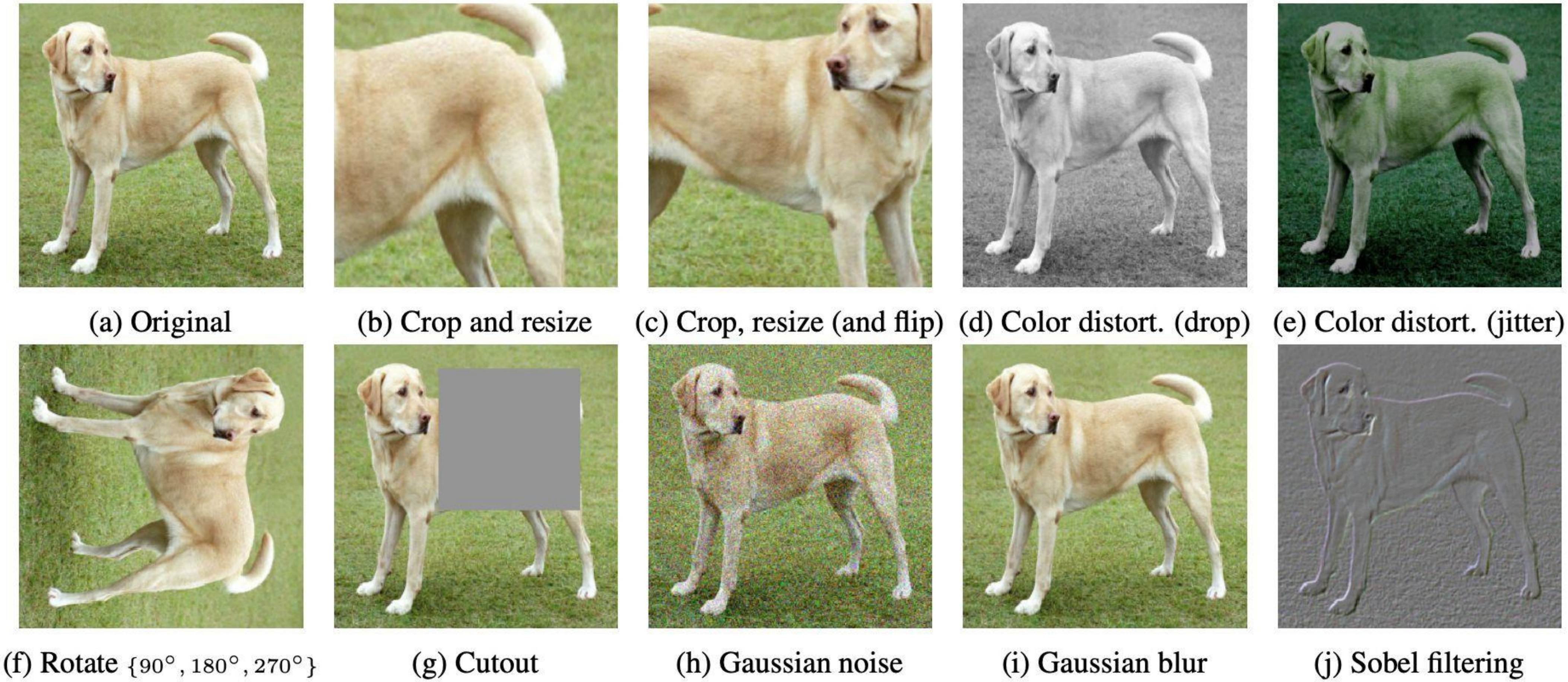
- Use a projection network *g(·)* to project features to a space where contrastive learning is applied

- Generate positive samples through data augmentation:
  - random cropping, random color distortion, and random blur.



Source: [Chen et al., 2020]

SimCLR: generating positive samples from data augmentation

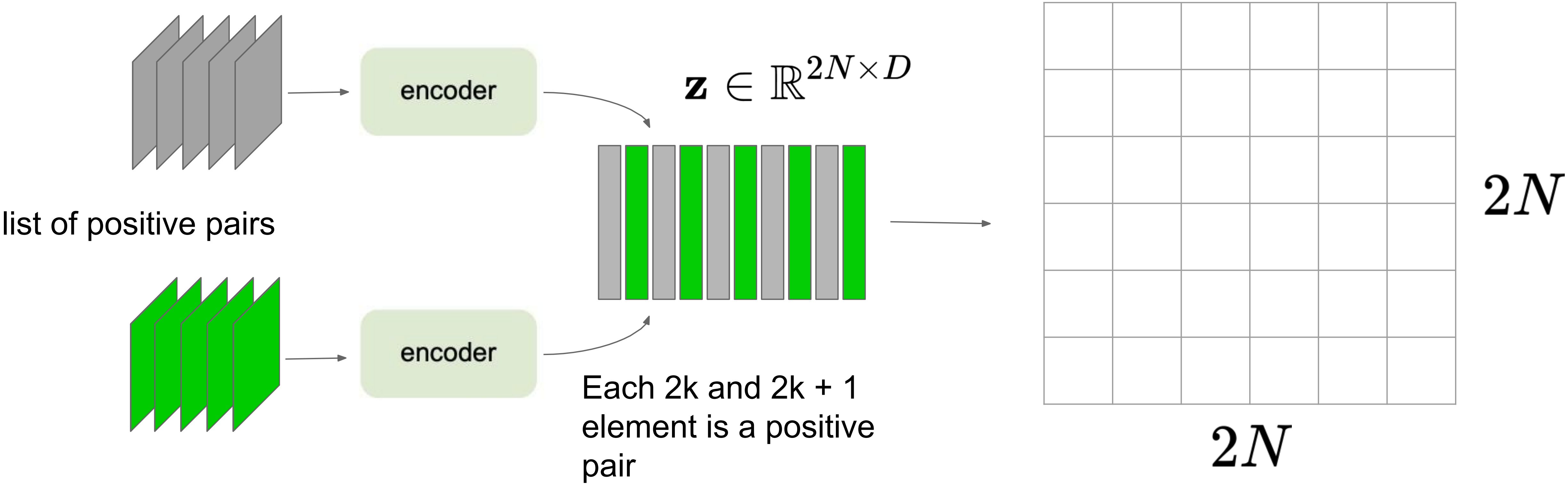


Source: [Chen et al., 2020]

SimCLR: mini-batch training

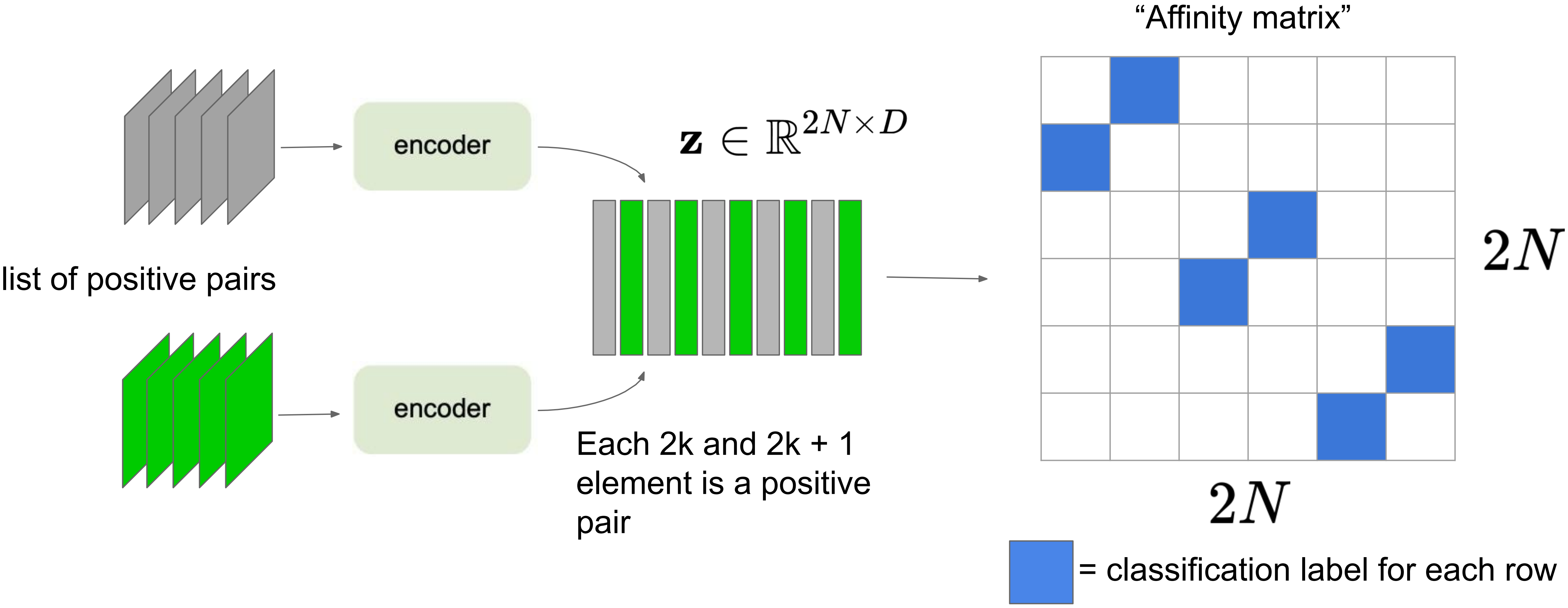
$$s_{i,j} = \frac{z_i^T z_j}{||z_i|| \, ||z_j||}$$

“Affinity matrix”



SimCLR: mini-batch training

$$s_{i,j} = \frac{z_i^T z_j}{||z_i|| \, ||z_j||}$$



SimCLR

Generate a positive pair by sampling data augmentation functions

Algorithm 1 SimCLR's main learning algorithm.

input: batch size N, constant τ, structure of f, g, $\mathcal{T}$.
for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ **do**
  for all $k \in \{1, \dots, N\}$ **do**
    draw two augmentation functions $t \sim \mathcal{T}$, $t' \sim \mathcal{T}$
    # the first augmentation
    $\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$ # representation
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$ # projection
    # the second augmentation
    $\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$ # representation
    $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$ # projection
  end for
  for all $i \in \{1, \dots, 2N\}$ and $j \in \{1, \dots, 2N\}$ **do**
    $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$ # pairwise similarity
  end for
  define $\ell(i,j)$ **as** $\ell(i,j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away $g(\cdot)$

Source: Chen et al., 2020

SimCLR

Algorithm 1 SimCLR's main learning algorithm.

input: batch size N, constant τ, structure of f, g, $\mathcal{T}$.
for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^N$ **do**
  for all $k \in \{1, \ldots, N\}$ **do**
    $\boxed{\text{draw two augmentation functions } t \sim \mathcal{T},\, t' \sim \mathcal{T}}$
    # the first augmentation
    $\boxed{\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)}$
    $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$ # representation
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$ # projection
    # the second augmentation
    $\boxed{\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)}$
    $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$ # representation
    $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$ # projection
  end for
  for all $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ **do**
    $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$ # pairwise similarity
  end for
  define $\ell(i, j)$ **as** $\boxed{\ell(i,j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away $g(\cdot)$

Generate a positive pair by sampling data augmentation functions

InfoNCE loss:
Use all non-positive samples in the batch as x^-

Source: [Chen et al., 2020]

SimCLR

Algorithm 1 SimCLR's main learning algorithm.

input: batch size N, constant τ, structure of f, g, $\mathcal{T}$.
for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ **do**
  for all $k \in \{1, \ldots, N\}$ **do**
    draw two augmentation functions $t \sim \mathcal{T}$, $t' \sim \mathcal{T}$
    # the first augmentation
    $\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$ # representation
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$ # projection
    # the second augmentation
    $\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$ # representation
    $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$ # projection
  end for
  for all $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ **do**
    $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$ # pairwise similarity
  end for
  define $\ell(i,j)$ **as** $\ell(i,j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away $g(\cdot)$

Generate a positive pair by sampling data augmentation functions

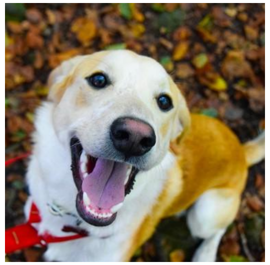
Iterate through and use each of the 2N sample as reference, compute average loss

InfoNCE loss: Use all non-positive samples in the batch as x^-

Source: [Chen et al., 2020]

Contrastive Learning with Data Augmentation

Batch of
N images

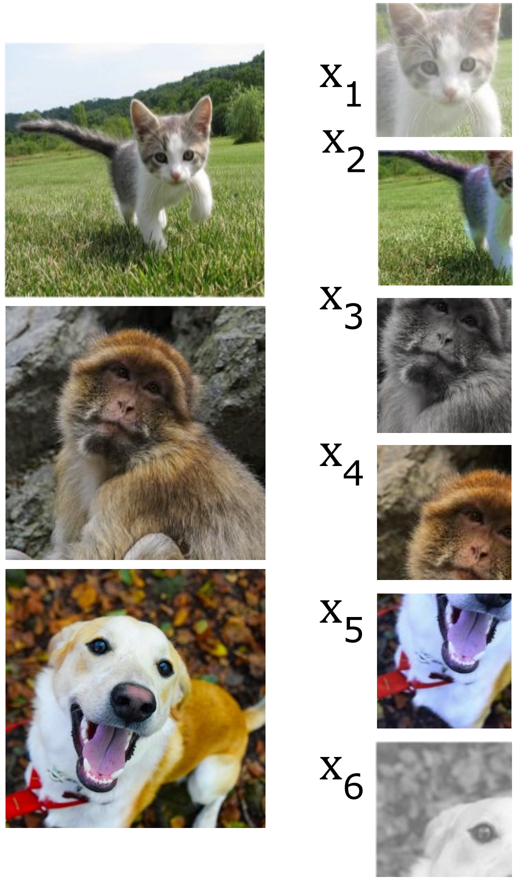


Hadsell et al, "Dimensionality Reduction by Learning and Invariant Mapping", CVPR 2006
Wu et al, "Unsupervised Feature Learning by Non-Parametric Instance-Level Discrimination", CVPR 2018
Van den Oord et al, "Representation Learning with Contrastive Predictive Coding", NeurIPS 2018
Hjelm et al, "Learning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

Contrastive Learning with Data Augmentation

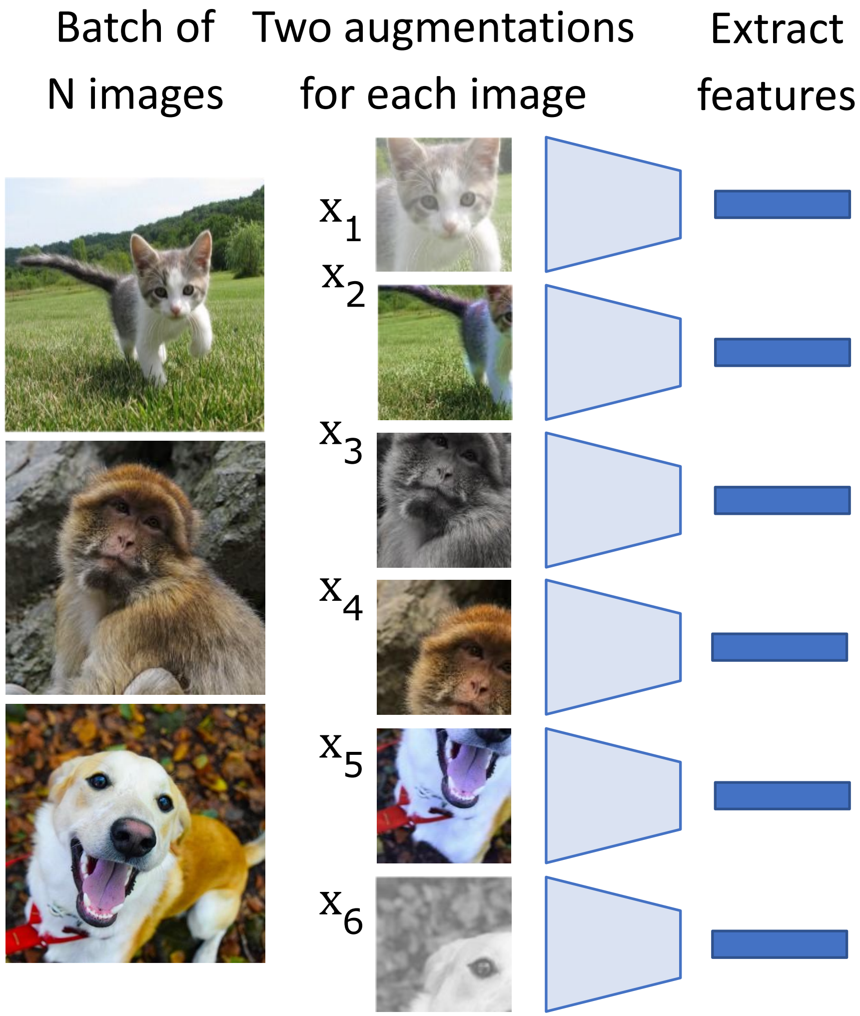
Batch of N images

Two augmentations for each image



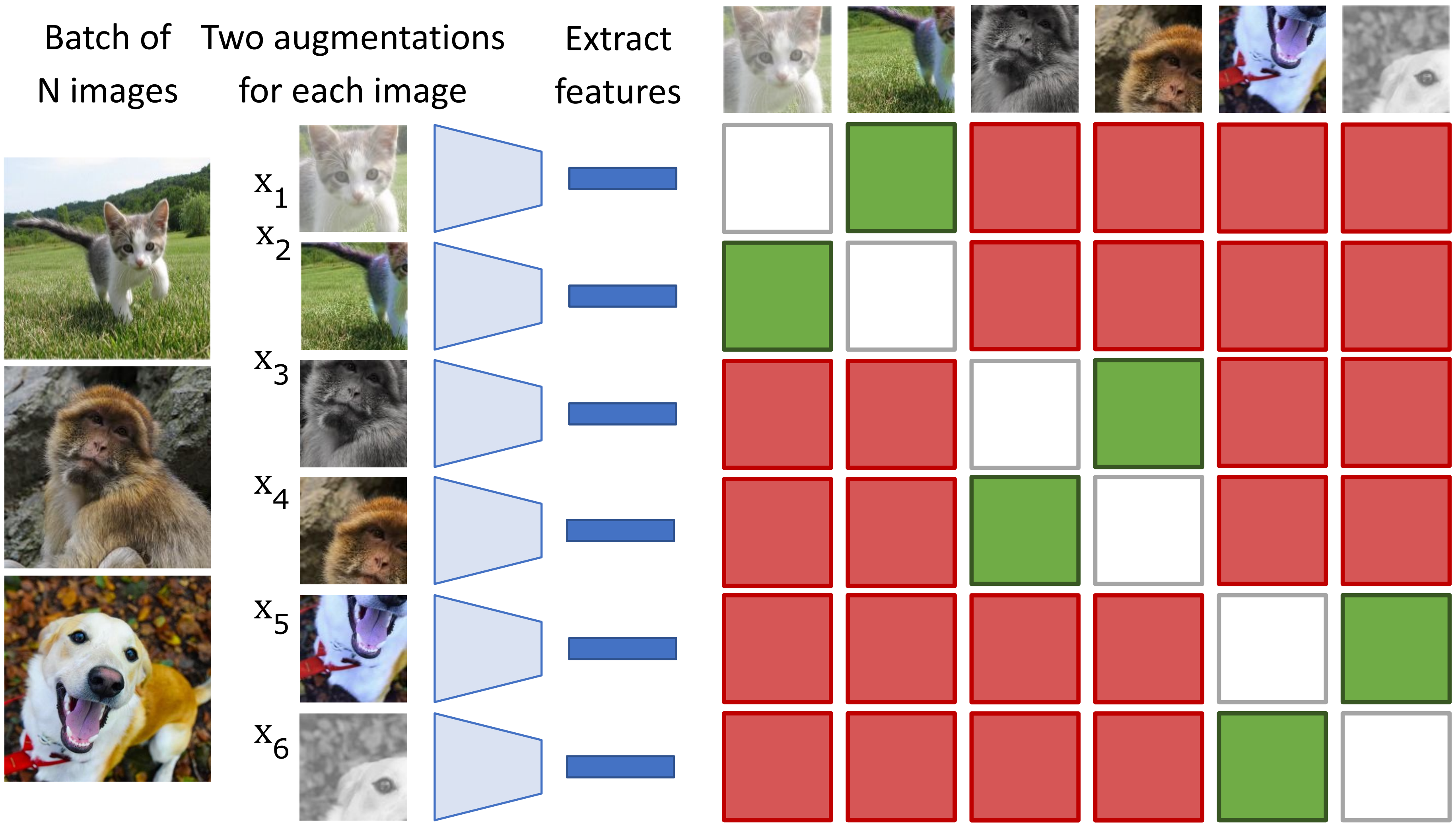
Hadsell et al, “Dimensionality Reduction by Learning and Invariant Mapping”, CVPR 2006
Wu et al, “Unsupervised Feature Learning by Non-Parametric Instance-Level Discrimination”, CVPR 2018
Van den Oord et al, “Representation Learning with Contrastive Predictive Coding”, NeurIPS 2018
Hjelm et al, “Learning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

Contrastive Learning with Data Augmentation



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

Contrastive Learning with Data Augmentation



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

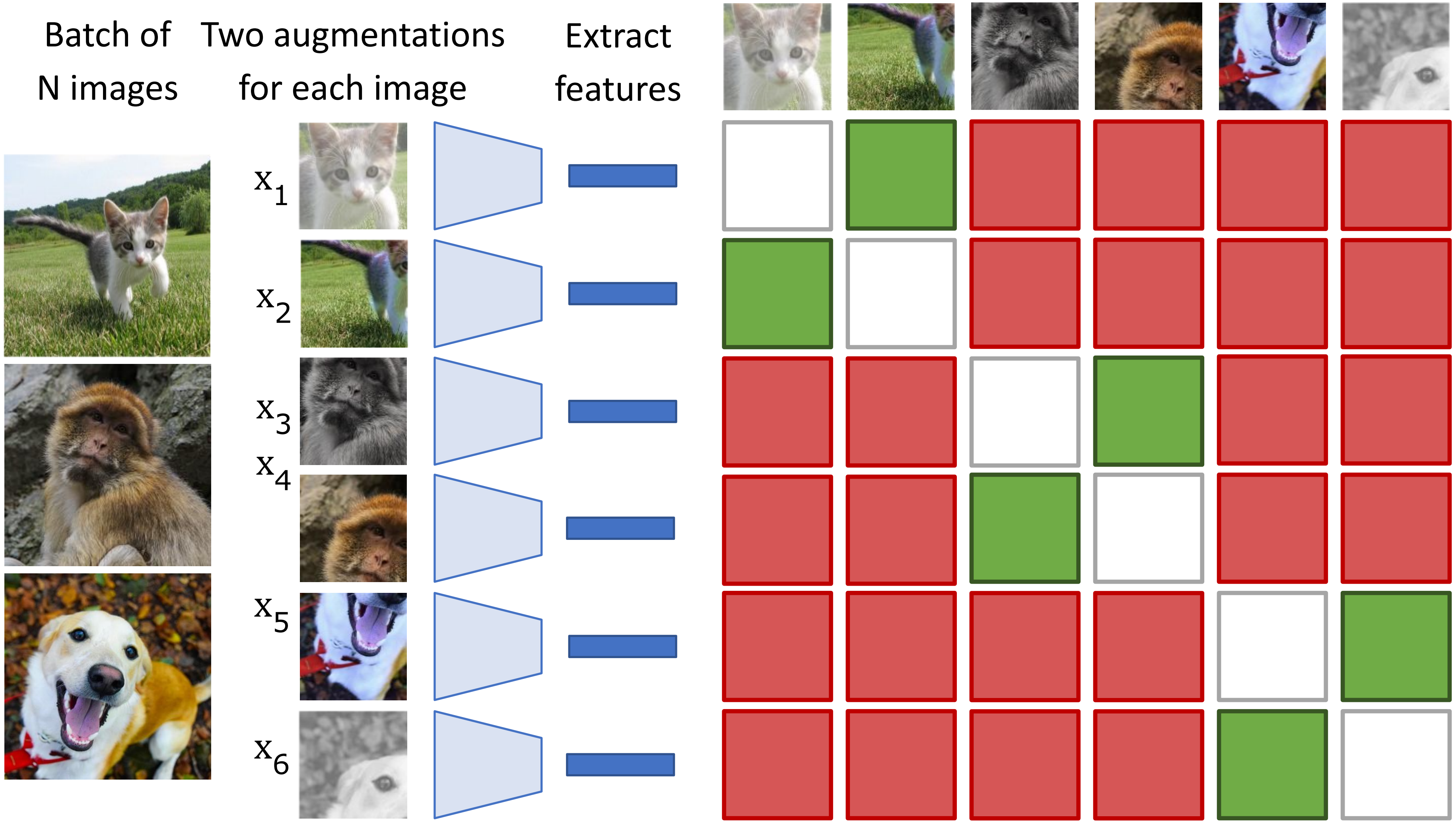
Hadsell et al, "Dimensionality Reduction by Learning and Invariant Mapping", CVPR 2006
Wu et al, "Unsupervised Feature Learning by Non-Parametric Instance-Level Discrimination", CVPR 2018
Van den Oord et al, "Representation Learning with Contrastive Predictive Coding", NeurIPS 2018
Hjelm et al, "Learning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

Contrastive Learning with Data Augmentation

Batch of N images

Two augmentations for each image

Extract features



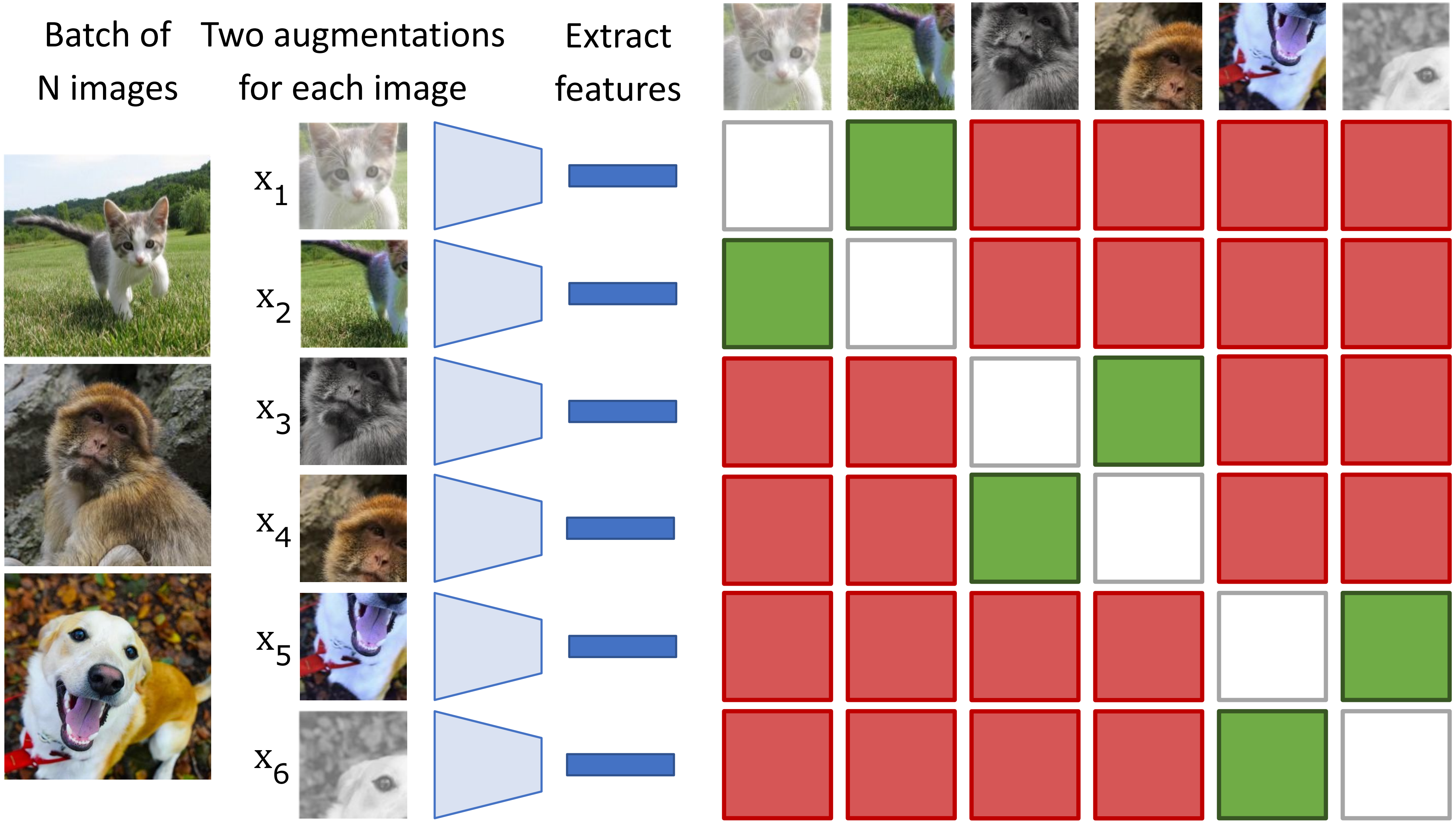
Each image tries to predict which of the *other* 2N-1 images came from the same original image

Similarity between x_i and x_j:

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

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

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:

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

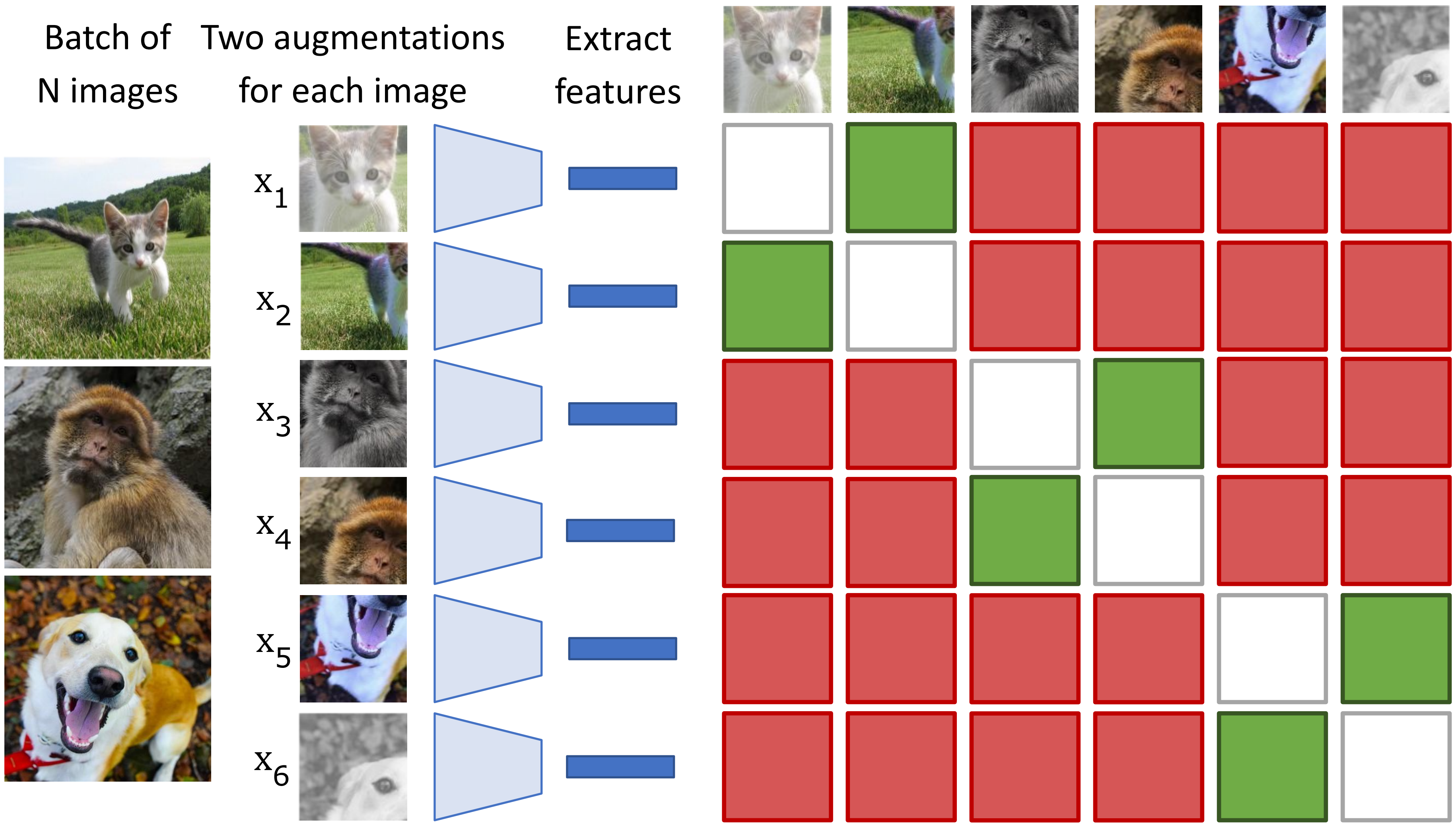
If (x_i, x_j) is a positive pair, then loss for x_i is:

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

(τ is a *temperature*)

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

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!

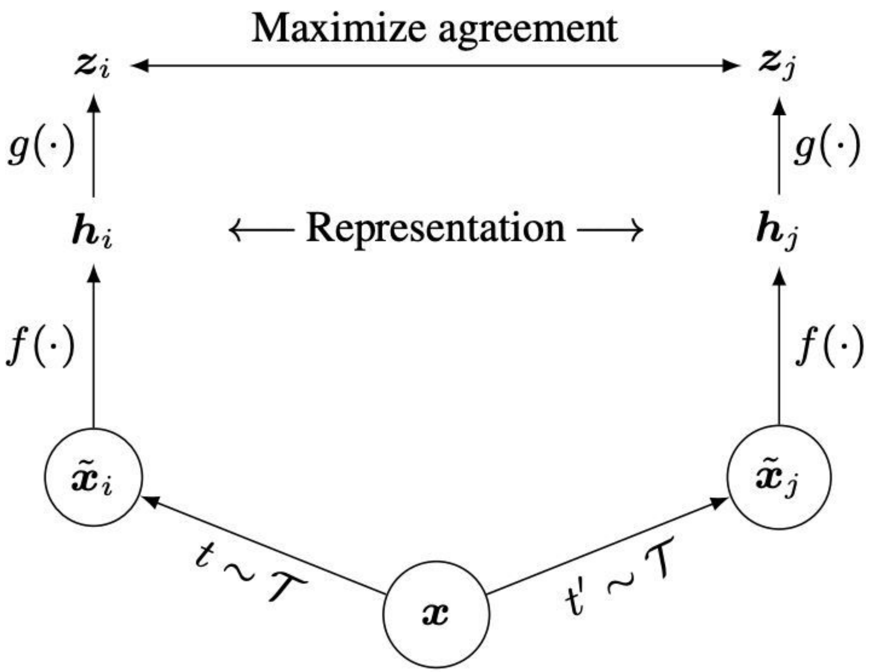
Hadsell et al, "Dimensionality Reduction by Learning and Invariant Mapping", CVPR 2006
Wu et al, "Unsupervised Feature Learning by Non-Parametric Instance-Level Discrimination", CVPR 2018
Van den Oord et al, "Representation Learning with Contrastive Predictive Coding", NeurIPS 2018
Hjelm et al, "Learning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

SimCLR: A Simple Framework for Contrastive Learning

- **SimCLR** [Chen et al., 2020]
  - A **simple** framework for contrastive learning without requiring specialized architectures or a memory bank

  - This paper founds that contrastive learning benefits from …

1. Strong augmentation (i.e., composition of multiple data augmentation operations)
2. A nonlinear MLP between the representation and the contrastive loss
3. Large batch sizes and longer training



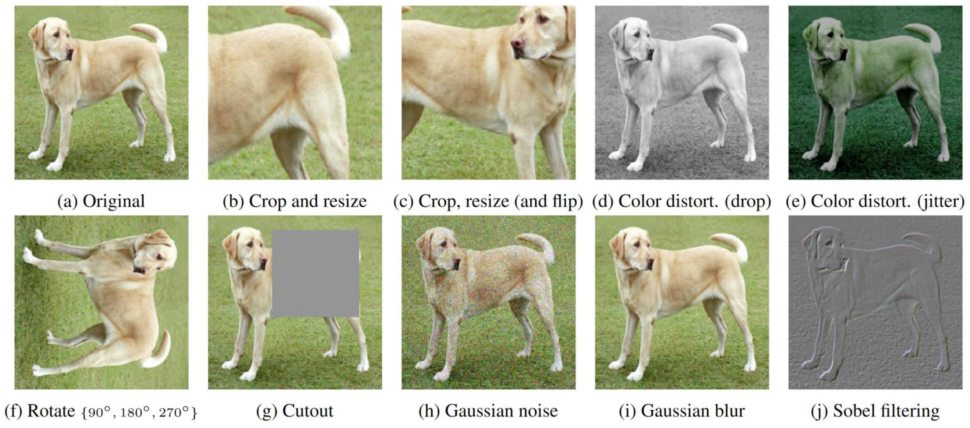
* source : https://ai.googleblog.com/2020/04/advancing-self-supervised-and-semi.html 86

SimCLR: A Simple Framework for Contrastive Learning

- **SimCLR** [Chen et al., 2020]
  - A **simple** framework for contrastive learning without requiring specialized architectures or a memory bank

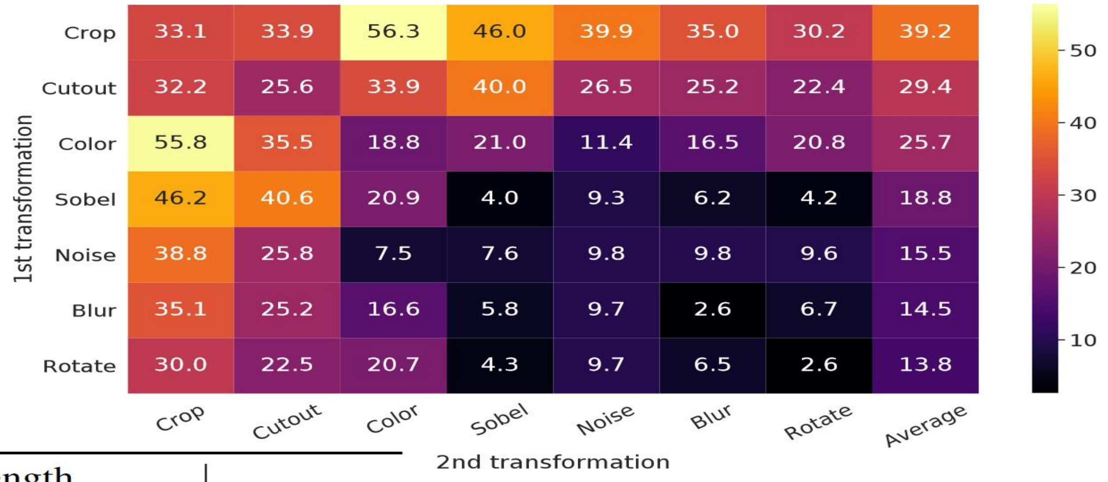
  - This paper founds that contrastive learning benefits from …

1. Strong augmentation (i.e., composition of multiple data augmentation operations)
   - Strong color distortion degrades supervised learning, but improves SimCLR
   - A stronger augmentation (AutoAugment) degrades SimCLR



(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



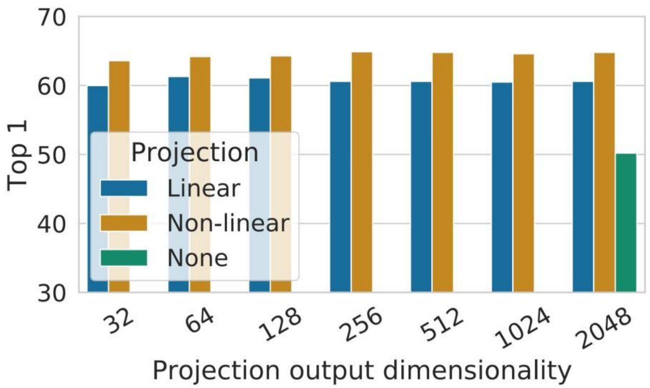
Methods	Color distortion strength 1/8	1/4	1/2	1	1 (+Blur)	AutoAug
SimCLR	59.6	61.0	62.6	63.2	64.5	61.1
Supervised	77.0	76.7	76.5	75.7	75.4	77.1

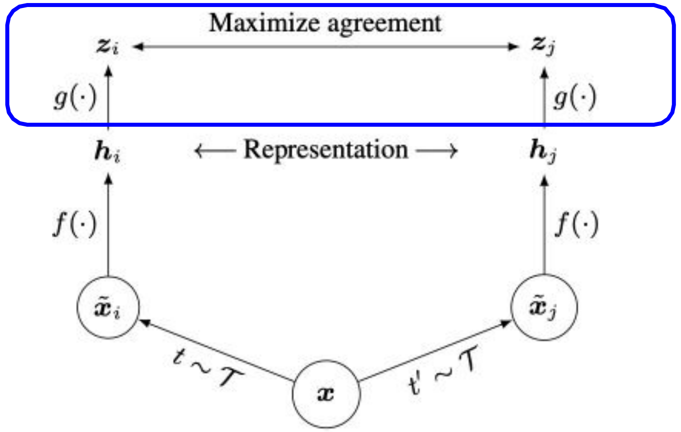
SimCLR: A Simple Framework for Contrastive Learning

- **SimCLR** [Chen et al., 2020]
  - A **simple** framework for contrastive learning without requiring specialized architectures or a memory bank

  - This paper founds that contrastive learning benefits from …

2. A nonlinear MLP between the representation and the contrastive loss
    -





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

SimCLR: A Simple Framework for Contrastive Learning

- **SimCLR** [Chen et al., 2020]
  - A **simple** framework for contrastive learning without requiring specialized architectures or a memory bank

  - This paper founds that contrastive learning benefits from …

3. Large batch sizes and longer training

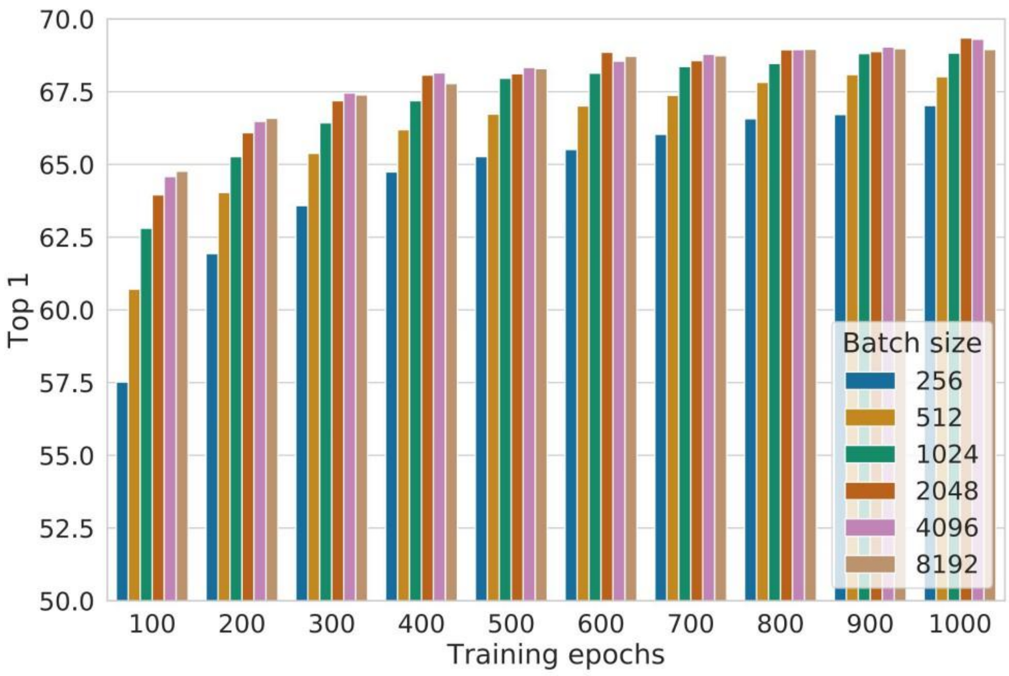


Figure 9. Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch.[10]

Large training batch size is crucial for SimCLR!

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

SSL via Invariance

- **SimCLR** [Chen et al., 2020]
  - A **simple** framework for contrastive learning without requiring specialized architectures or a memory bank
  - SimCLR achieves outstanding performance in various downstream tasks

Fine-grained image classification tasks

	Food	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Linear evaluation:												
SimCLR (ours)	**76.9**	**95.3**	80.2	48.4	**65.9**	60.0	61.2	**84.2**	**78.9**	89.2	**93.9**	**95.0**
Supervised	75.2	**95.7**	**81.2**	**56.4**	64.9	**68.8**	**63.8**	83.8	**78.7**	**92.3**	**94.1**	94.2
Fine-tuned:												
SimCLR (ours)	**89.4**	**98.6**	**89.0**	**78.2**	**68.1**	**92.1**	**87.0**	**86.6**	**77.8**	92.1	**94.1**	97.6
Supervised	88.7	98.3	**88.7**	**77.8**	67.0	91.4	**88.0**	86.5	**78.8**	**93.2**	**94.2**	**98.0**
Random init	88.3	96.0	81.9	**77.0**	53.7	91.3	84.8	69.4	64.1	82.7	72.5	92.5

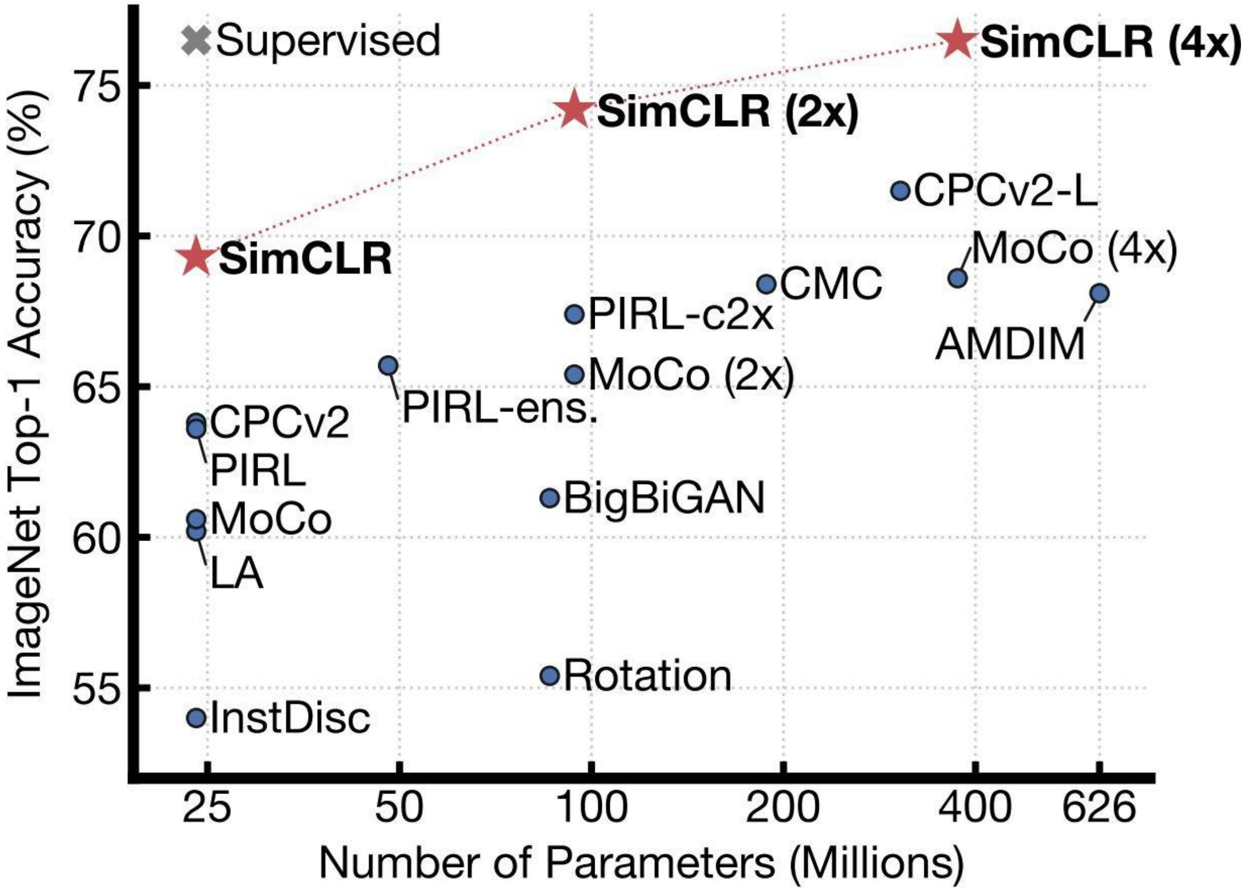
Semi-supervised learning in ImageNet

Method	Architecture	Label fraction 1% (Top 5)	Label fraction 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**

Linear evaluation in ImageNet

Method	Architecture	Param (M)	Top 1	Top 5
Methods using ResNet-50:				
Local Agg.	ResNet-50	24	60.2	-
MoCo	ResNet-50	24	60.6	-
PIRL	ResNet-50	24	63.6	-
CPC v2	ResNet-50	24	63.8	85.3
SimCLR (ours)	ResNet-50	24	**69.3**	**89.0**
Methods using other architectures:				
Rotation	RevNet-50 (4×)	86	55.4	-
BigBiGAN	RevNet-50 (4×)	86	61.3	81.9
AMDIM	Custom-ResNet	626	68.1	-
CMC	ResNet-50 (2×)	188	68.4	88.2
MoCo	ResNet-50 (4×)	375	68.6	-
CPC v2	ResNet-161 (∗)	305	71.5	90.1
SimCLR (ours)	ResNet-50 (2×)	94	74.2	92.0
SimCLR (ours)	ResNet-50 (4×)	375	**76.5**	**93.2**

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% Top 5	Label fraction 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.

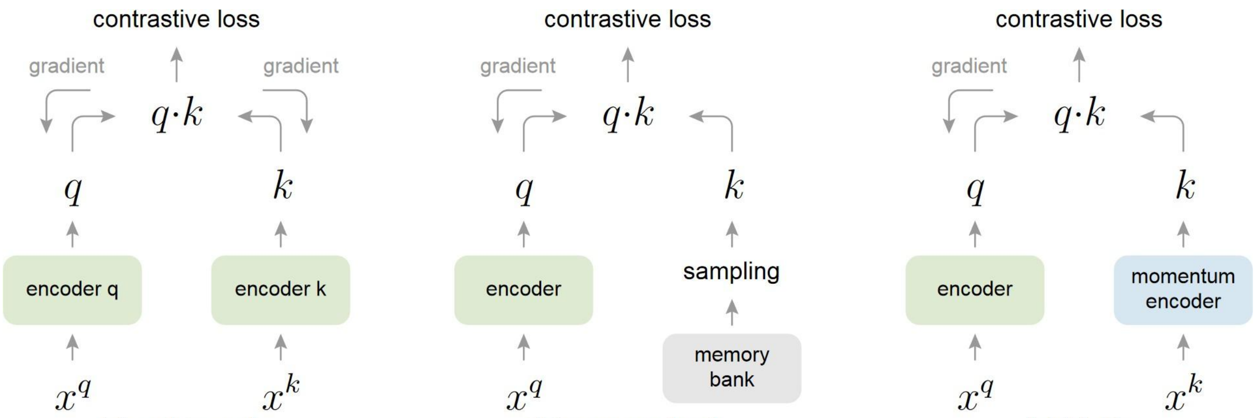
Finetune the encoder with 1% / 10% of labeled data on ImageNet.

Source: Chen et al., 2020

Momentum Contrastive Learning (MoCo)

- **Momentum Contrast (MoCo)** [He et al., 2019]
  - **Key issue: the number of negatives** is very crucial in contrastive learning
  - How to resolve this issue in prior works? **Memory Bank**
    - Note: representations in the memory bank are momentum-updated

  - **MoCo's idea:** use a **momentum-updated encoder** and maintain a **queue**



(a) end-to-end (b) memory bank (c) MoCo

- **Momentum encoder** increases the key representations' consistency
- **Queue** allows us to use recent and many negative samples

Momentum Contrastive Learning (MoCo)

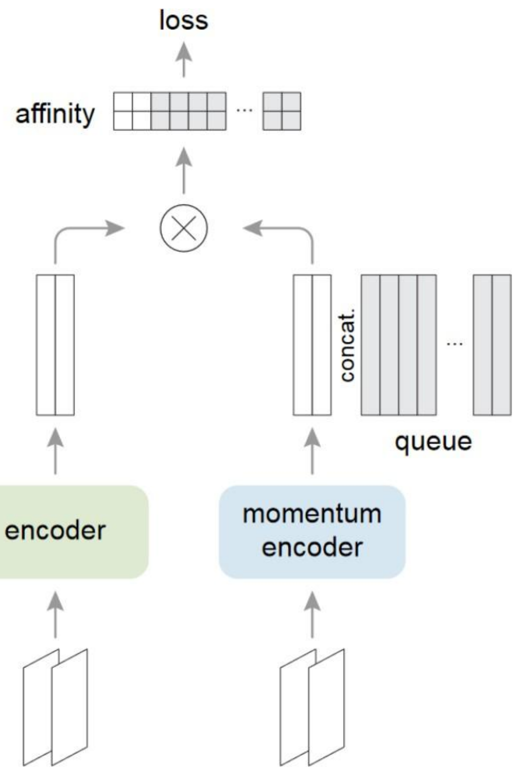
- **Momentum Contrast (MoCo)** [He et al., 2019]
  - **Key issue: the number of negatives** is very crucial in contrastive learning
  - How to resolve this issue in prior works? **Memory Bank**
    - Note: representations in the memory bank are momentum-updated

  - **MoCo's idea:** use a **momentum-updated encoder** and maintain a **queue**

  - MoCo also optimizes contrastive learning objective

$$\mathcal{L}_{q,k^+,\{k^-\}} = -\log \frac{\exp(q \cdot k^+/\tau)}{\exp(q \cdot k^+/\tau) + \sum_{k^-} \exp(q \cdot k^-/\tau)}$$

Randomly augmented samples →



Momentum Contrastive Learning (MoCo)

- **Momentum Contrast (MoCo)** [He et al., 2019]
  - **Key issue: the number of negatives** is very crucial in contrastive learning
  - How to resolve this issue in prior works? **Memory Bank**
    - Note: representations in the memory bank are momentum-updated

  - **MoCo's idea:** use a **momentum-updated encoder** and maintain a **queue**

  - MoCo also optimizes contrastive learning objective

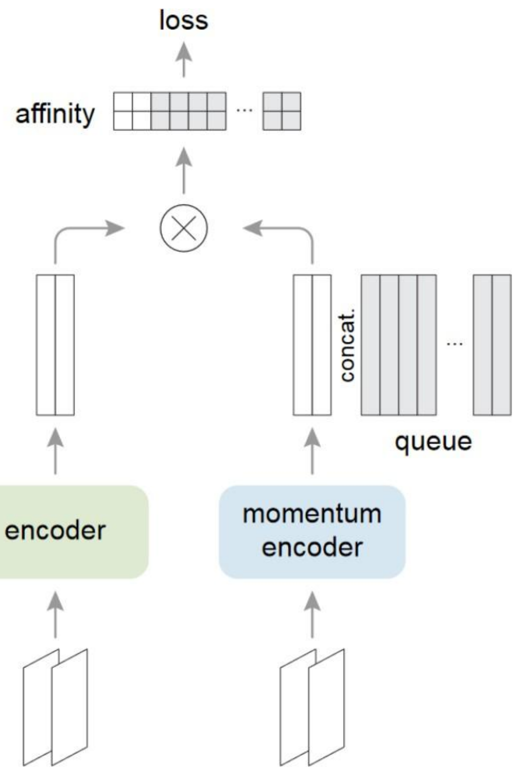
$$\mathcal{L}_{q,k^+,\{k^-\}} = -\log \frac{\exp(q \cdot k^+/\tau)}{\exp(q \cdot k^+/\tau) + \sum_{k^-} \exp(q \cdot k^-/\tau)}$$

  - After encoder is updated,
    - Momentum encoder is updated by

$$\theta_{\texttt{momentum}} \leftarrow m\theta_{\texttt{momentum}} + (1-m)\theta$$

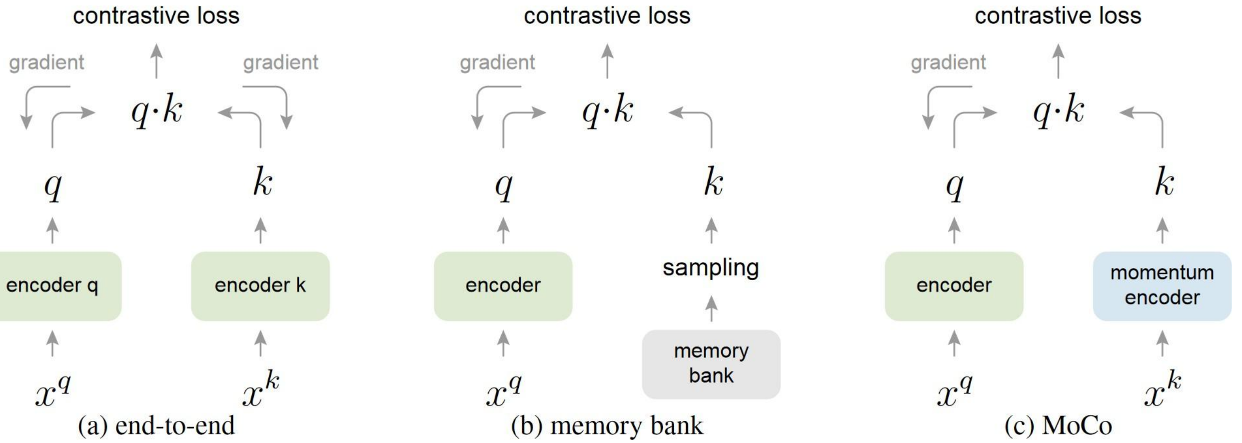
    - Add the current positive keys k^+ into the queue

Randomly augmented samples →



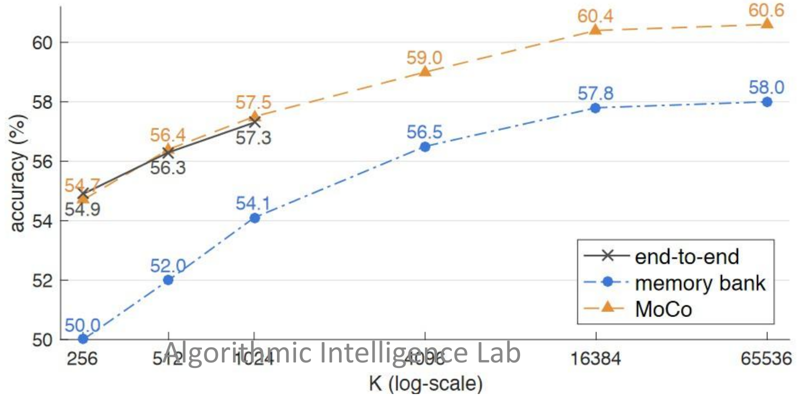
Momentum Contrastive Learning (MoCo)

- **Momentum Contrast (MoCo)** [He et al., 2019]
  - **MoCo's idea:** use a **momentum-updated encoder** and maintain a **queue**

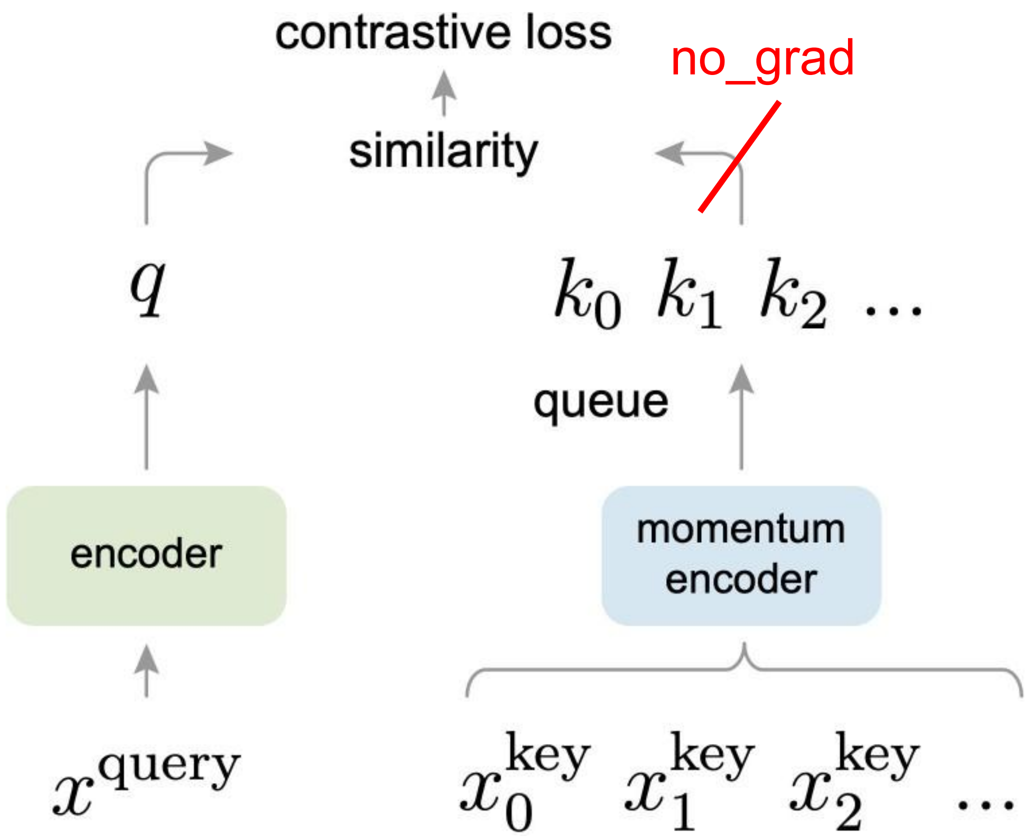


  - **Momentum encoder** increases the key representations' consistency
  - **Queue** allows us to use recent and many negative samples

momentum m	0	0.9	0.99	0.999	0.9999
accuracy (%)	*fail*	55.2	57.8	59.0	58.9



Momentum Contrastive Learning (MoCo)

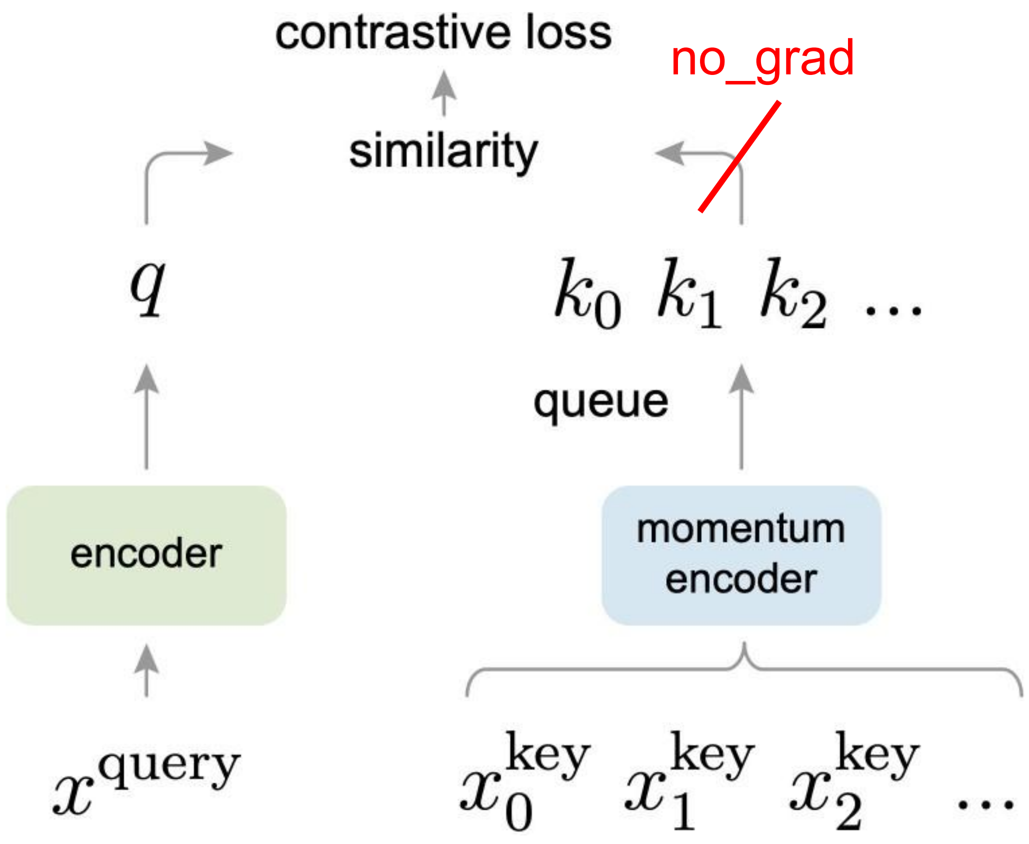


Key differences to SimCLR:

- 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.

Source: He 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:

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

Source: He et al., 2020

MoCo

Algorithm 1 Pseudocode of MoCo in a PyTorch-like style.

```
# f_q, f_k: encoder networks for query and key
# queue: dictionary as a queue of K keys (CxK)
# m: momentum
# t: temperature

f_k.params = f_q.params # initialize
for x in loader: # load a minibatch x with N samples
   x_q = aug(x) # a randomly augmented version
   x_k = aug(x) # another randomly augmented version

   q = f_q.forward(x_q) # queries: NxC
   k = f_k.forward(x_k) # keys: NxC
   k = k.detach() # no gradient to keys

   # positive logits: Nx1
   l_pos = bmm(q.view(N,1,C), k.view(N,C,1))

   # negative logits: NxK
   l_neg = mm(q.view(N,C), queue.view(C,K))

   # logits: Nx(1+K)
   logits = cat([l_pos, l_neg], dim=1)

   # contrastive loss, Eqn.(1)
   labels = zeros(N) # positives are the 0-th
   loss = CrossEntropyLoss(logits/t, labels)

   # SGD update: query network
   loss.backward()
   update(f_q.params)

   # momentum update: key network
   f_k.params = m*f_k.params+(1-m)*f_q.params

   # update dictionary
   enqueue(queue, k) # enqueue the current minibatch
   dequeue(queue) # dequeue the earliest minibatch
```

bmm: batch matrix multiplication; mm: matrix multiplication; cat: concatenation.

Generate a positive pair by sampling data augmentation functions

No gradient through the positive sample

Use the running queue of keys as the negative samples

InfoNCE loss

Update f_k through momentum

Update the FIFO negative sample queue

Source: He et al., 2020

“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]

MoCo vs. SimCLR vs. MoCo V2

case	unsup. pre-train				ImageNet	VOC detection		
	MLP	aug+	cos	epochs	acc.	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]

MoCo vs. SimCLR 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

MoCo vs. SimCLR vs. MoCo V2

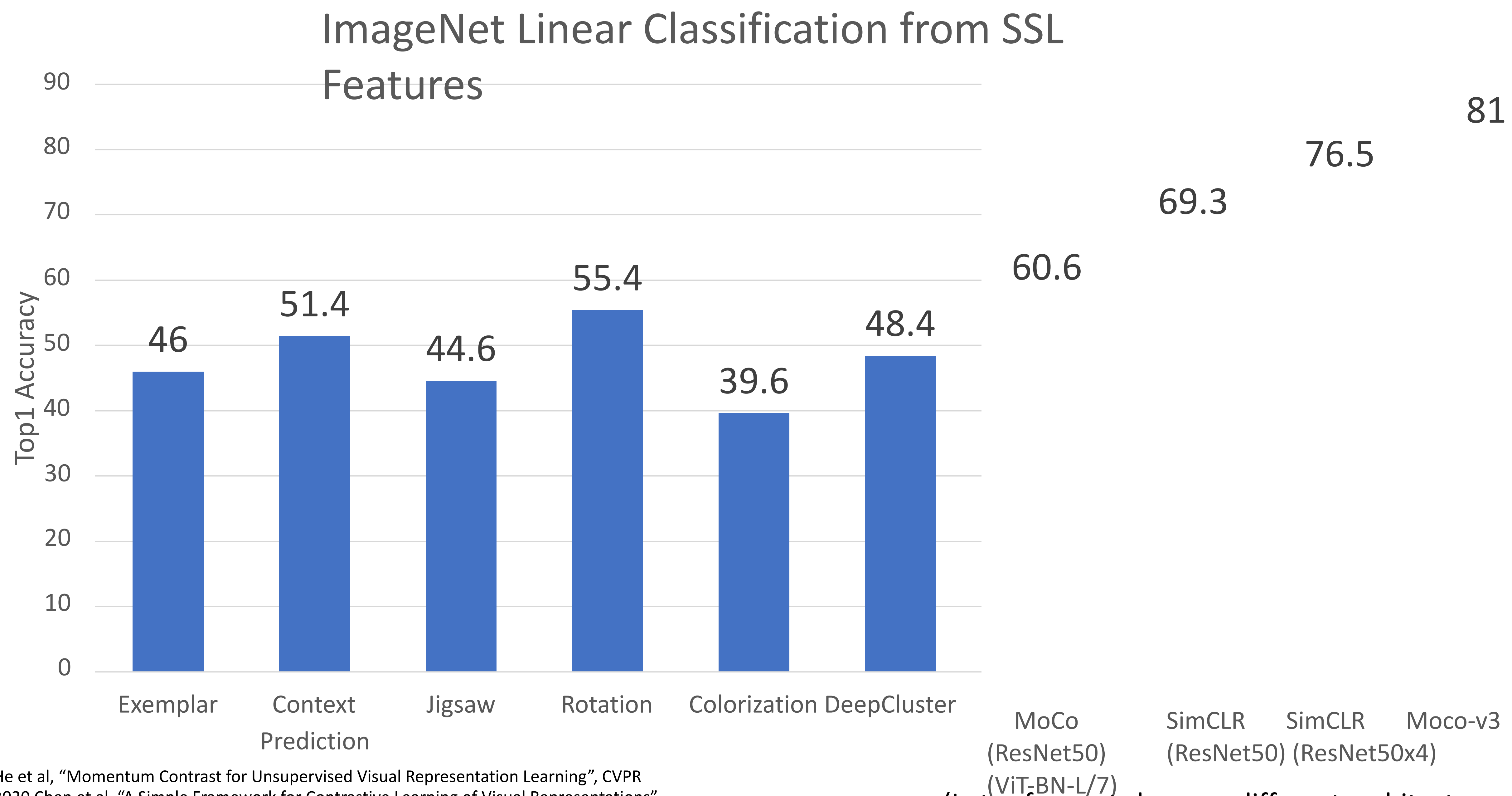
mechanism	batch	memory / GPU	time / 200-ep.
MoCo	256	**5.0G**	**53 hrs**
end-to-end	256	7.4G	65 hrs
end-to-end	4096	93.0G†	n/a

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)

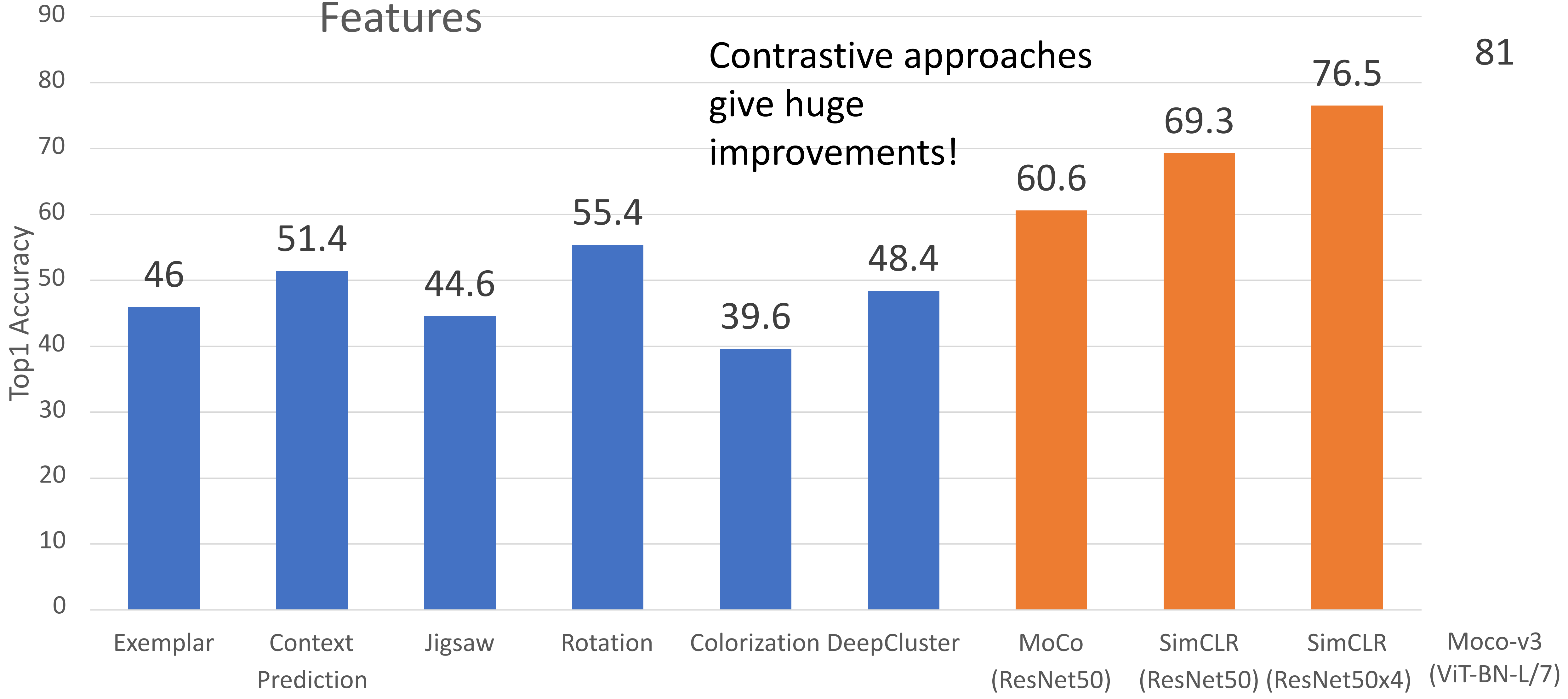
Source: Chen et al., 2020



(Lots of caveats here … different architectures, etc)

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

ImageNet Linear Classification from SSL Features

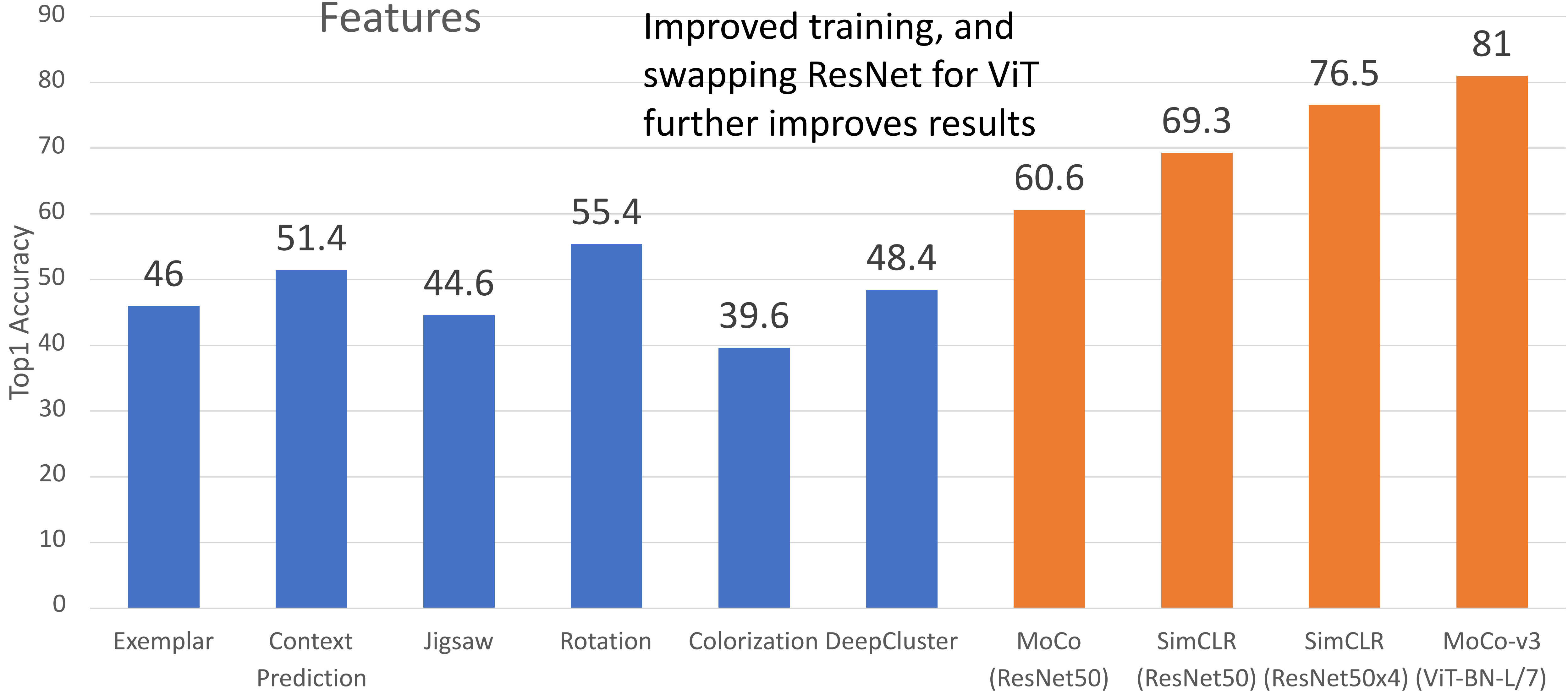


Contrastive approaches give huge improvements!

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

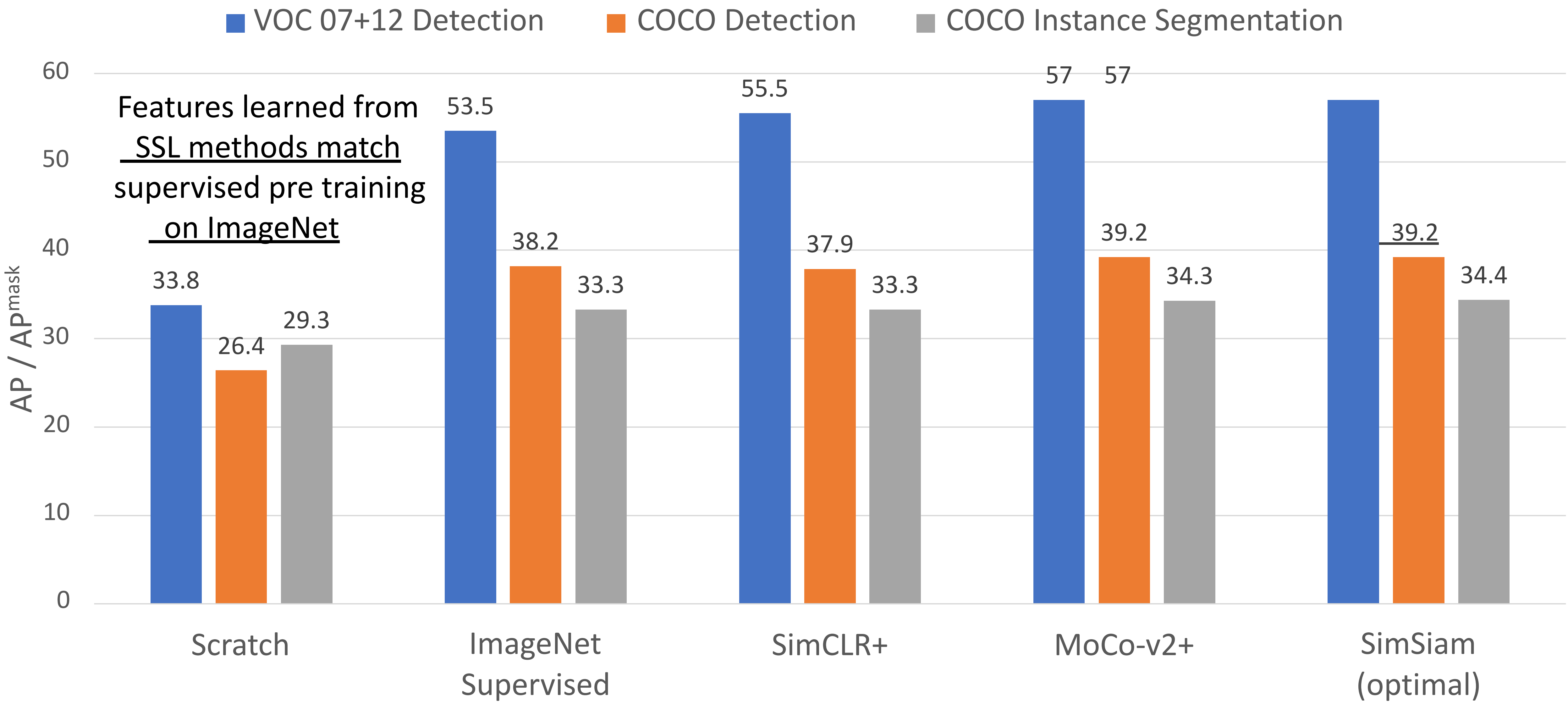
(Lots of caveats here … different architectures, etc)

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

Contrastive SSL Pre-training then Fine-tuning on Detection



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

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

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

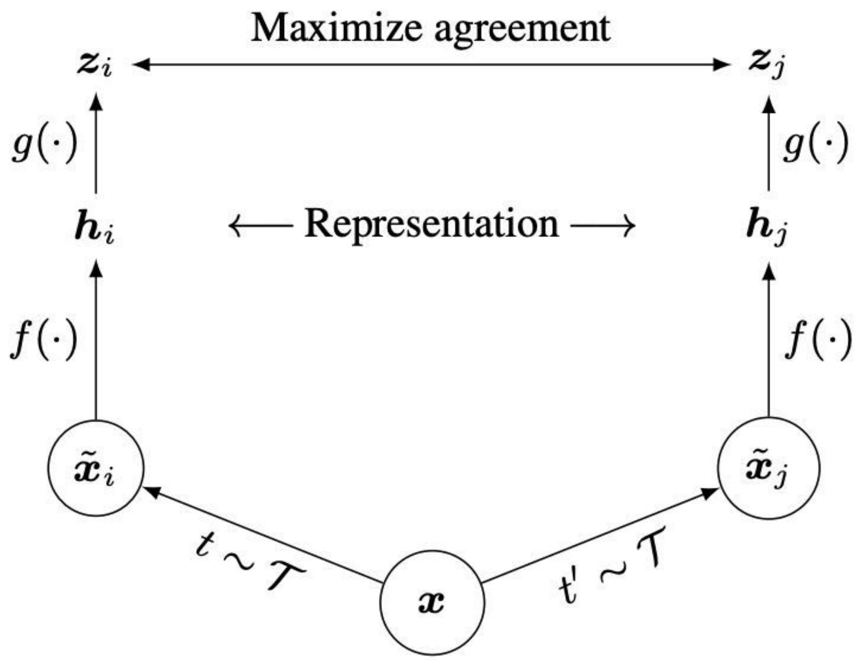
Commonly known as the InfoNCE loss (van den Oord et al., 2018)
A *lower bound* on the mutual information between *f(x)* and *f(x⁺)*

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

Summary: Contrastive Representation Learning

SimCLR: a simple framework for contrastive representation learning

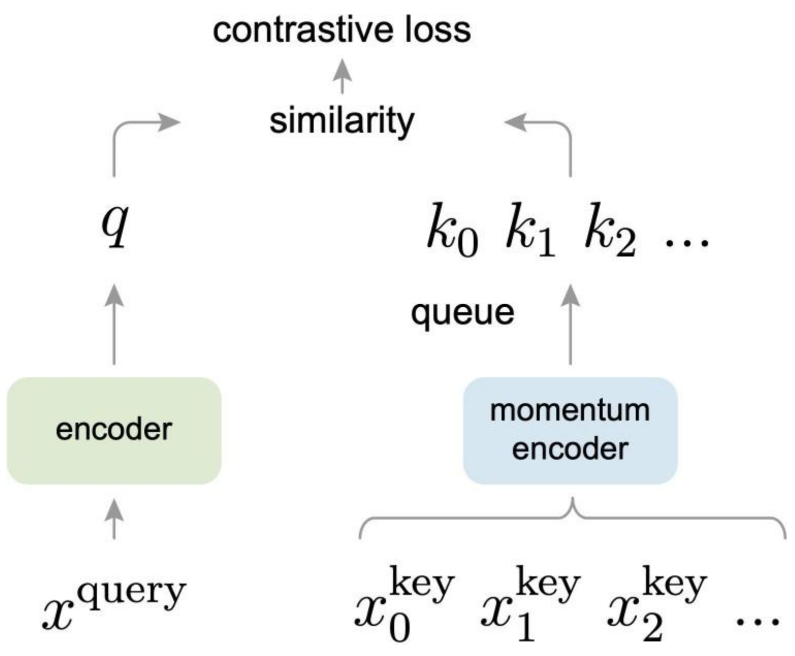
- Key ideas: non-linear projection head to allow flexible representation learning
- Simple to implement, effective in learning visual representation
- Requires large training batch size to be effective; large memory footprint



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



Summary: Contrastive Representation Learning

- **Limitations** in contrastive learning (with negatives)
  - It is sensitive to the number of negative $\Rightarrow$ a large batch size or a queue is required
  - Are all the different instances negative?

Positive pair $f($  $)$ $\approx$ $f($  $)$

Negative pair $f($  $)$ $\neq$ $f($  $)$

This relation might be not true

- **Q)** can we learn representations without negative samples?
- Simply minimizing $\|f($  $) - f($  $)\|$ leads to mode collapse, i.e., $\forall x, f(x) = c$

- **Next:** Positive-only approaches

"Self" Distillation

- What we want $f_\theta(I) = f_\theta(\mathrm{augment}(I))$
- How we do it $f_\theta^{\mathrm{student}}(I) = f_\theta^{\mathrm{teacher}}(\mathrm{augment}(I))$
- Prevent trivial solutions by asymmetry
  - Asymmetric learning rule between student teacher
  - Asymmetric architecture between student teacher

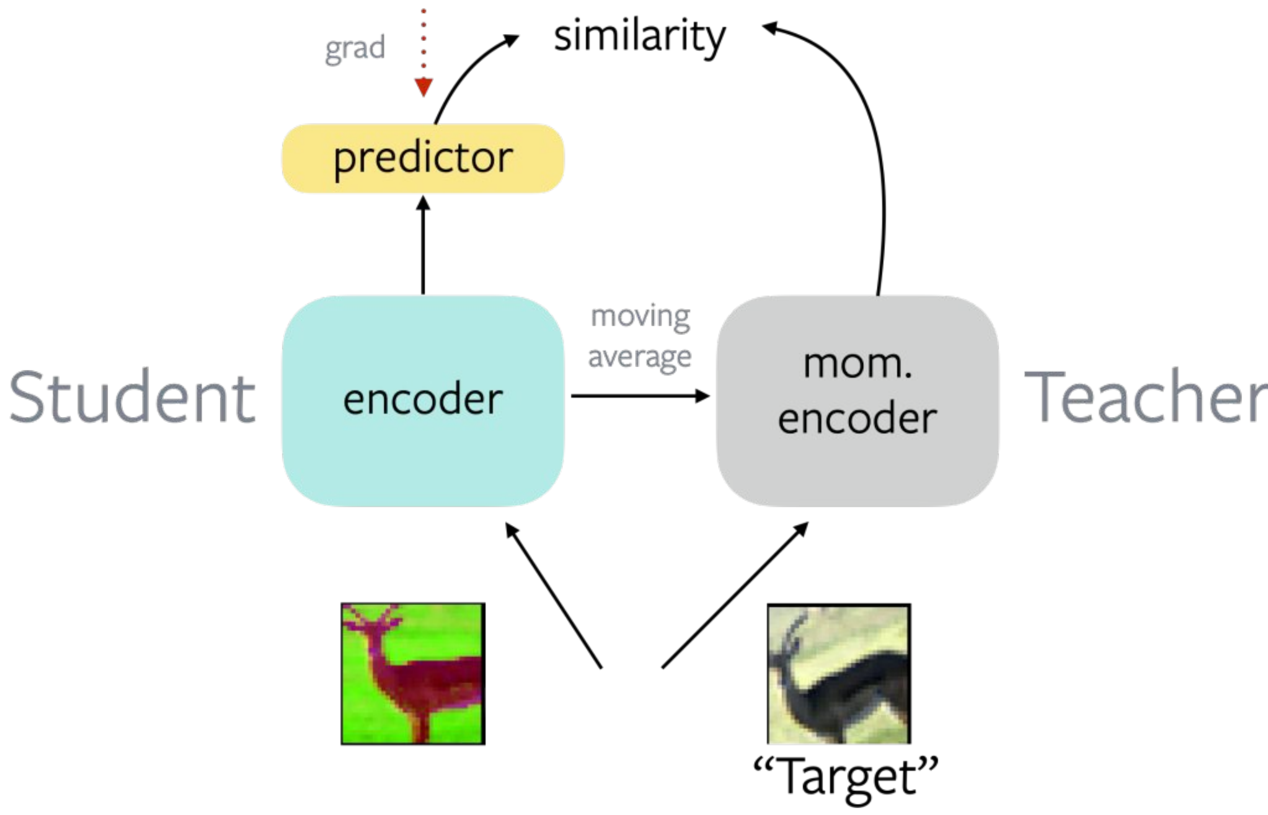
- **Bootstrap You Own Latent (BYOL)** [Grill et al., 2020]
  - **Idea:** directly bootstrap the representations

- What we want

$$f_\theta(I) = f_\theta(\text{augment}(I))$$

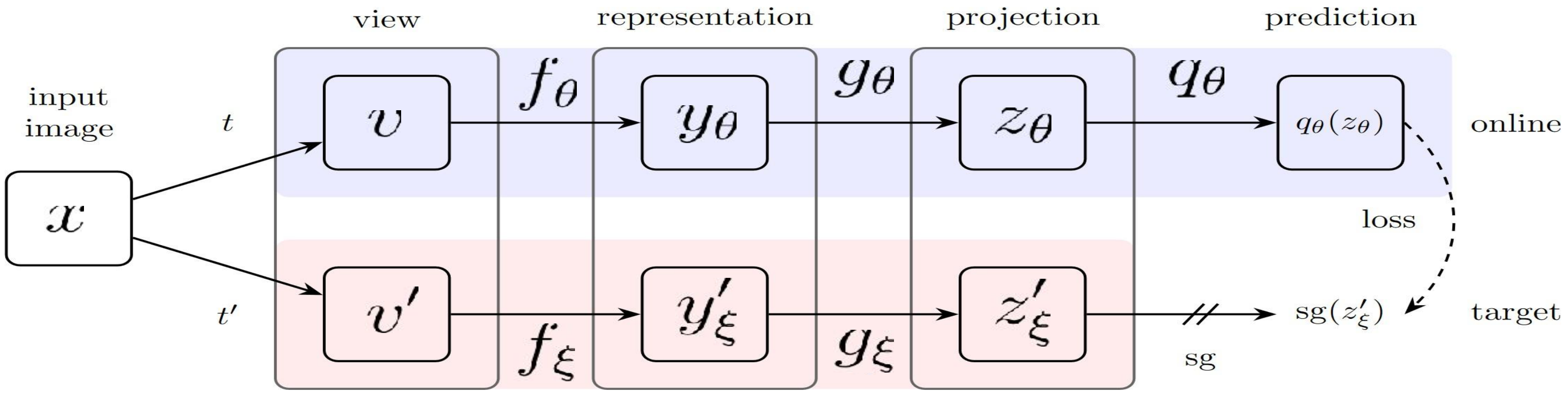
- How we do it

$$f_\theta^{\text{student}}(I) = f_\theta^{\text{teacher}}(\text{augment}(I))$$



BYOL - Grill et al., 2020

- **Bootstrap You Own Latent (BYOL)** [Grill et al., 2020]
  - **Idea:** directly bootstrap the representations



Objective

$$\mathcal{L}_{\text{BYOL}} = \left\| \frac{q_\theta(z_\theta)}{\|q_\theta(z_\theta)\|} - \frac{z'_\xi}{\|z'_\xi\|} \right\|^2$$

Update

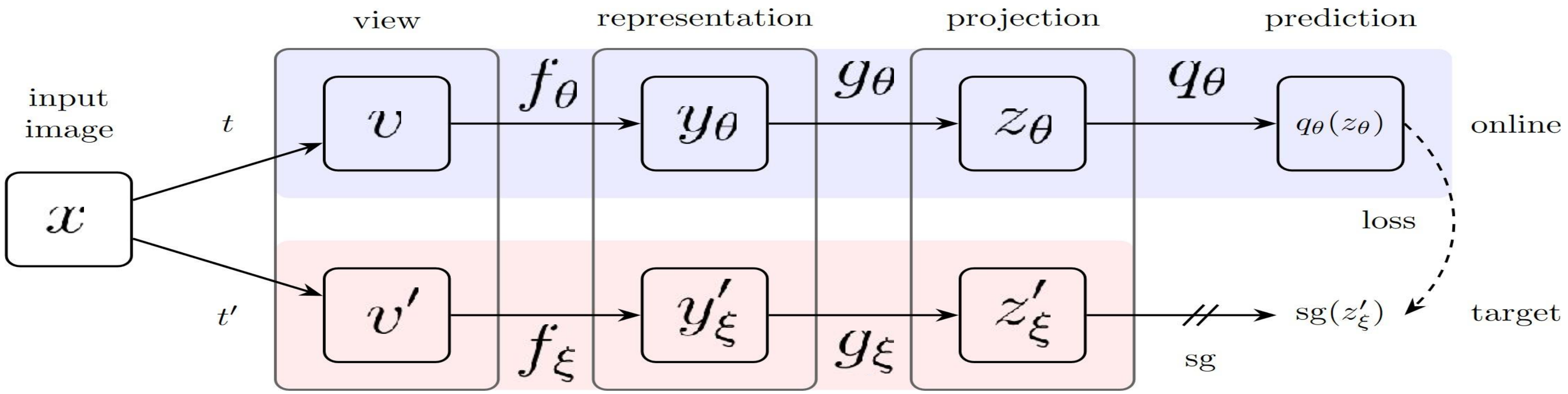
$$\theta \leftarrow \text{optimizer}(\theta, \nabla_\theta \mathcal{L}_{\text{BYOL}})$$

$$\xi \leftarrow \tau\xi + (1-\tau)\theta$$

- **Key components:** target (momentum) network, predictor, stop-gradient (sg)

* source : [Grill et al., 2020] 25

- **Bootstrap You Own Latent (BYOL)** [Grill et al., 2020]
  - **Idea:** directly bootstrap the representations



Objective

$$\mathcal{L}_{\texttt{BYOL}} = \left\| \frac{q_\theta(z_\theta)}{\|q_\theta(z_\theta)\|} - \frac{z'_\xi}{\|z'_\xi\|} \right\|^2$$

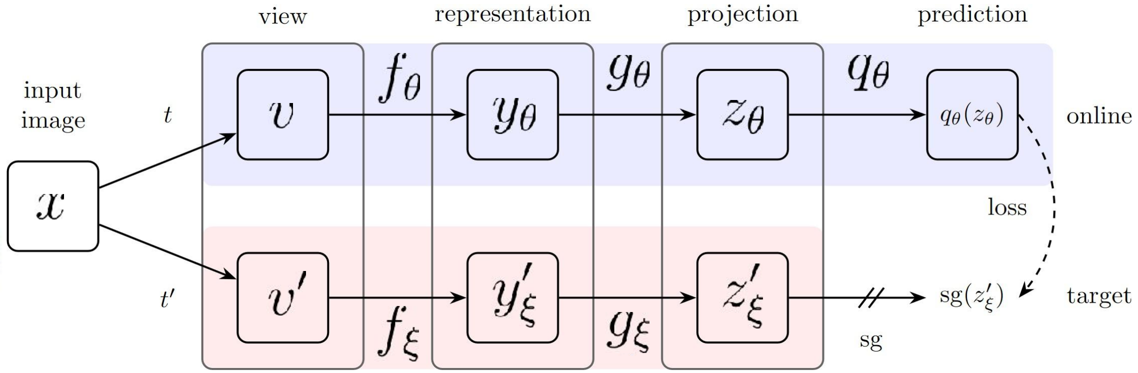
Where:

- $\mathcal{L}_{\text{BYOL}}$: The loss function that measures the difference between two normalized representations.
- $q_\theta(z_\theta)$: The predicted representation from the online network.
- z'_ξ: The target representation from the target network.
- $\|q_\theta(z_\theta)\|$: The L2 norm (magnitude) of the predicted representation.
- $\|z'_\xi\|$: The L2 norm (magnitude) of the target representation.
- $\frac{q_\theta(z_\theta)}{\|q_\theta(z_\theta)\|}$: The L2-normalized predicted representation.
- $\frac{z'_\xi}{\|z'_\xi\|}$: The L2-normalized target representation.
- $\|\cdot\|^2$: The squared Euclidean distance between the two normalized vectors.

Update

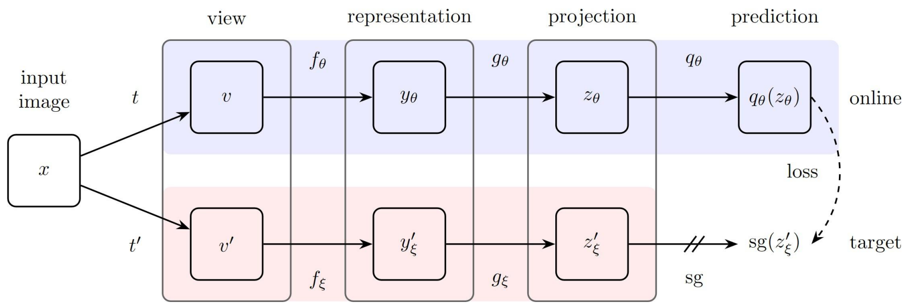
$$\theta \leftarrow \text{optimizer}(\theta, \nabla_\theta \mathcal{L}_{\texttt{BYOL}})$$
$$\xi \leftarrow \tau\xi + (1-\tau)\theta$$

1. **Two Views of an Image**:
    - Given an image, BYOL applies two different **augmentations** to create two different views of the same image.
    - One view is processed by an **online network** (with parameters θ).
    - The other view is processed by a **target network** (with parameters ξ).
2. **Feature Extraction**:
    - z_θ is the representation of the first view, extracted by the online encoder.
    - z'_ξ is the representation of the second view, extracted by the target encoder.
3. **Projection & Prediction**:
    - The online network has an extra **predictor network** q_θ, which maps z_θ to a predicted representation $q_\theta(z_\theta)$.
    - The target network **does not** have a predictor; it directly outputs z'_ξ.
4. **Normalization**:
    - Both $q_\theta(z_\theta)$ and z'_ξ are **L2-normalized**, meaning they are converted to unit vectors.
    - This ensures that their magnitudes do not affect the loss, focusing only on **directional similarity**.
5. **Loss Computation**:
    - The loss measures the **squared Euclidean distance** (or equivalently, cosine similarity) between the predicted vector from the online network and the representation from the target network.

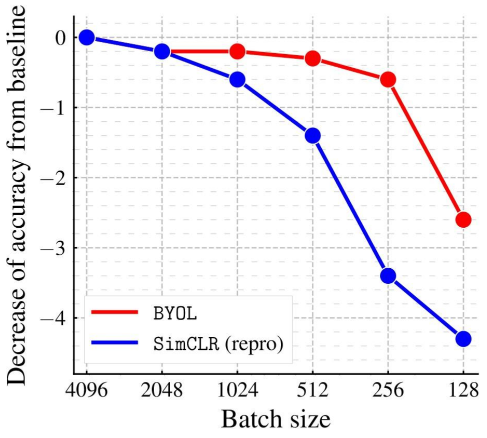


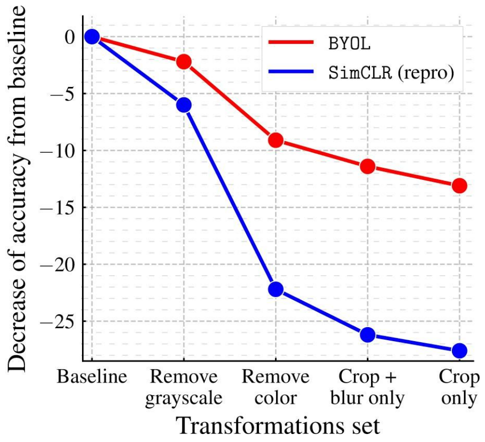
(BYOL) [Grill et al., 2020]

- **Bootstrap You Own Latent (BYOL)** [Grill et al., 2020]
    - **Idea:** directly bootstrap the representations

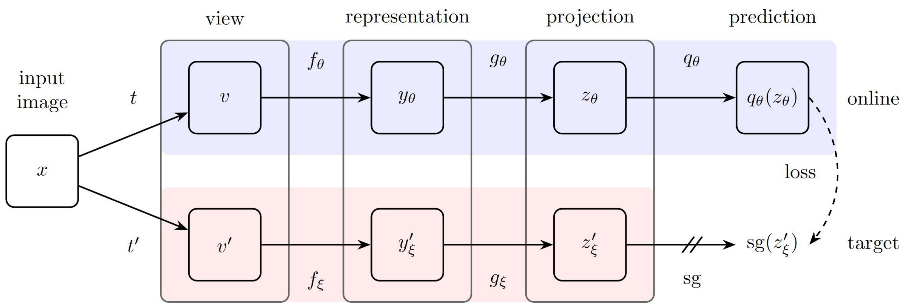


- BYOL is **more robust** to the choice of **batch sizes** and **augmentations**

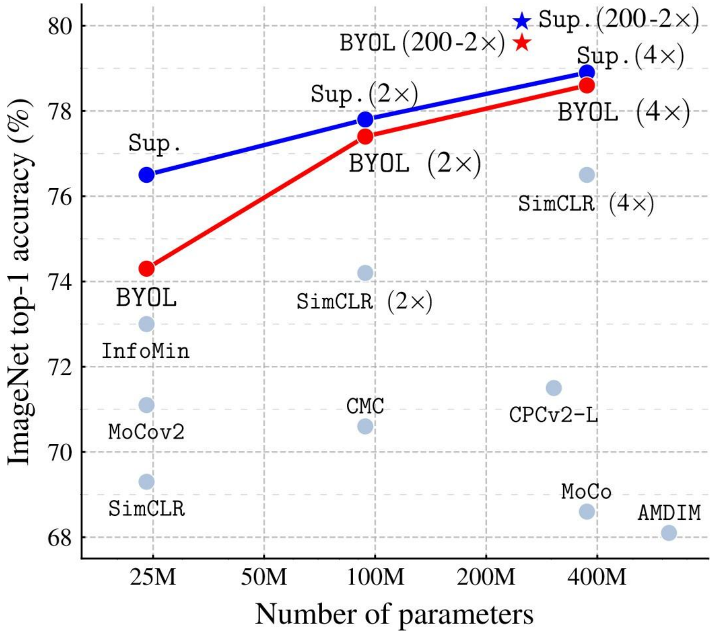




- **Bootstrap You Own Latent (BYOL)** [Grill et al., 2020]
  - **Idea:** directly bootstrap the representations



    - BYOL is **more robust** to the choice of **batch sizes** and **augmentations**
    - BYOL achieves 74.3% linear evaluation accuracy; supervised learning does 76.5%



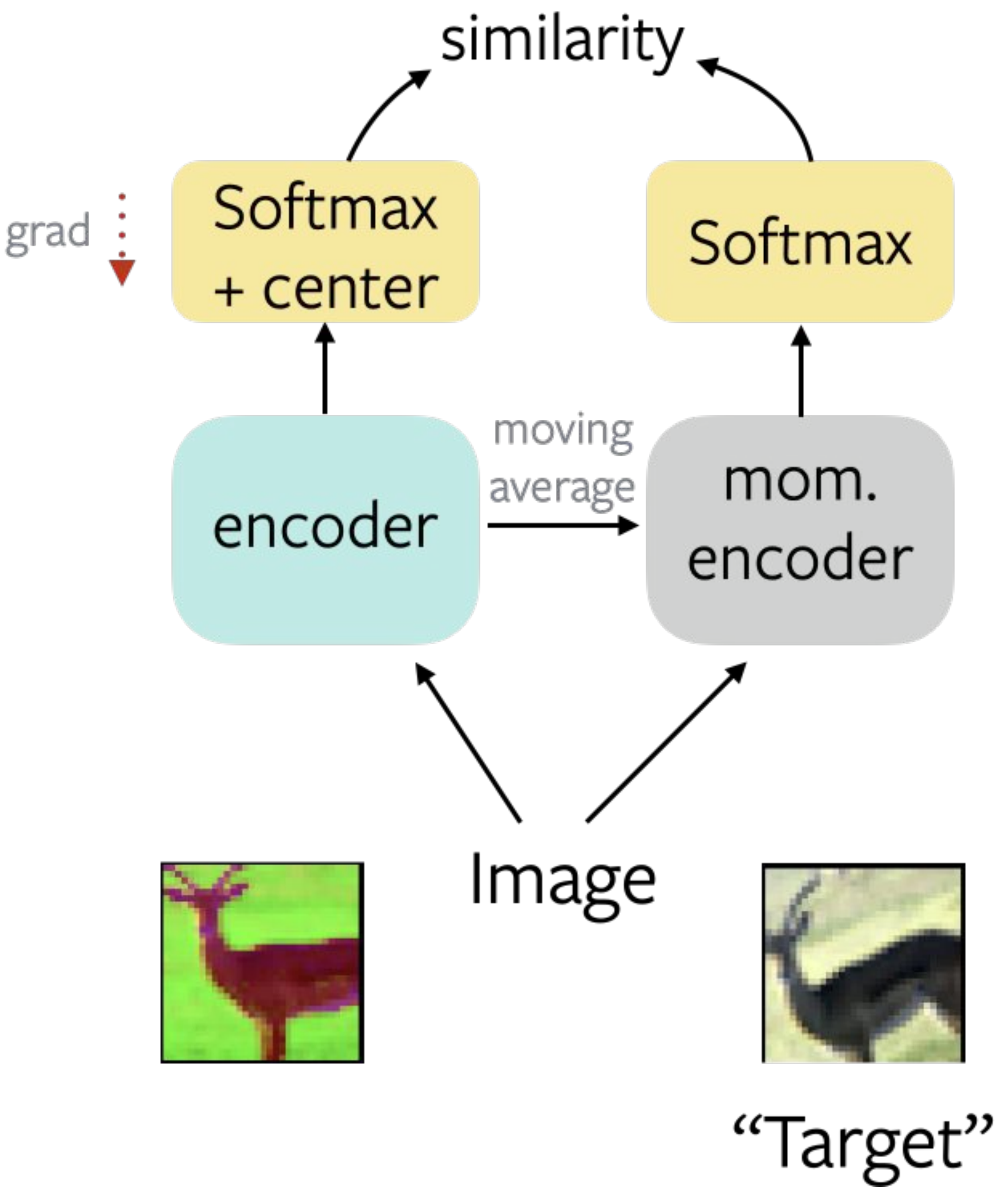
- **DINO** [Caron et al., 2021]
  - **Idea**: representation learning via self knowledge-distillation

- What we want

$$f_\theta(I) = f_\theta(\text{augment}(I))$$

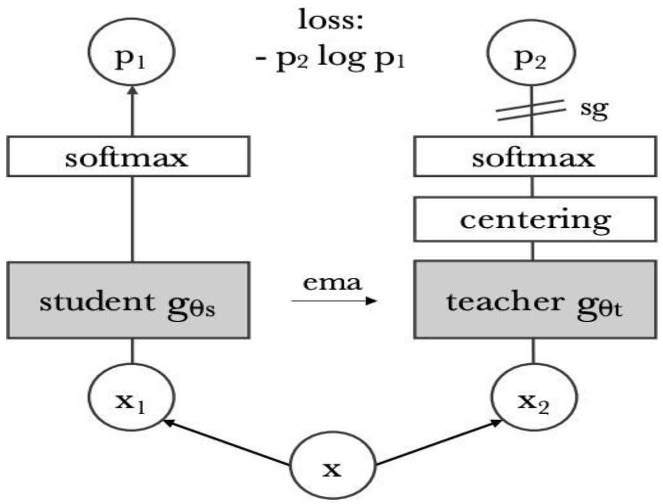
- How we do it

$$f_\theta^{\text{student}}(I) = f_\theta^{\text{teacher}}(\text{augment}(I))$$



DINO - Caron et al., 2022

- **DINO** [Caron et al., 2021]
  - **Idea**: representation learning via self knowledge-distillation



Objective

$$\mathcal{L}_{DINO} = H(P_t(x), P_s(x))$$

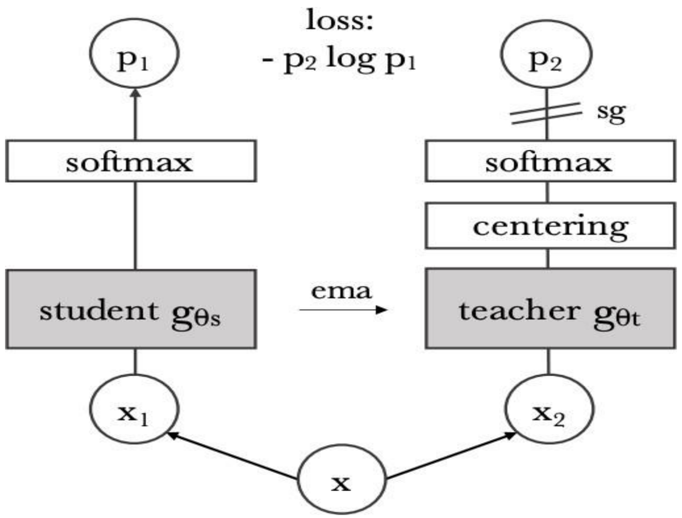
Update

$$\theta_s \leftarrow optimizer\left(\theta_s, \nabla_{\theta_s}\mathcal{L}_{DINO}\right)$$
$$\theta_t \leftarrow \lambda\theta_t + (1-\lambda)\theta_s$$

- $H(P_t(x), P_s(x))$ represents the **cross-entropy loss** between the two probability distributions $P_t(x)$ and $P_s(x)$.
- $P_t(x)$ is the output (probability distribution) of the **teacher network**.
- $P_s(x)$ is the output of the **student network**.

- **Key components:**
  - (self) knowledge-distillation
    - Distill the teacher (EMA version of a student) knowledge to the student
  - multi-crop: a strategy to generate positive views
  - centering and sharpening: a strategy to avoid collapse

- **DINO** [Caron et al., 2021]
  - **Idea**: representation learning via self knowledge-distillation



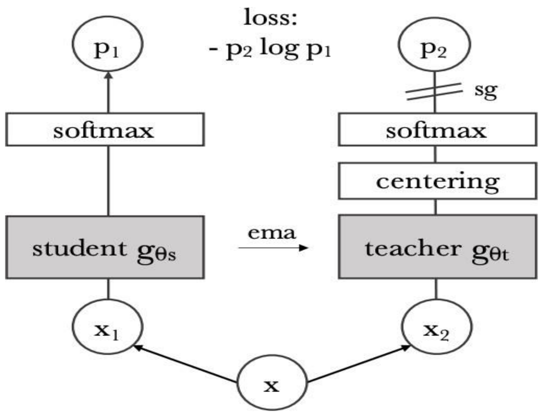
1. **Two Networks (Teacher & Student)**:
   - The **student network** learns to predict the teacher's softmax outputs.
   - The **teacher network** is a momentum-updated version of the student (like BYOL, without requiring negative samples).
2. **Output Probability Distributions**:
   - Both networks process **different augmentations** of the same image.
   - Their outputs are converted into probability distributions using a **softmax function**.
3. **Cross-Entropy Loss**:
   - The loss encourages the student to match the teacher's predictions.
   - Since the teacher network updates slowly (using an exponential moving average of the student), it provides a stable learning target.

- DINO constructs a set of views V via **multi-crop** strategy:
  - (1) global views: x_1^g, x_2^g
  - (2) local views with smaller resolution

- All crops are passed through the student; only the global views are passed through the teacher: "**local-to-global**" correspondences
  - Therefore, the loss is written as:

$$\min_{\theta_s} \sum_{x \in \{x_1^g, x_2^g\}} \sum_{\substack{x' \in V \\ x' \neq x}} H(P_t(x), P_s(x'))$$

- **DINO** [Caron et al., 2021]
  - **Idea**: representation learning via self knowledge-distillation



- DINO avoids the collapse via **centering** and **sharpening**
  - Centering: adding a bias term c to the teacher

$$g_t(x) \leftarrow g_t(x) + c$$

    - The center c is updated with an exponential moving average

$$c \leftarrow mc + (1-m)\frac{1}{B}\sum_{i=1}^{B} g_{\theta_t}(x_i)$$

  - Sharpening: using a low value for the temperature τ_t in the teacher softmax normalization

- **DINO** [Caron et al., 2021]
  - **Idea**: representation learning via self knowledge-distillation
  - **DINO** [Caron et al., 2021]
    - DINO outperforms previous contrastive methods in classification tasks
    - Self-supervised ViT features contain explicit information about the semantic segmentation of an image

Method	Arch.	Param.	im/s	Linear	k-NN
Supervised	RN50	23	1237	79.3	79.3
SCLR [12]	RN50	23	1237	69.1	60.7
MoCov2 [15]	RN50	23	1237	71.1	61.9
InfoMin [67]	RN50	23	1237	73.0	65.3
BarlowT [81]	RN50	23	1237	73.2	66.0
OBoW [27]	RN50	23	1237	73.8	61.9
BYOL [30]	RN50	23	1237	74.4	64.8
DCv2 [10]	RN50	23	1237	75.2	67.1
SwAV [10]	RN50	23	1237	**75.3**	65.7
DINO	RN50	23	1237	**75.3**	**67.5**
Supervised	ViT-S	21	1007	79.8	79.8
BYOL* [30]	ViT-S	21	1007	71.4	66.6
MoCov2* [15]	ViT-S	21	1007	72.7	64.4
SwAV* [10]	ViT-S	21	1007	73.5	66.3
DINO	ViT-S	21	1007	**77.0**	**74.5**
Comparison across architectures					
SCLR [12]	RN50w4	375	117	76.8	69.3
SwAV [10]	RN50w2	93	384	77.3	67.3
BYOL [30]	RN50w2	93	384	77.4	–
DINO	ViT-B/16	85	312	78.2	76.1
SwAV [10]	RN50w5	586	76	78.5	67.1
BYOL [30]	RN50w4	375	117	78.6	–
BYOL [30]	RN200w2	250	123	79.6	73.9
DINO	ViT-S/8	21	180	79.7	**78.3**
SCLRv2 [13]	RN152w3+SK	794	46	79.8	73.1
DINO	ViT-B/8	85	63	**80.1**	77.4

Top-1 accuracy for linear and k-NN evaluations on the validation set of ImageNet

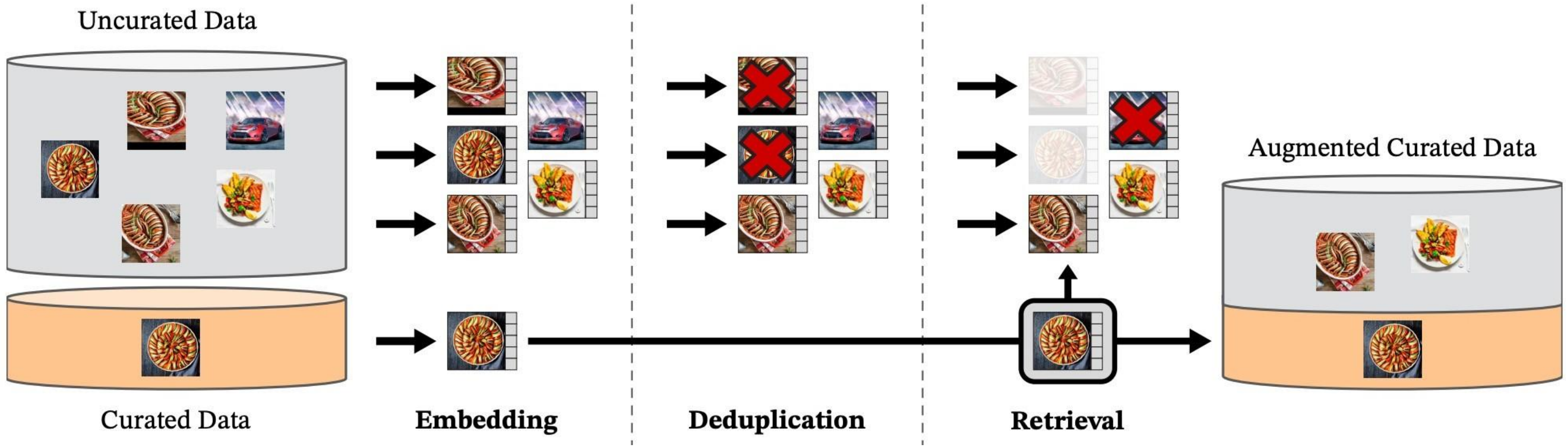
Self-attention map on [CLS] of self-supervised ViT

Method	Data	Arch.	$(\mathcal{J}\&\mathcal{F})_m$	$\mathcal{J}_m$	$\mathcal{F}_m$
Supervised					
ImageNet	INet	ViT-S/8	66.0	63.9	68.1
STM [48]	I/D/Y	RN50	81.8	79.2	84.3
Self-supervised					
CT [71]	VLOG	RN50	48.7	46.4	50.0
MAST [40]	YT-VOS	RN18	65.5	63.3	67.6
STC [37]	Kinetics	RN18	67.6	64.8	70.2
DINO	INet	ViT-S/16	61.8	60.2	63.4
DINO	INet	ViT-B/16	62.3	60.7	63.9
DINO	INet	ViT-S/8	**69.9**	**66.6**	**73.1**
DINO	INet	ViT-B/8	**71.4**	**67.9**	**74.9**

Video instance segmentation on top of self-supervised feature

DINO v2: Learning Robust Visual Features without Supervision

- **DINO v2** [Oquab et al., 2023]
  - Data preprocessing (LVD-142M dataset)
    - Curated dataset from ImageNet and fine-grained dataset
    - Uncurated dataset sourced from crawled web data
    - **Deduplication**: remove near-duplicate images to increase diversity
    - **Self-supervised image retrieval**: using ImageNet-22k pretrained ViT-H/16, retrieve relevant data from uncurated source using K-means clustering



DINO v2: Learning Robust Visual Features without Supervision

- **DINO v2** [Oquab et al., 2023]
  - Data preprocessing (LVD-142M dataset)
    - Curated dataset from ImageNet and fine-grained dataset
    - Uncurated dataset sourced from crawled web data
    - **Deduplication**: remove near-duplicate images to increase diversity
    - **Self-supervised image retrieval**: using ImageNet-22k pretrained ViT-H/16, retrieve relevant data from uncurated source using K-means clustering
  - LVD-142M maintains ImageNet-1K performance while improving in other domains

Training Data	INet-1k	Im-A	ADE-20k	Oxford-M
INet-22k	85.9	73.5	46.6	62.5
INet-22k \ INet-1k	85.3	70.3	46.2	58.7
Uncurated data	83.3	59.4	48.5	54.3
LVD-142M	85.8	73.9	47.7	64.6

DINO v2: Learning Robust Visual Features without Supervision

- **DINO v2** [Oquab et al., 2023]
  - Training method
    - Use both image-level objective in DINO and MIM objective in iBOT
    - KoLeo regularizer: minimize the differential entropy of features
      - Encourage features to be uniformly distributed

$$\mathcal{L}_{\mathrm{koleo}} = -\frac{1}{n}\sum_{i=1}^{n} \log(d_{n,i}), \text{ where } d_{n,i} = \min_{j \neq i} \|x_i - x_j\|$$

- Effect of KoLeo loss term and Masked Image Modeling from iBOT

KoLeo	INet-1k	Im-A	ADE-20k	Oxford-M
✗	85.3	70.6	47.2	55.6
✓	85.8	72.8	47.1	63.9

(a) Koleo loss

MIM	INet-1k	Im-A	ADE-20k	Oxford-M
✗	85.3	72.0	44.2	64.3
✓	85.8	72.8	47.1	63.9

(b) MIM objective in iBOT

DINO v2: Learning Robust Visual Features without Supervision

- **DINO v2** [Oquab et al., 2023]
  - DINO v2 matches domain generalization performance of CLIP
    - Linear probing experiments on ImageNet-A/R/C/Sketch

Method	Arch	Data	Im-A	Im-R	Im-C↓	Sketch
OpenCLIP	ViT-G/14	LAION	63.8	**87.8**	45.3	**66.4**
MAE	ViT-H/14	INet-1k	10.2	34.4	61.4	21.9
DINO	ViT-B/8	INet-1k	23.9	37.0	56.6	25.5
iBOT	ViT-L/16	INet-22k	41.5	51.0	43.9	38.5
DINOv2	ViT-S/14	LVD-142M	33.5	53.7	54.4	41.2
	ViT-B/14	LVD-142M	55.1	63.3	42.7	50.6
	ViT-L/14	LVD-142M	71.3	74.4	31.5	59.3
	ViT-g/14	LVD-142M	**75.9**	78.8	**28.2**	62.5

DINO v2: Learning Robust Visual Features without Supervision

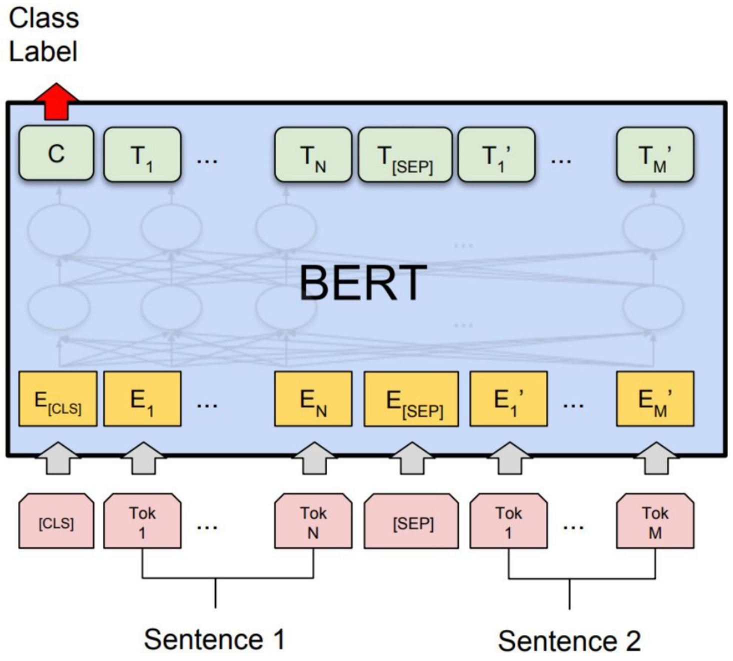
- **DINO v2** [Oquab et al., 2023]
  - DINO v2 is better at transferring to vision tasks
    - Semantic segmentation on ADE20K, Cityscapes, Pascal VOC with frozen feature
    - Depth estimation on NYUd, KITTI, NYUd -> SUN RGB-D with frozen feature

		NYUd (0.330)			KITTI (2.10)			NYUd $\rightarrow$ SUN RGB-D (0.421)		
Method	Arch.	lin. 1	lin. 4	DPT	lin. 1	lin. 4	DPT	lin. 1	lin. 4	DPT
OpenCLIP	ViT-G/14	0.541	0.510	0.414	3.57	3.21	2.56	0.537	0.476	0.408
MAE	ViT-H/14	0.517	0.483	0.415	3.66	3.26	2.59	0.545	0.523	0.506
DINO	ViT-B/8	0.555	0.539	0.492	3.81	3.56	2.74	0.553	0.541	0.520
iBOT	ViT-L/16	0.417	0.387	0.358	3.31	3.07	2.55	0.447	0.435	0.426
DINOv2	ViT-S/14	0.449	0.417	0.356	3.10	2.86	2.34	0.477	0.431	0.409
	ViT-B/14	0.399	0.362	0.317	2.90	2.59	2.23	0.448	0.400	0.377
	ViT-L/14	0.384	0.333	0.293	2.78	2.50	2.14	0.429	0.396	0.360
	ViT-g/14	**0.344**	**0.298**	**0.279**	**2.62**	**2.35**	**2.11**	**0.402**	**0.362**	**0.338**

Table of Contents

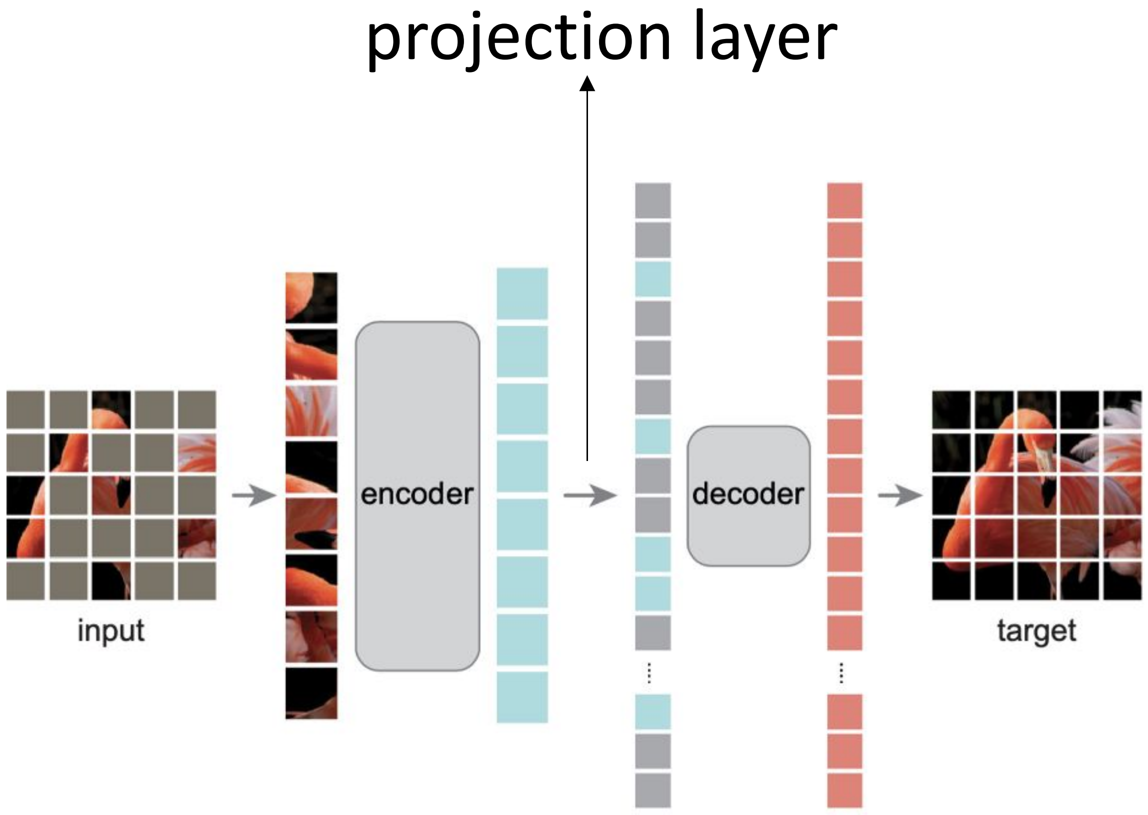
What is Masked Auto-Encoding (MAE)?

- Very simple method, but highly effective
- BERT-like masked modeling objective, but with crucial design changes for computer vision



BERT-*unlike*: Encoder-Decoder

- MAE:
  - *Large* encoder on *visible* tokens
  - Small decoder on *all* tokens
  - Projection layer to connect the two

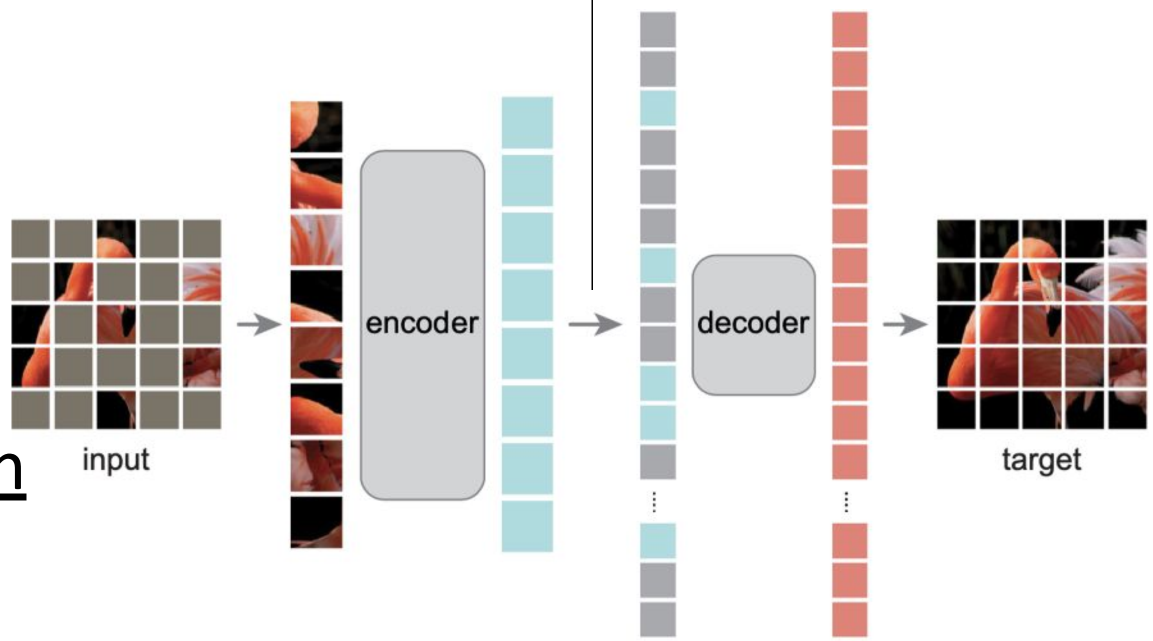


BERT-*unlike*: Encoder-Decoder

- MAE:
  - *Large* encoder on *visible* tokens
  - Small decoder on *all* tokens
  - Projection layer to connect the two

- Very efficient when coupled with <u>high</u> mask ratio (75%)

projection layer



Masked Autoencoders (MAE)

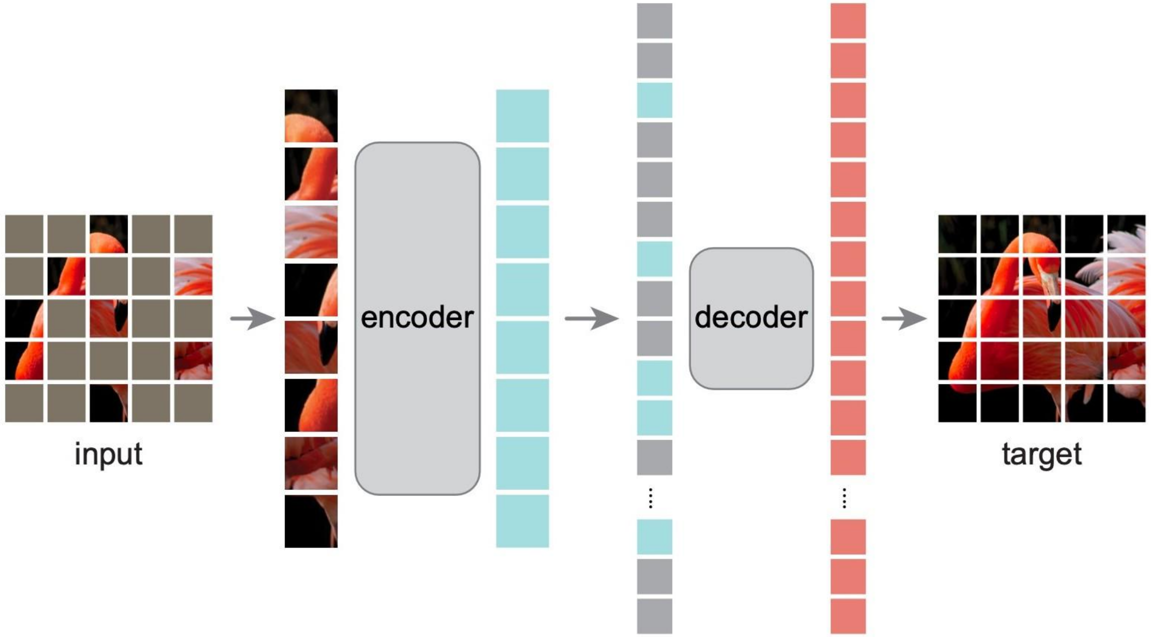
- **MAE** [He et al., 2022]
  - **Task:** Predicting the **pixel** values for each masked patch
    - Objective: **MSE loss of masked patches**

The loss function is:

$$\mathcal{L} = \frac{1}{|M|} \sum_{i \in M} \|f_\theta(x_v)_i - x_i\|^2$$

where:

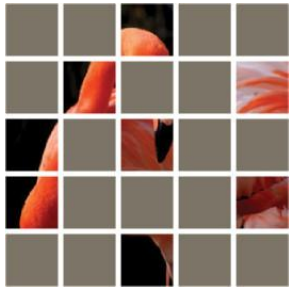
- M is the set of masked patches.
- $|M|$ is the number of masked patches.
- $f_\theta(x_v)$ is the MAE model's reconstruction for the masked patches.
- x_i is the ground-truth pixel value of the masked patch.



- **Key components:**
  - High masking ratio (75%):
    - BERT masks 15% of tokens, MAE needs higher masking ratio
  - Asymmetric encoder-decoder architecture:
    - MAE allows to train very large transformer encoder by using the lightweight decoder => it significantly reduces the pre-training time

How MAE Works?

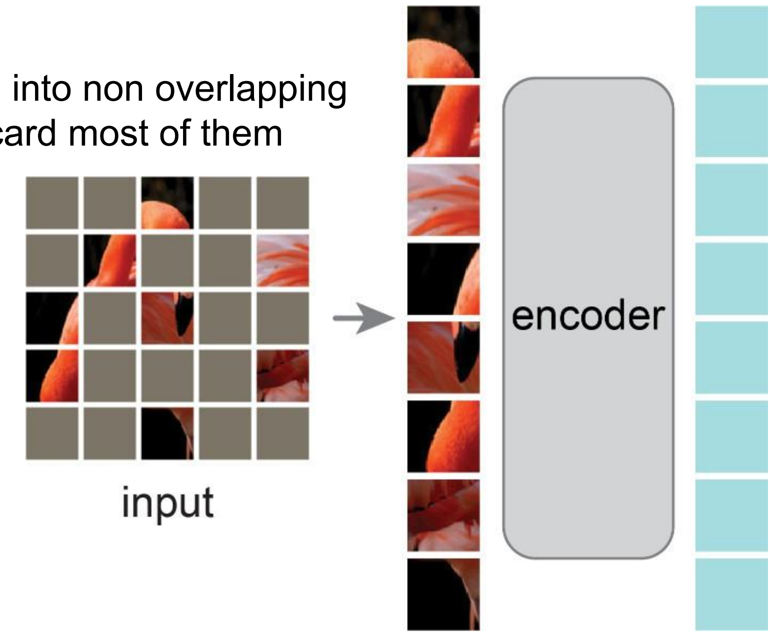
Divide image into non overlapping
patches, discard most of them



Random masking

How MAE Works?

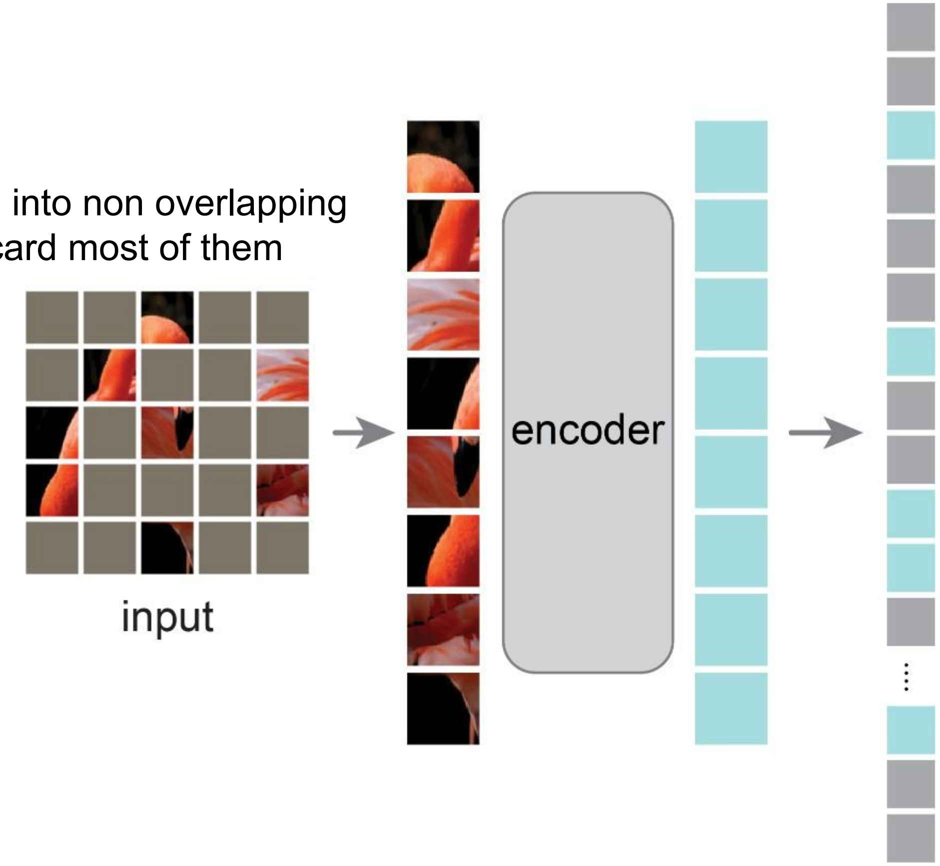
Divide image into non overlapping patches, discard most of them



Encode visible patches with ViT

How MAE Works?

Divide image into non overlapping patches, discard most of them

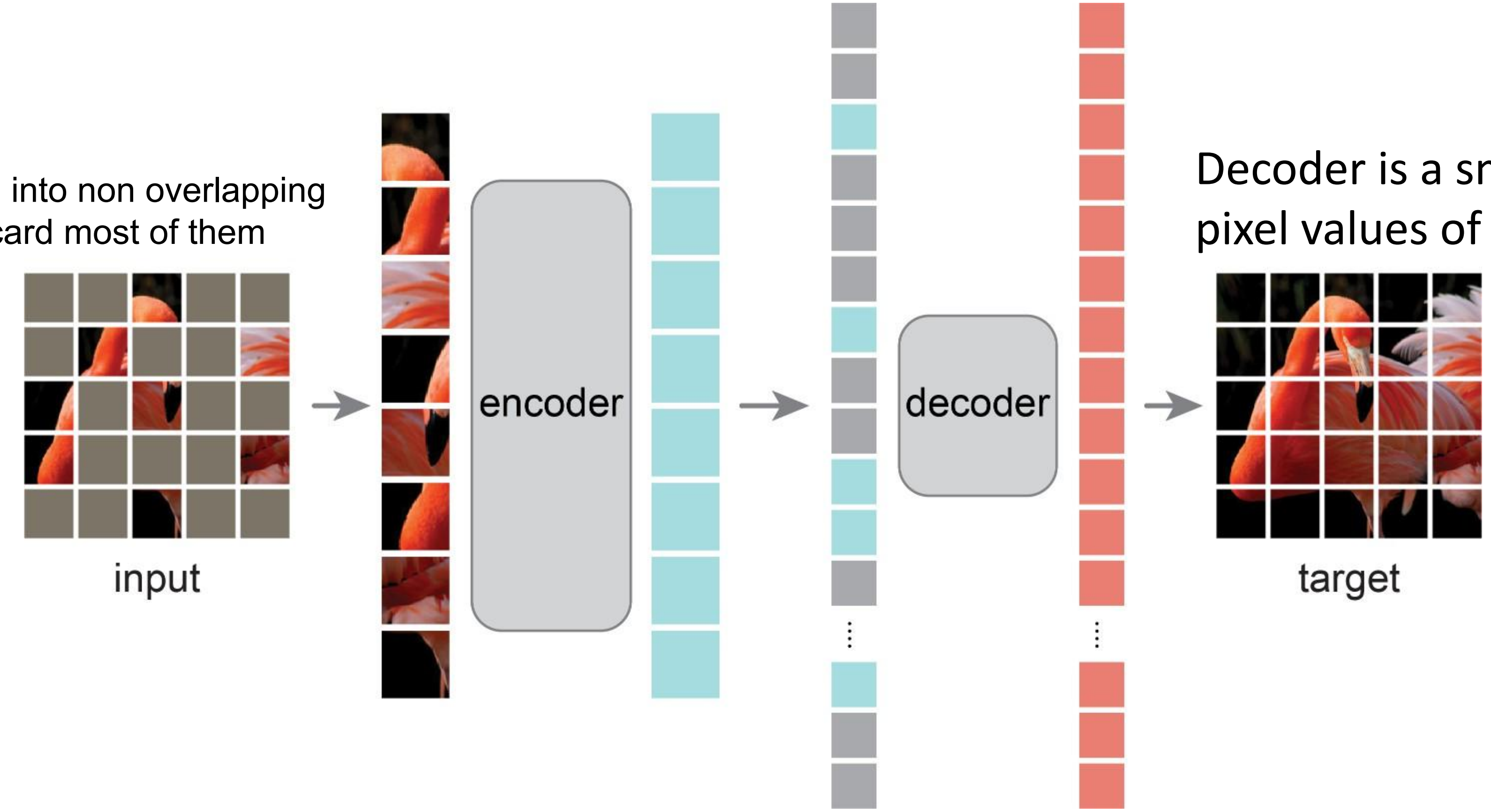


Add mask tokens

How MAE Works?

Divide image into non overlapping patches, discard most of them

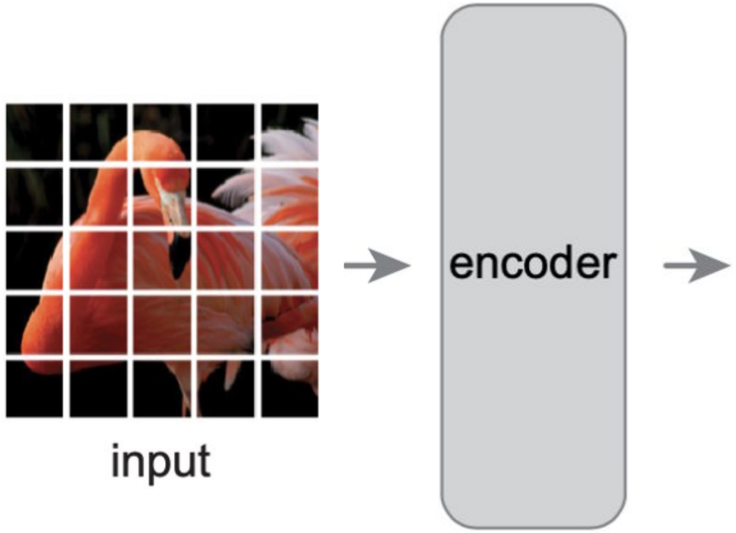
Decoder is a small ViT that predicts pixel values of the masked patches



Reconstruct

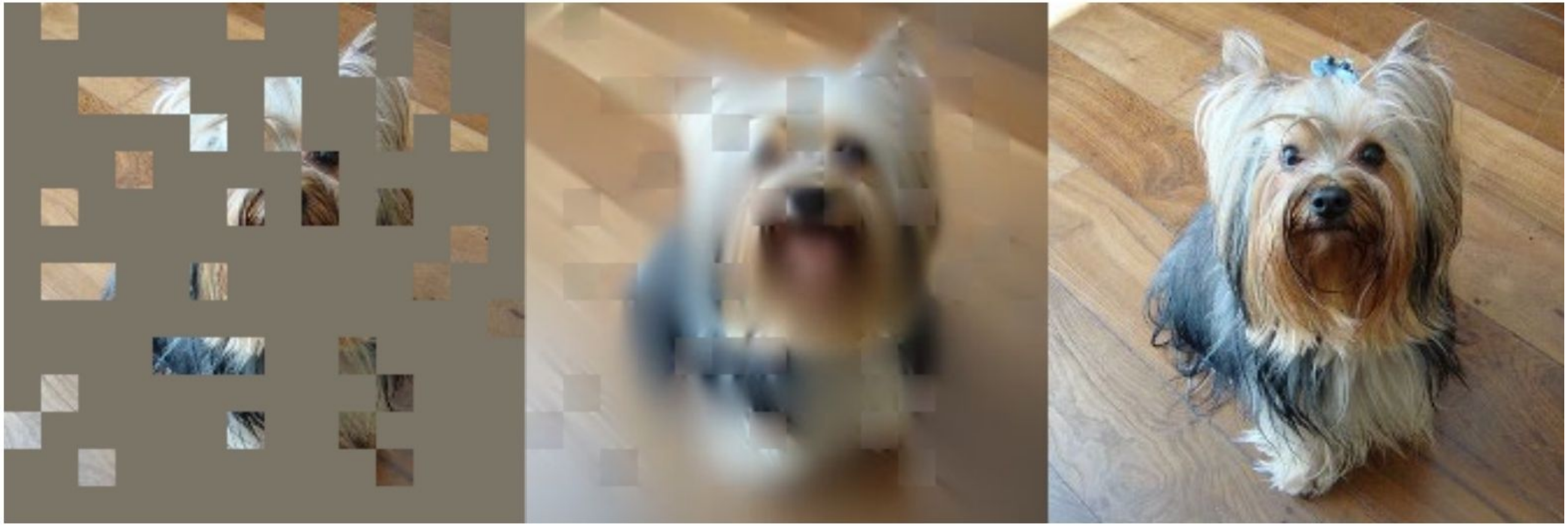
MAE for Downstream Tasks: *Encoder Only*

- After MAE pre-training, just *throw away* the decoder
- Encoder is used for representations with *full-sequence* input



Masked Autoencoders (MAE): Reconstructions

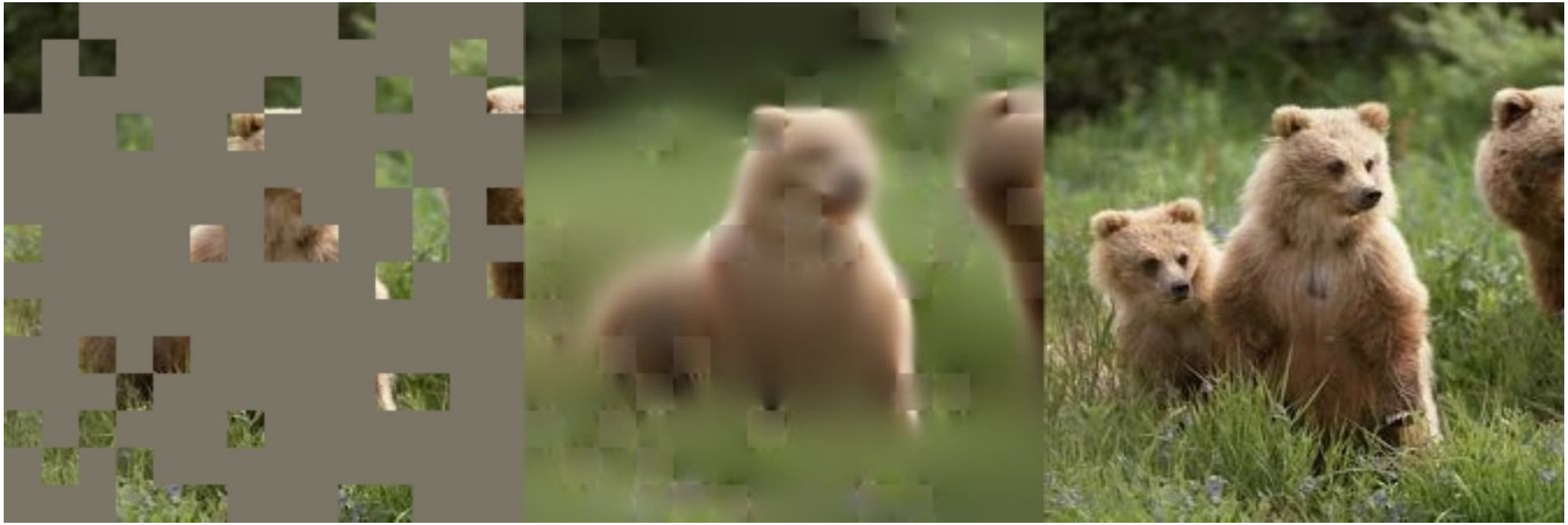
Input Patches Prediction Actual Image



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

Masked Autoencoders (MAE): Reconstructions

Input Patches | Prediction | Actual Image



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

Masked Autoencoders (MAE): Reconstructions

Input Patches | Prediction | Actual Image

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

MAE Reconstruction Example

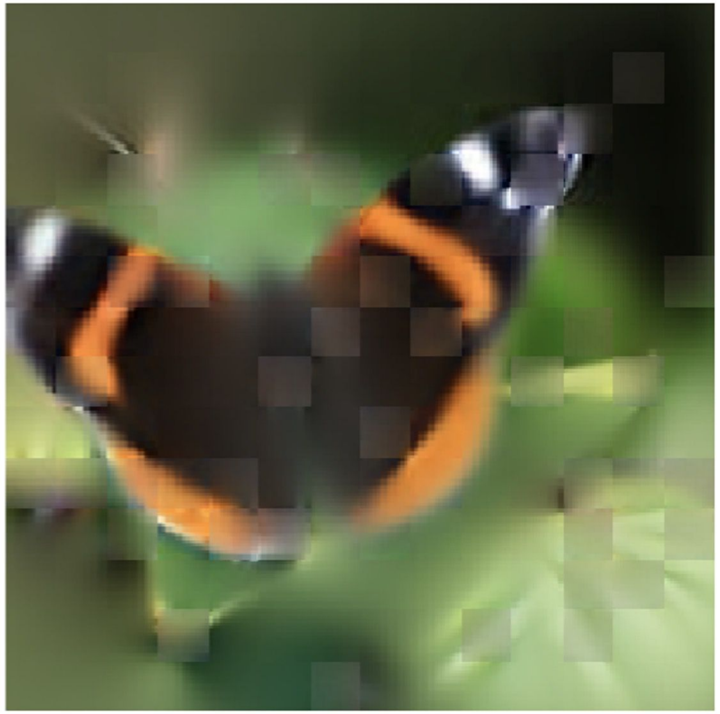
Masked input: 80%

?

You guess?

MAE Reconstruction Example

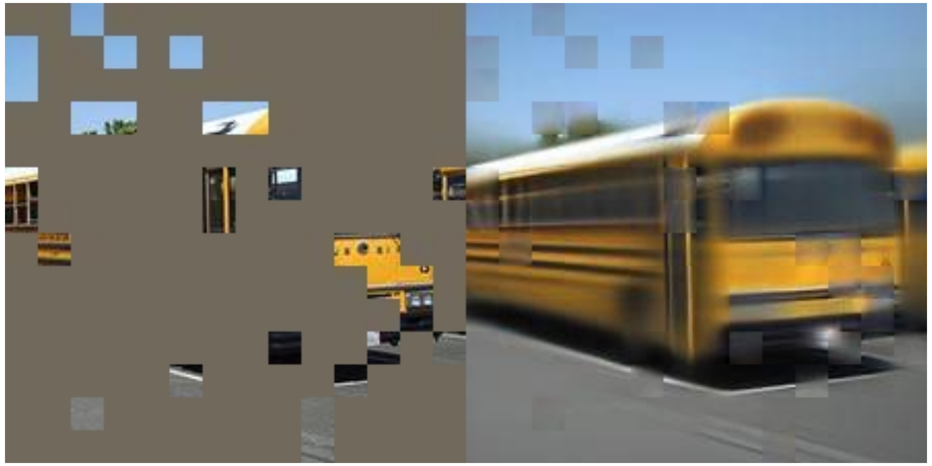
Masked input: 80%



MAE's guess

original

75% mask



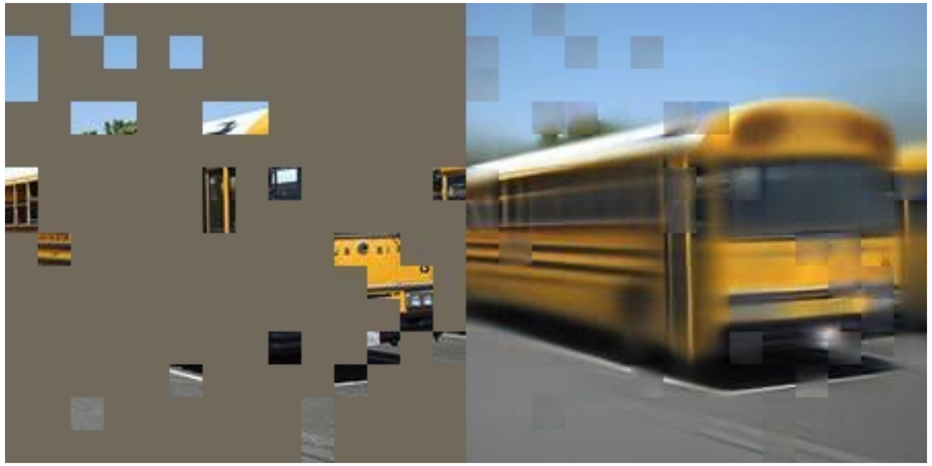
85% mask

MAE Can Generalize

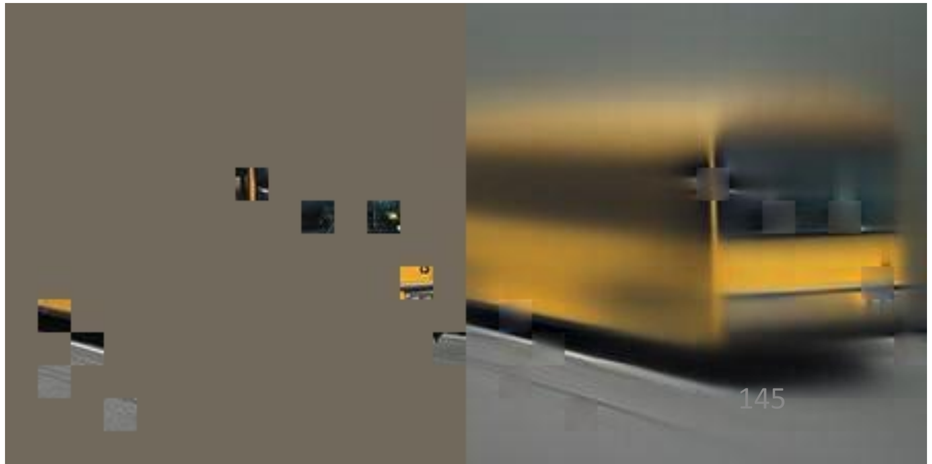
MAE Can Generalize

original

75% mask



85% mask



95% mask

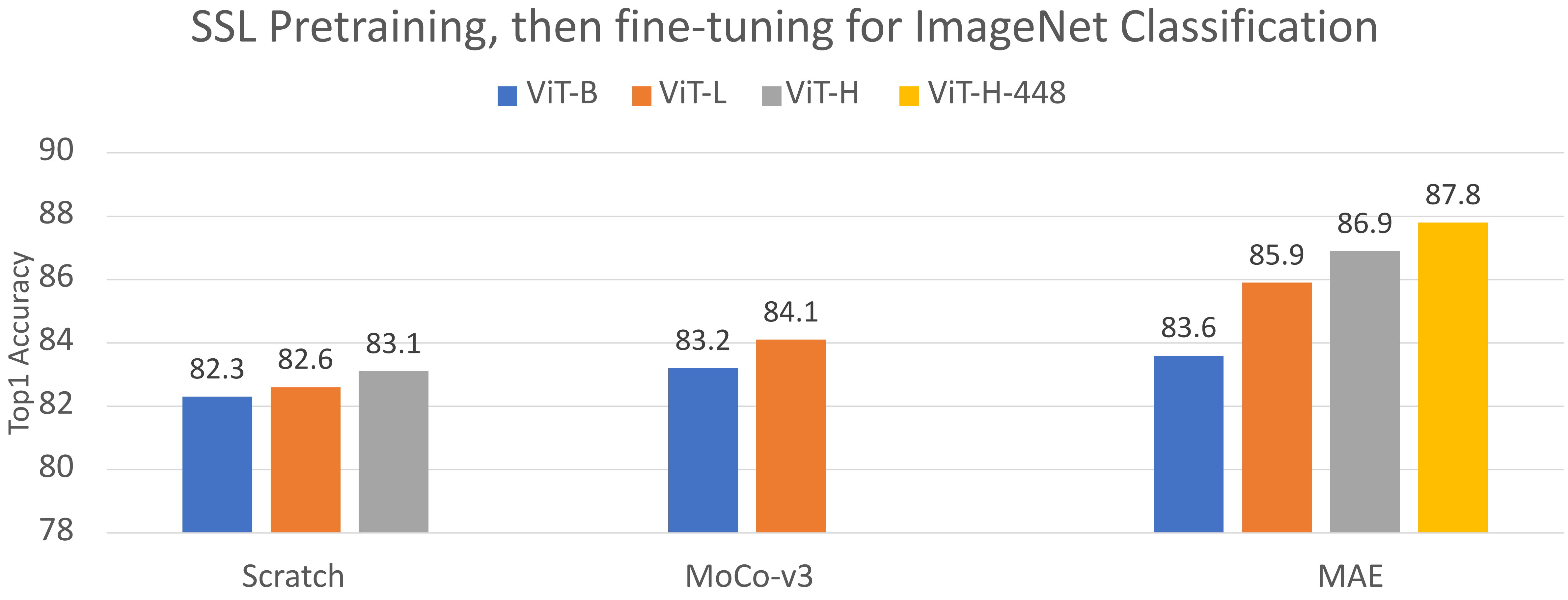
MAE Can Generalize

original

75% mask

85% mask

95% mask



MAE Pretraining outperforms training from scratch, and allows scaling to larger ViT models

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

Masked Autoencoders (MAE)

- **MAE** [He et al., 2022]
  - **Task:** Predicting the **pixel** values for each masked patch
  - **Asymmetric encoder-decoder architecture:** MAE uses the **lightweight decoder**

blocks	ft	lin
1	84.8	65.5
2	**84.9**	70.0
4	**84.9**	71.9
8	**84.9**	**73.5**
12	84.4	73.3

(a) **Decoder depth**. A deep decoder can improve linear probing accuracy.

dim	ft	lin
128	**84.9**	69.1
256	84.8	71.3
512	**84.9**	**73.5**
768	84.4	73.1
1024	84.3	73.1

(b) **Decoder width**. The decoder can be narrower than the encoder (1024-d).

- The decoder depth is less influential for improving fine-tuning
  - Only a single transformer block decoder can perform strongly with fine-tuning

- MAE decoder uses the decoder with 8 blocks and a width of 512-d, which has 9% FLOPs per token *vs.* ViT-L

Masked Autoencoders (MAE)

- **MAE** [He et al., 2022]
  - **Task:** Predicting the **pixel** values for each masked patch
  - **Other properties of MAE**

case	ft	lin	FLOPs
encoder w/ [M]	84.2	59.6	3.3×
encoder w/o [M]	**84.9**	**73.5**	**1**×

(c) **Mask token.** An encoder without mask tokens is more accurate and faster (Table 2).

case	ft	lin
pixel (w/o norm)	84.9	73.5
pixel (w/ norm)	**85.4**	**73.9**
PCA	84.6	72.3
dVAE token	85.3	71.6

(d) **Reconstruction target.** Pixels as reconstruction targets are effective.

case	ft	lin
none	84.0	65.7
crop, fixed size	84.7	73.1
crop, rand size	**84.9**	**73.5**
crop + color jit	84.3	71.9

(e) **Data augmentation.** Our MAE works with minimal or no augmentation.

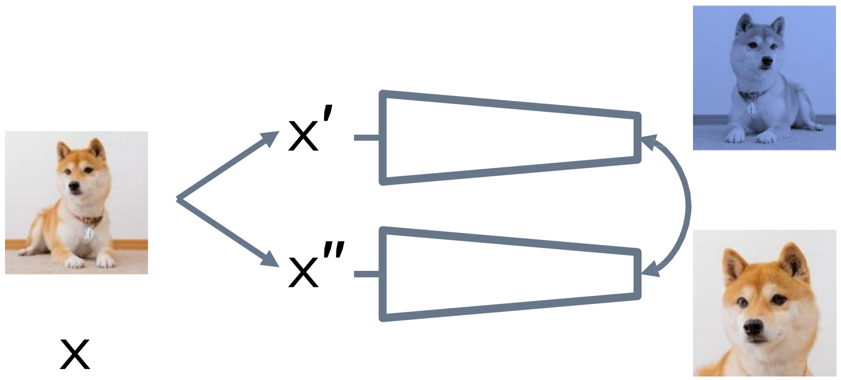
(c) MAE skips the mask token [M] in the encoder and apply it later in the decoder
- It is more accurate and decreases the computation time

(d) Predicting pixels with *per-patch* normalization improves accuracy

(e) MAE works well using cropping-only augmentation
- MAE behaves decently even if using no data augmentation

Analysis: Augmentations

case	ft	lin
none	84.0	65.7
crop, fixed size	84.7	73.1
crop, rand size	**84.9**	**73.5**
crop + color jit	84.3	71.9

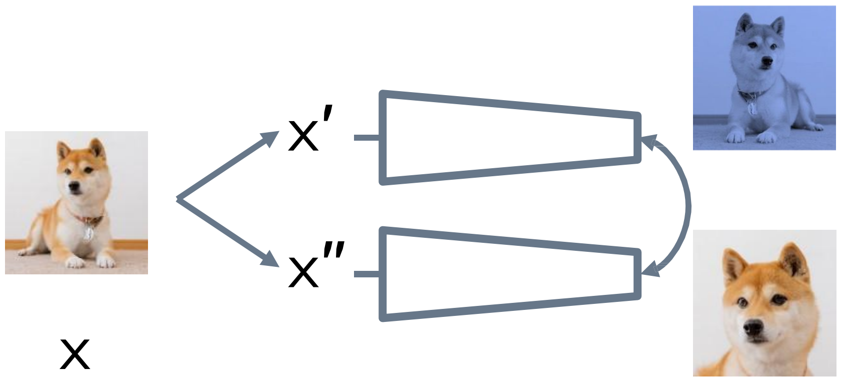


- MAE can work with minimal data augmentation
- For Contrastive / Siamese learning, augmentation is <u>crucial</u>

[Assran et al, ECCV 2022] [Assran et al, CVPR 2023]

Analysis: Augmentations

case	ft	lin
none	84.0	65.7
crop, fixed size	84.7	73.1
crop, rand size	**84.9**	**73.5**
crop + color jit	84.3	71.9



- MAE can work with minimal data augmentation
- For Contrastive / Siamese learning, augmentation is <u>crucial</u>
- Masking as a strong "augmentation": MSN, I-JEPA

Masked Autoencoders (MAE)

- **MAE** [He et al., 2022]
  - **Task:** Predicting the **pixel** values for each masked patch
  - **Other properties of MAE**

case	ratio	ft	lin
random	75	**84.9**	**73.5**
block	50	83.9	72.3
block	75	82.8	63.9
grid	75	84.0	66.0

(f) **Mask sampling**. Random sampling works the best. See Figure 6 for visualizations.

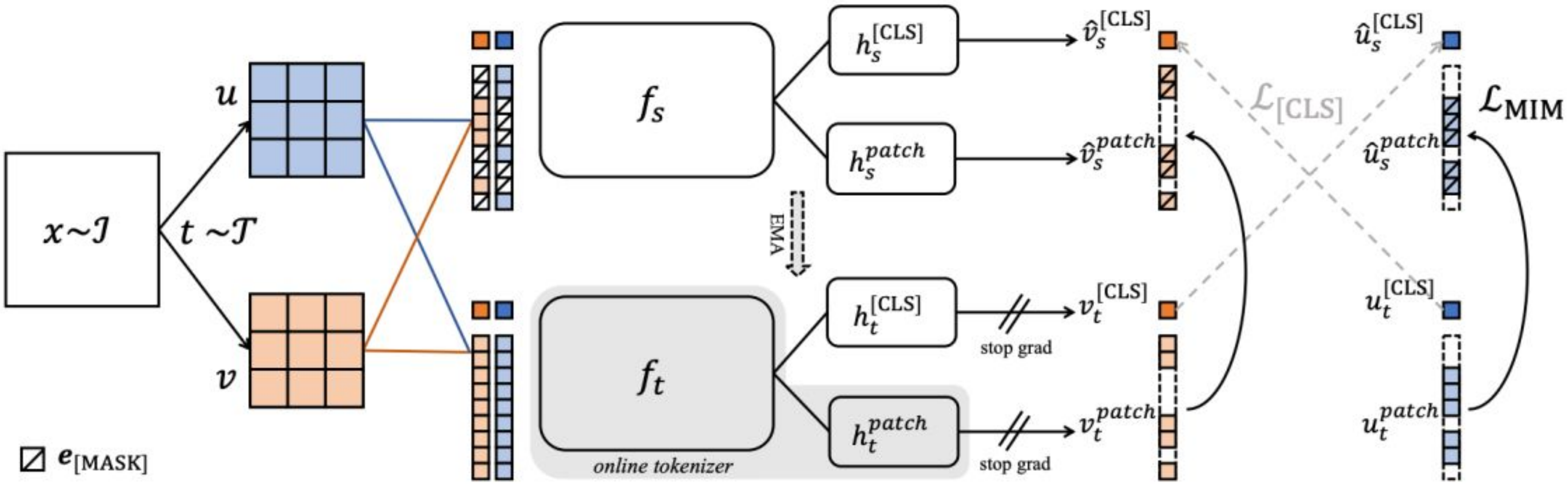
random 75% block 50% grid 75%

(f) Random patch masking is better than block-wise and grid-wise sampling
- Block-wise sampling: Removes large random blocks
- Grid-wise sampling: Keeps one of every four patches

Table of Contents

Both

- **Image-BERT Pretraining with online tokenizer (IBOT)** [Zhou et al., 2022]
  - Perform patch-level self-distillation on masked patch tokens (while DINO is done with image-level objective)
  - Use data augmentation for invariance learning
  - Unlike BEiT, image tokenizer is jointly learned (i.e., online tokenizer)



- **Image-BERT Pretraining with online tokenizer (IBOT)** [Zhou et al., 2022]
  - IBOT shows strong performance on linear probing as well as fine-tuning
  - IBOT demonstrates high transferability on various downstream tasks such as semi-supervised learning, unsupervised learning, object detection, and segmentation

Table 4: **Semi-supervised learning on ImageNet-1K.** 1% and 10% denotes label fraction. SD denotes self-distillation.

Method	Arch.	1%	10%
SimCLRv2	RN50	57.9	68.1
BYOL	RN50	53.2	68.8
SwAV	RN50	53.9	70.2
SimCLRv2+SD	RN50	60.0	70.5
DINO	ViT-S/16	60.3	74.3
iBOT	ViT-S/16	**61.9**	**75.1**

Table 5: **Unsupervised learning on ImageNet-1K.** † denotes k-means clustering on frozen features.

Method	Arch.	ACC	ARI	NMI	FMI
Self-label†	RN50	30.5	16.2	75.4	-
InfoMin†	RN50	33.2	14.7	68.8	-
SCAN	RN50	39.9	27.5	72.0	-
DINO	ViT-S/16	41.4	29.8	76.8	32.8
iBOT	ViT-S/16	**43.4**	**32.8**	**78.6**	**35.6**

Table 6: **Object detection (Det.) & instance segmentation (ISeg.) on COCO and Semantic segmentation (Seg.) on ADE20K.** We report the results of ViT-S/16 (left) and ViT-B/16 (right). Seg.† denotes using a linear head for semantic segmentation.

Method	Arch.	Param.	Det.	ISeg.	Seg.
			AP^b	AP^m	mIoU
Sup.	Swin-T	29	48.1	41.7	44.5
MoBY	Swin-T	29	48.1	41.5	44.1
Sup.	ViT-S/16	21	46.2	40.1	44.5
iBOT	ViT-S/16	21	**49.4**	**42.6**	**45.4**

Method	Det.	ISeg.	Seg.†	Seg.
	AP^b	AP^m	mIoU	mIoU
Sup.	49.8	43.2	35.4	46.6
BEiT	50.1	43.5	27.4	45.8
DINO	50.1	43.4	34.5	46.8
iBOT	**51.2**	**44.2**	**38.3**	**50.0**

Table of Contents

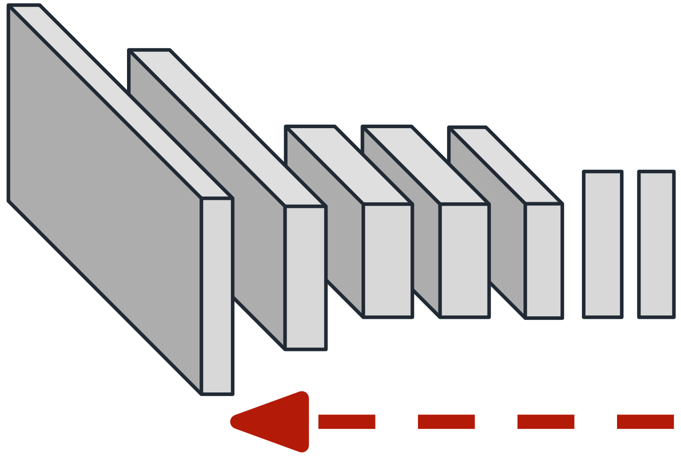
Evaluation

How to evaluate?

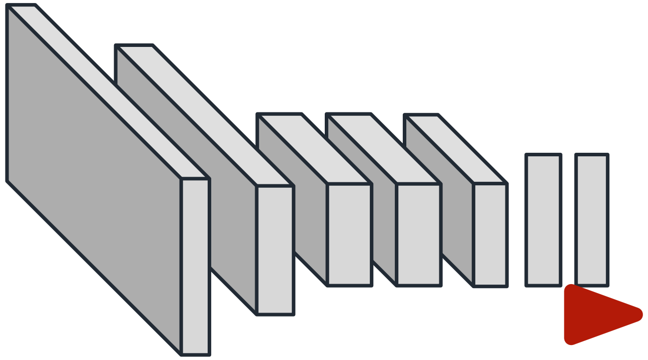
Most standard way:

1. Use the pretrained network from self-supervised learning
2. Use some amount of **labeled data** for the downstream task Measure performance

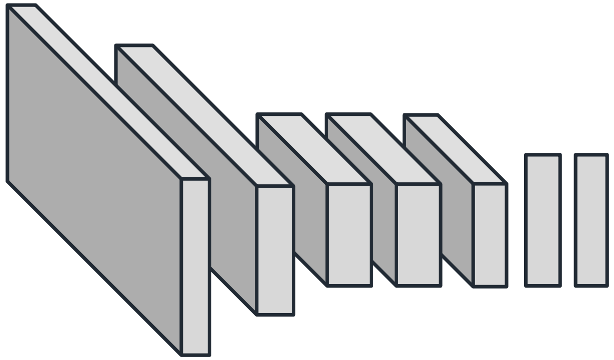
How to use the labeled data?



Fine-tune all layers

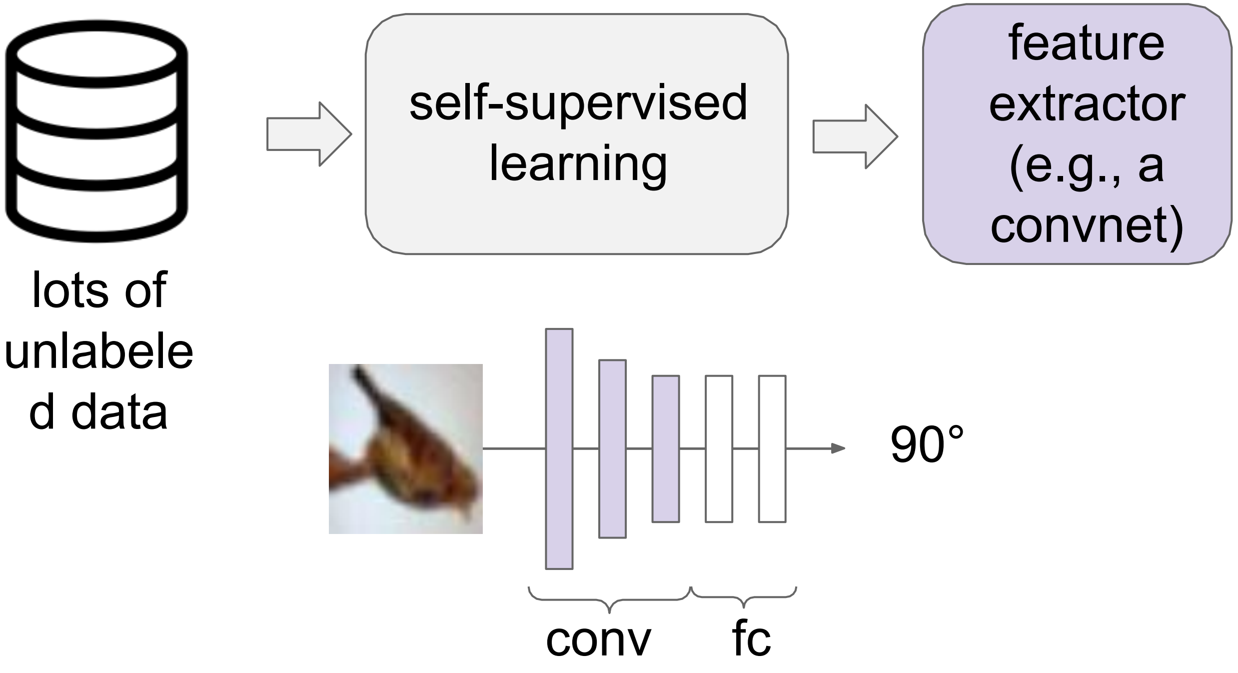


Linear classifier



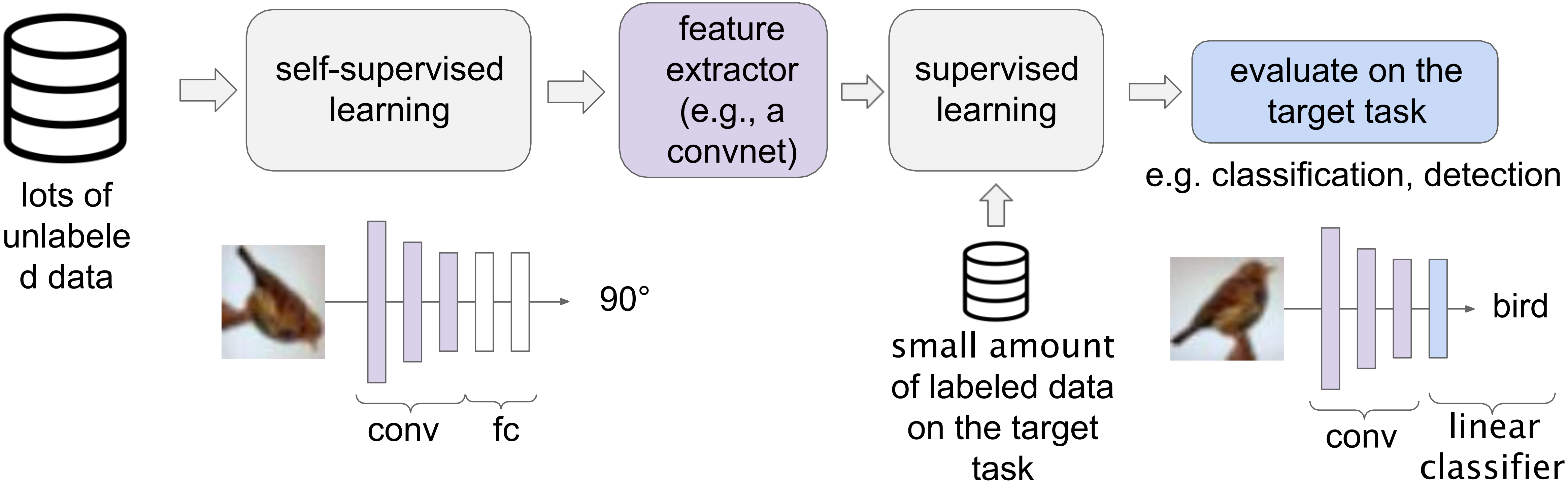
kNN

How to evaluate a self-supervised learning method?



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

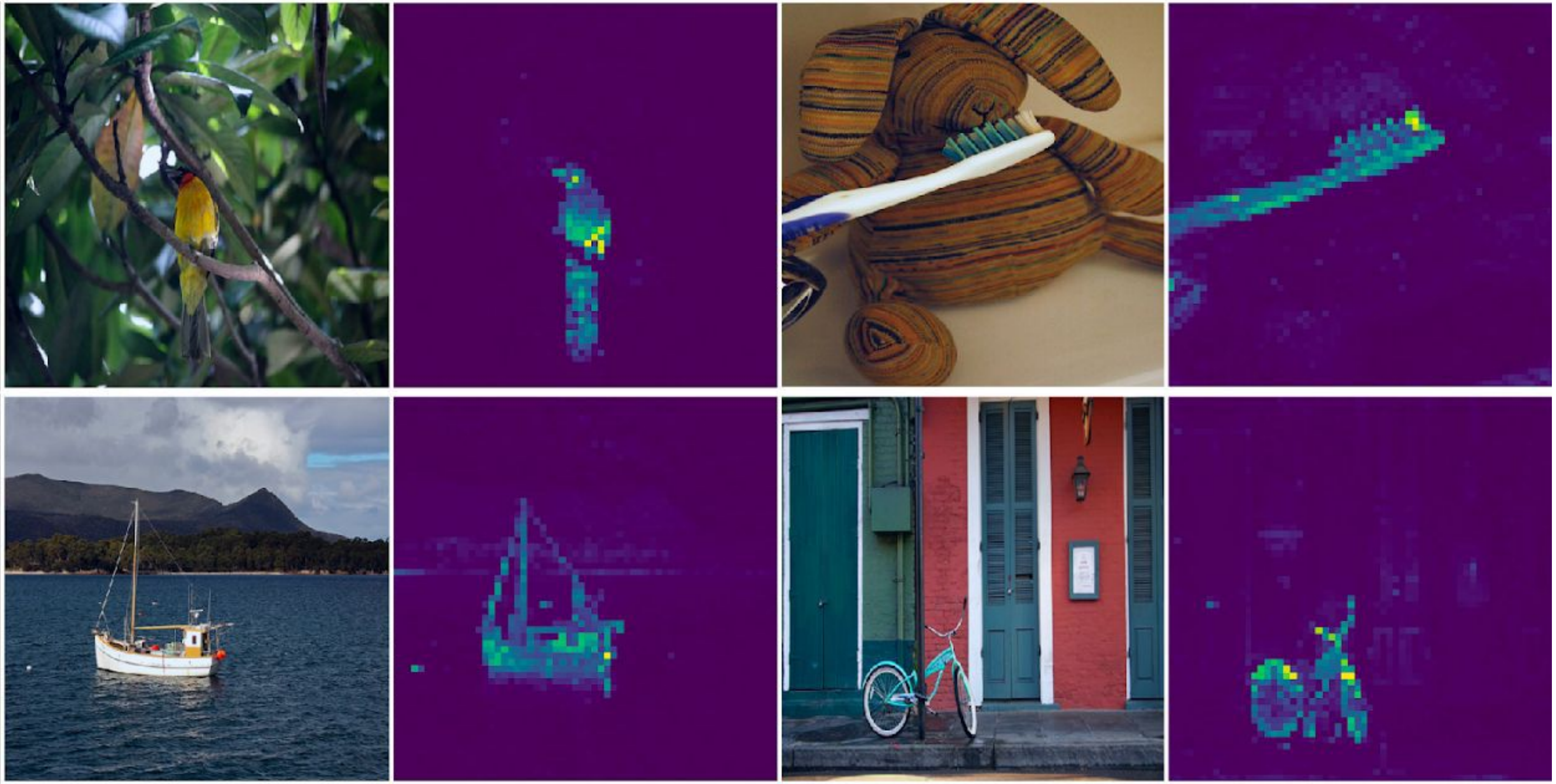
How to evaluate a self-supervised learning method?



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

Are the models useful without any labeled data?



DINO - Caron et al., 2022

Large
Language
Models

Vision - Language GAP

- Self-supervised learning allows representation learning at *scale*

- Masked auto-encoders as a step toward scalable vision learners

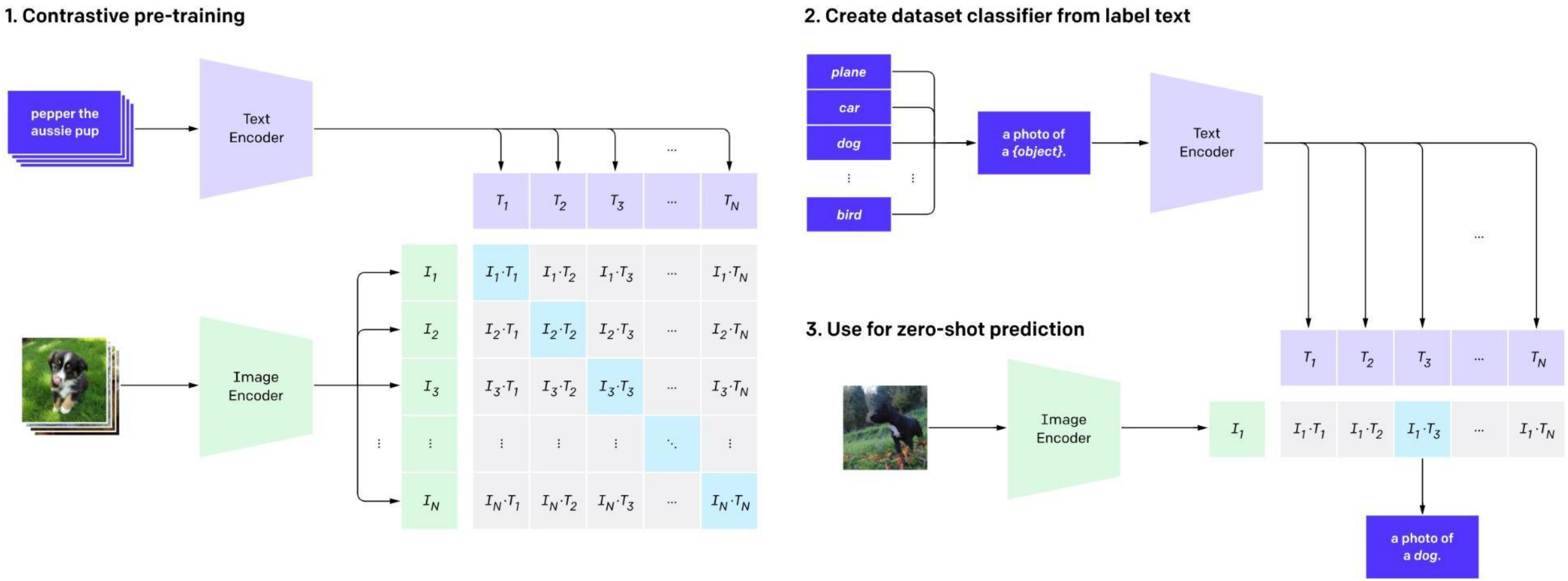
- Still need to close the gap with large language models

MAE

Table of Contents

CLIP: Contrastive Language-Image Pre-training

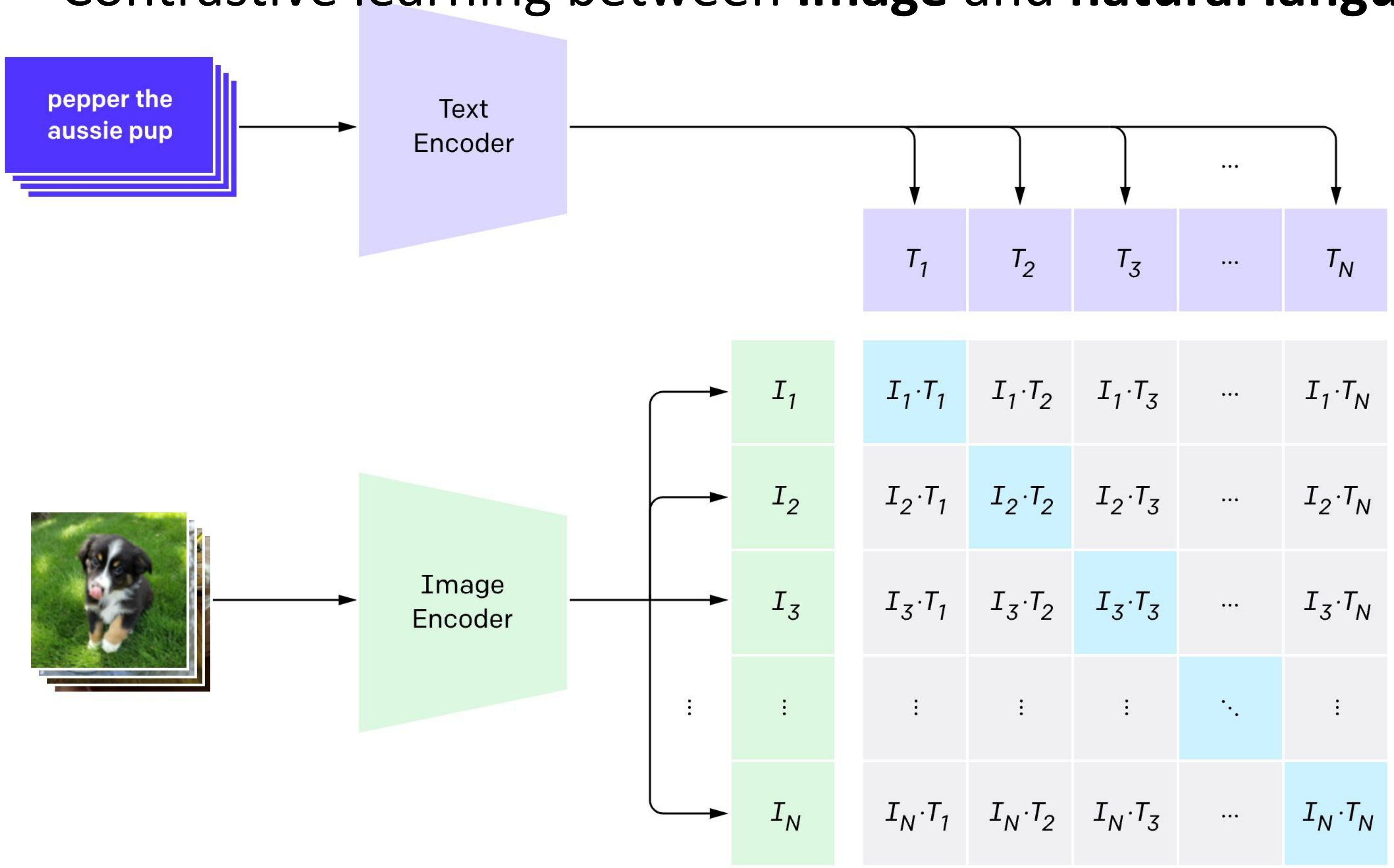
- Contrastive learning between **image** and **natural language** sentences



CLIP (*Contrastive Language–Image Pre-training*) Radford *et al.*, 2021

CLIP: Contrastive Language-Image Pre-training

- Contrastive learning between **image** and **natural language** sentences



Contrastive loss: Each image predicts which caption matches

Large-scale training on 400M (image, text) pairs from the internet

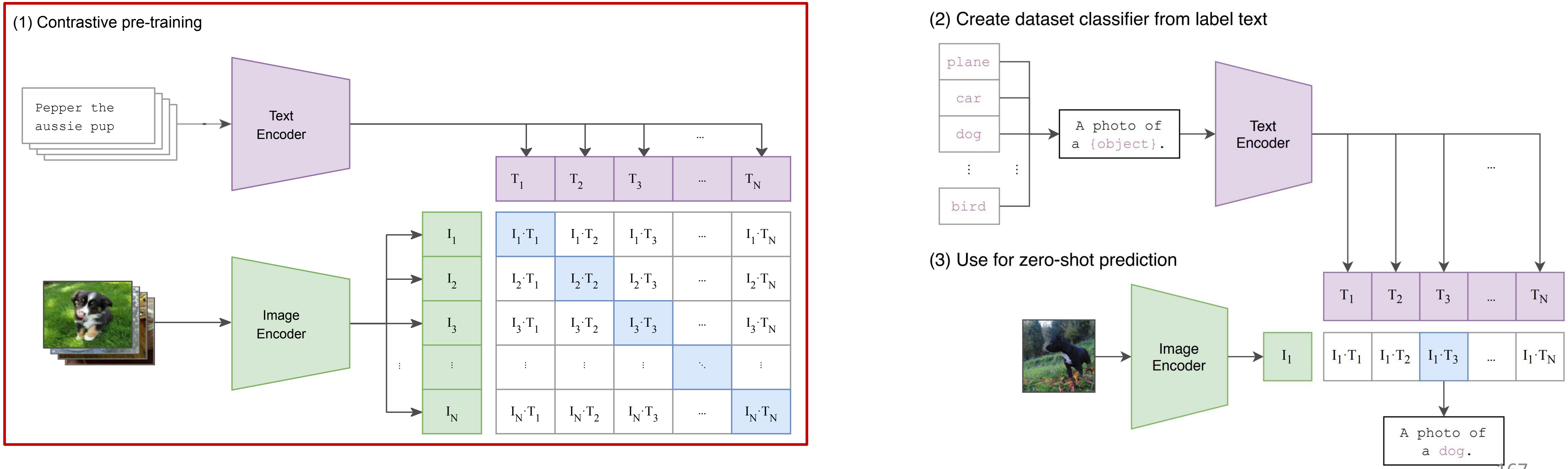
Radford et al, "Learning Transferable Visual Models form Natural Language Supervision", ICML 2021
Jia et al, "Scaling up Visual and Vision-Language Representation Learning with Noisy Text Supervision", ICLR 2021

CLIP: Contrastive Language-Image Pre-training

CLIP [Radford et al., 2020]

- Simple contrastive learning between **image** and **text** embeddings
- Trained on large-scale web image-text pairs

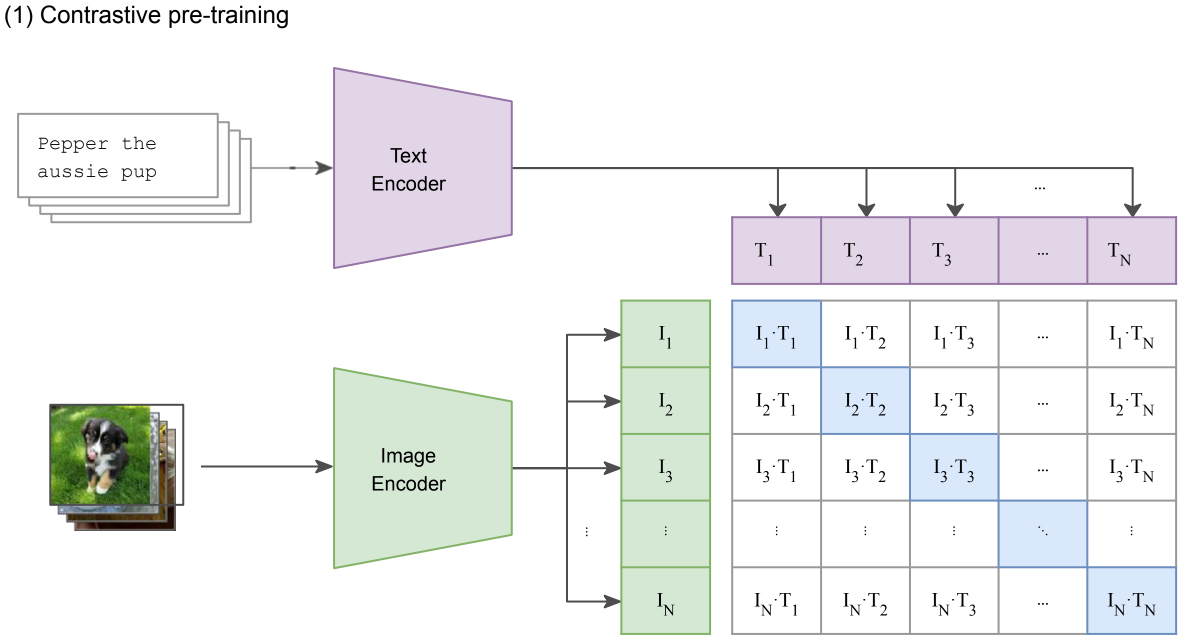
$$L_{\mathrm{CLIP}} = -\frac{1}{2N}\sum_{i=1}^{N}\log\frac{\exp(I_i\cdot T_i)}{\sum_{j=1}^{N}\exp(I_i\cdot T_j)} - \frac{1}{2N}\sum_{j=1}^{N}\log\frac{\exp(I_j\cdot T_j)}{\sum_{i=1}^{N}\exp(I_i\cdot T_j)}$$



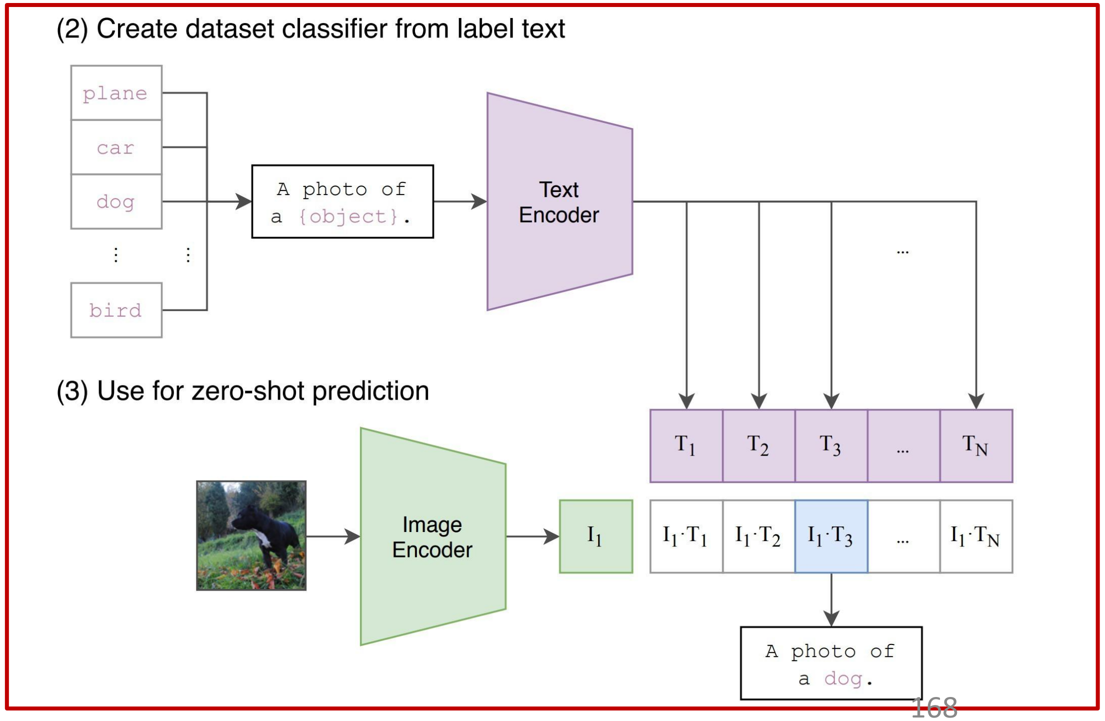
CLIP: Contrastive Language-Image Pre-training

CLIP [Radford et al., 2020]

- **Zero-shot transfer**
  - Transfer learning without seeing the images or labels
  - **Prompt Engineering:** ”A photo of a [MASK]”
  - Choose class that maximizes similarity with respect to image



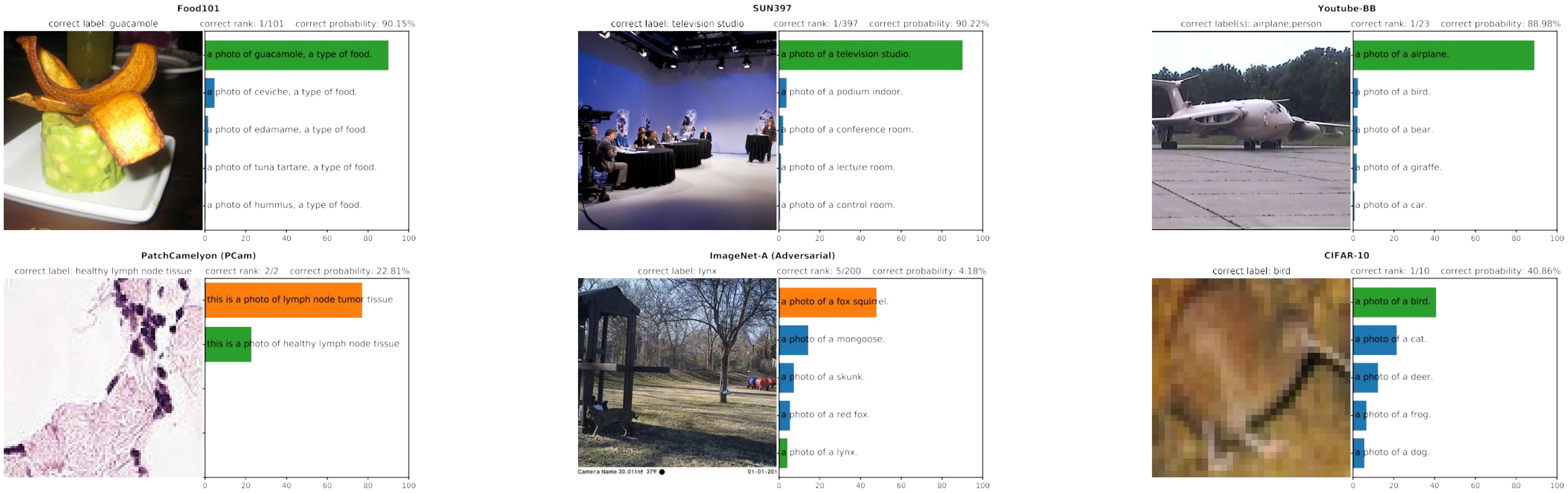
Language enables zero- shot classification: Classify images into categories without any additional training data!



CLIP: Contrastive Language-Image Pre-training

CLIP [Radford et al., 2020]

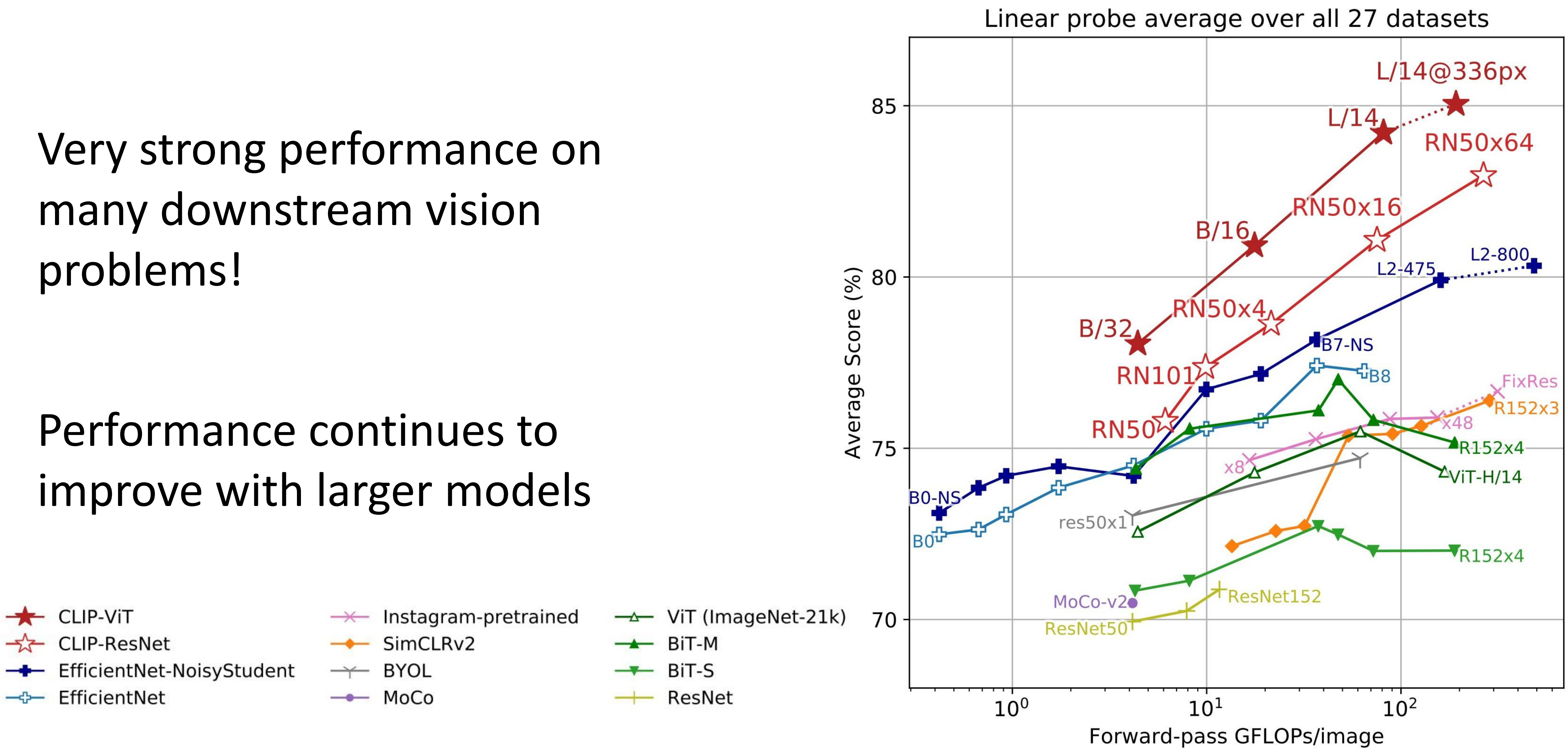
- Zero-shot transfer
  - Transfer learning without seeing the images or labels
  - **Prompt Engineering**: ”A photo of a [MASK]”
  - Choose class that maximizes similarity with respect to image



CLIP: Contrastive Language-Image Pre-training

Very strong performance on many downstream vision problems!

Performance continues to improve with larger models

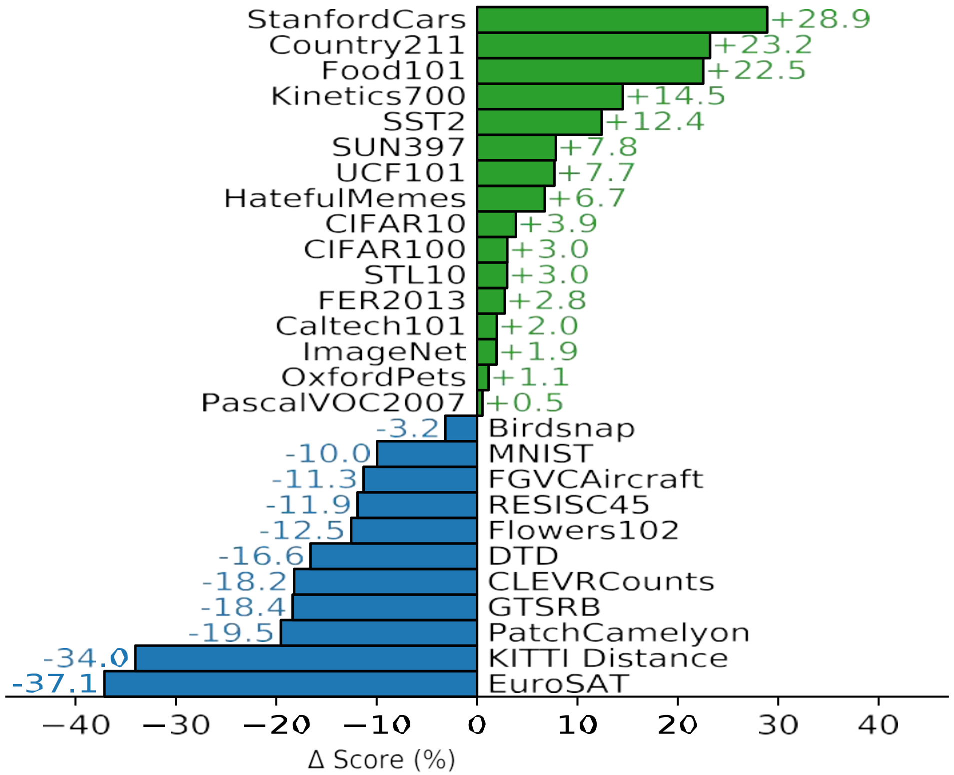


Radford et al, "Learning Transferable Visual Models form Natural Language Supervision", ICML 2021

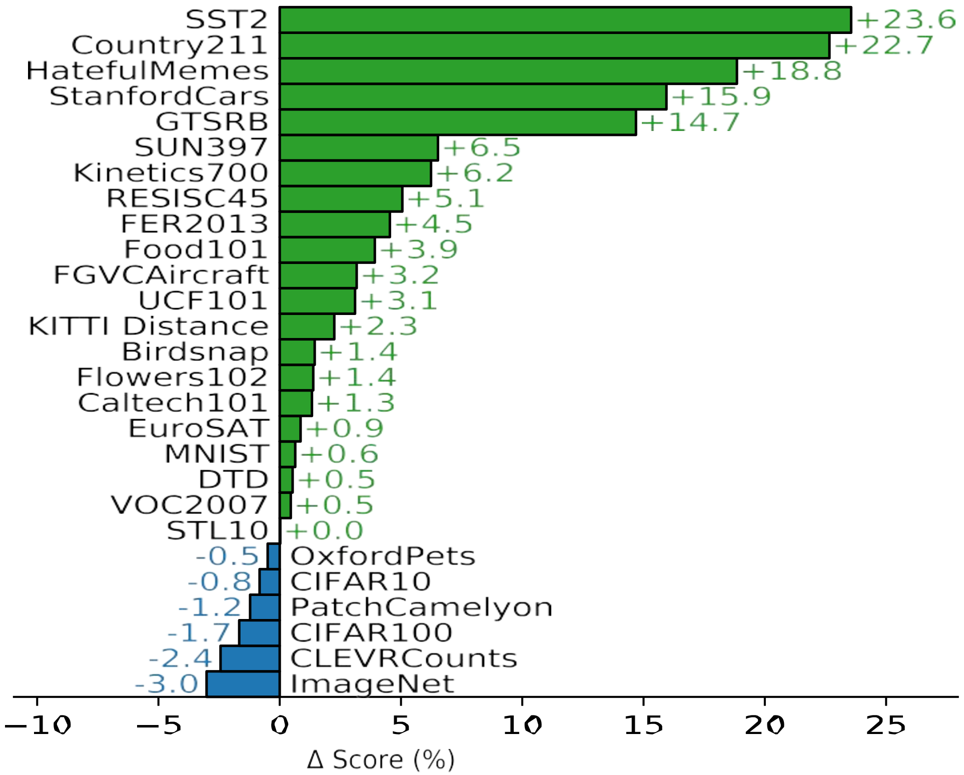
CLIP: Contrastive Language-Image Pre-training

CLIP [Radford et al., 2020]

- A zero-shot CLIP classifier shows a competitive performance with a fully supervised linear classifier fitted on ResNet-50 features
- Linear-probing with CLIP image features outperform the best ImageNet model



Zero-Shot CLIP vs. Linear Probe on ResNet50

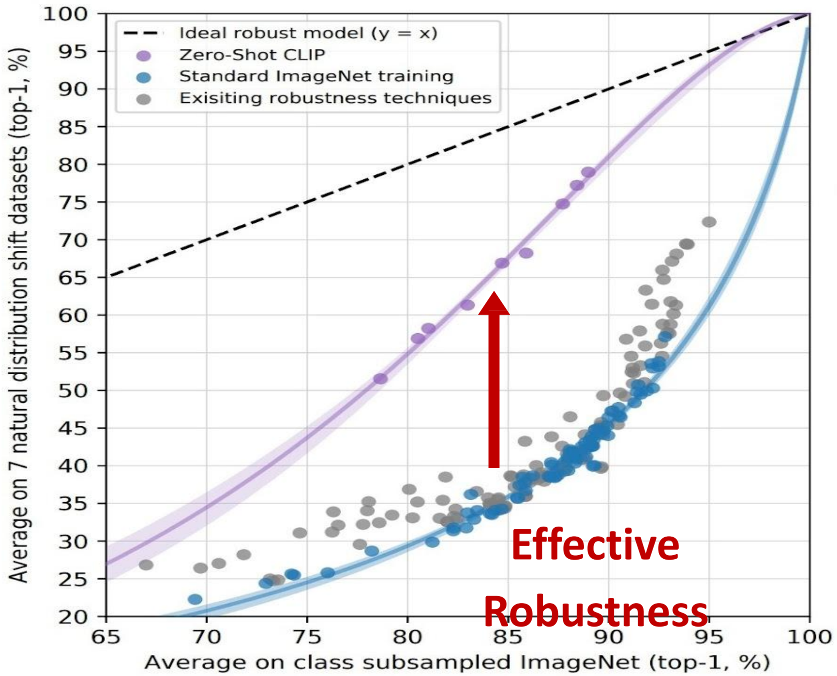


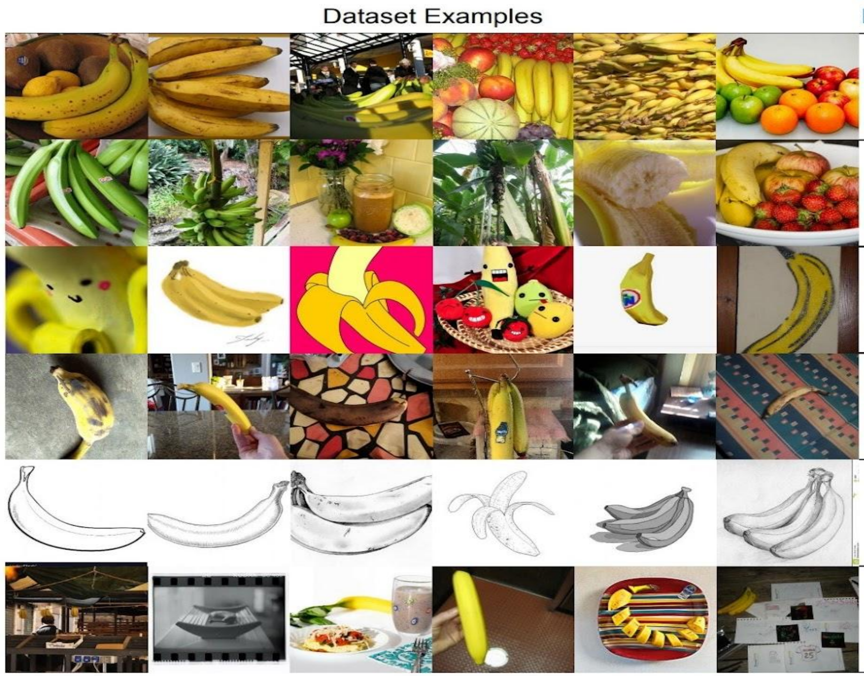
Logistic Regression on CLIP vs. EfficientNet L2 NS

CLIP: Contrastive Language-Image Pre-training

CLIP [Radford et al., 2020]

- Zero-shot CLIP classifier is more robust to natural **distributional shift**
  - Interestingly, [Ilharco et al., 2021] show that CLIP have high **effective robustness** even at small scale



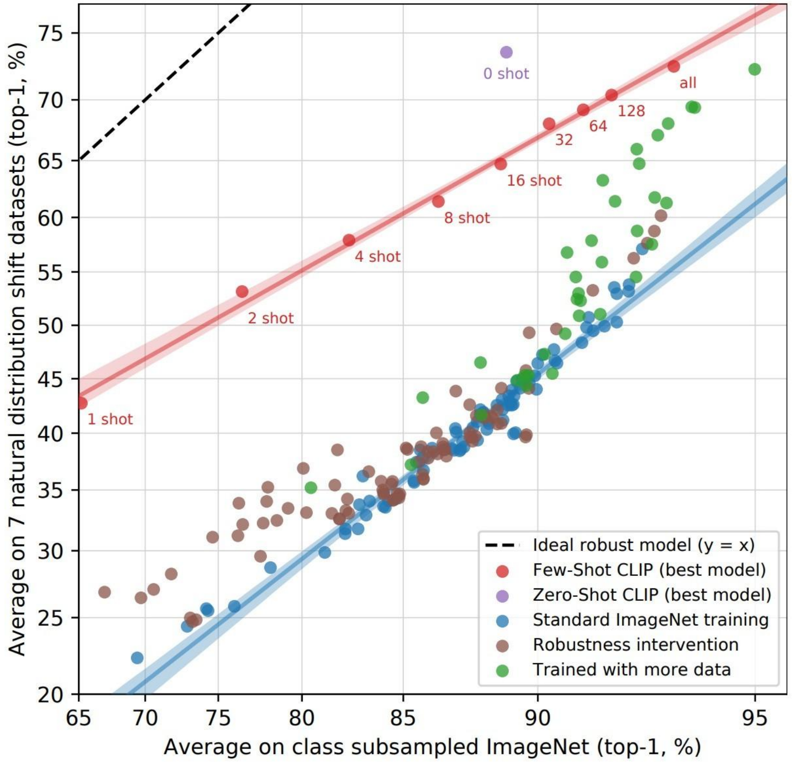


	ImageNet ResNet101	Zero-Shot CLIP	Δ Score
ImageNet	**76.2**	**76.2**	0%
ImageNetV2	64.3	**70.1**	+5.8%
ImageNet-R	37.7	**88.9**	+51.2%
ObjectNet	32.6	**72.3**	+39.7%
ImageNet Sketch	25.2	**60.2**	+35.0%
ImageNet-A	2.7	**77.1**	+74.4%

CLIP: Contrastive Language-Image Pre-training

CLIP [Radford et al., 2020]

- Zero-shot CLIP classifier is more robust to natural **distributional shift**
  - Interestingly, [Ilharco et al., 2021] show that CLIP have high **effective robustness** even at small scale
- Few-shot CLIP classifier also shows high effective robustness, but less than zero- shot CLIP classifier

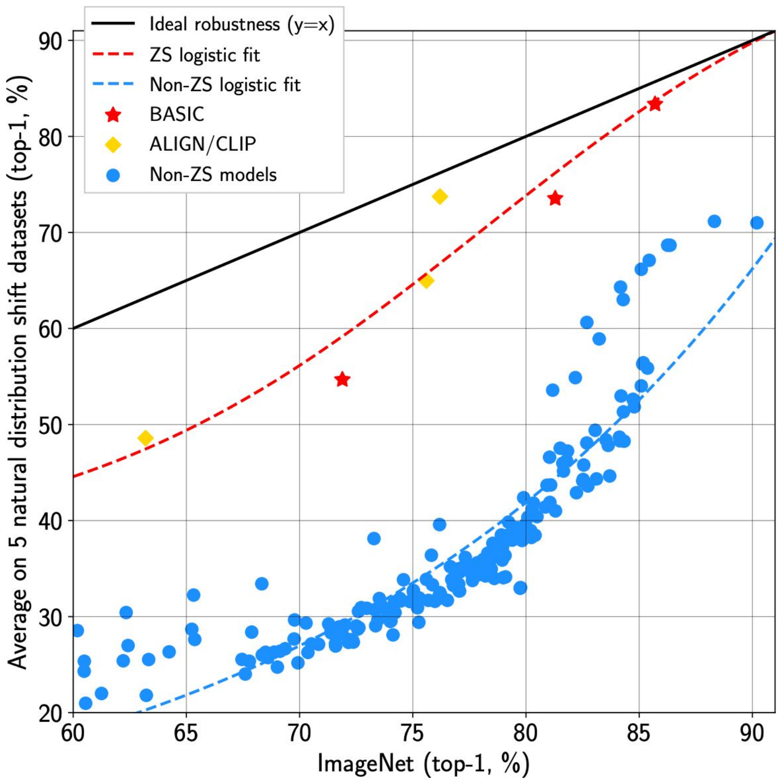


CLIP: Contrastive Language-Image Pre-training

- Scaling Up dataset size for improved CLIP

Follow-up studies showed scaling dataset size improves performance

- CLIP uses carefully filtered **400M** image-text pairs from web
- **ALIGN** [Jia et al., 2020] collected noisy **1.8B** image-text pairs to scale CLIP
- **BASIC** [Pham et al., 2021] used **6.6B** image-text pairs with bigger model size



CLIP: Contrastive Language-Image Pre-training

- **Limitation**

https://cdn.openai.com/papers/dall-e-2.pdf

Table of Contents

Meta

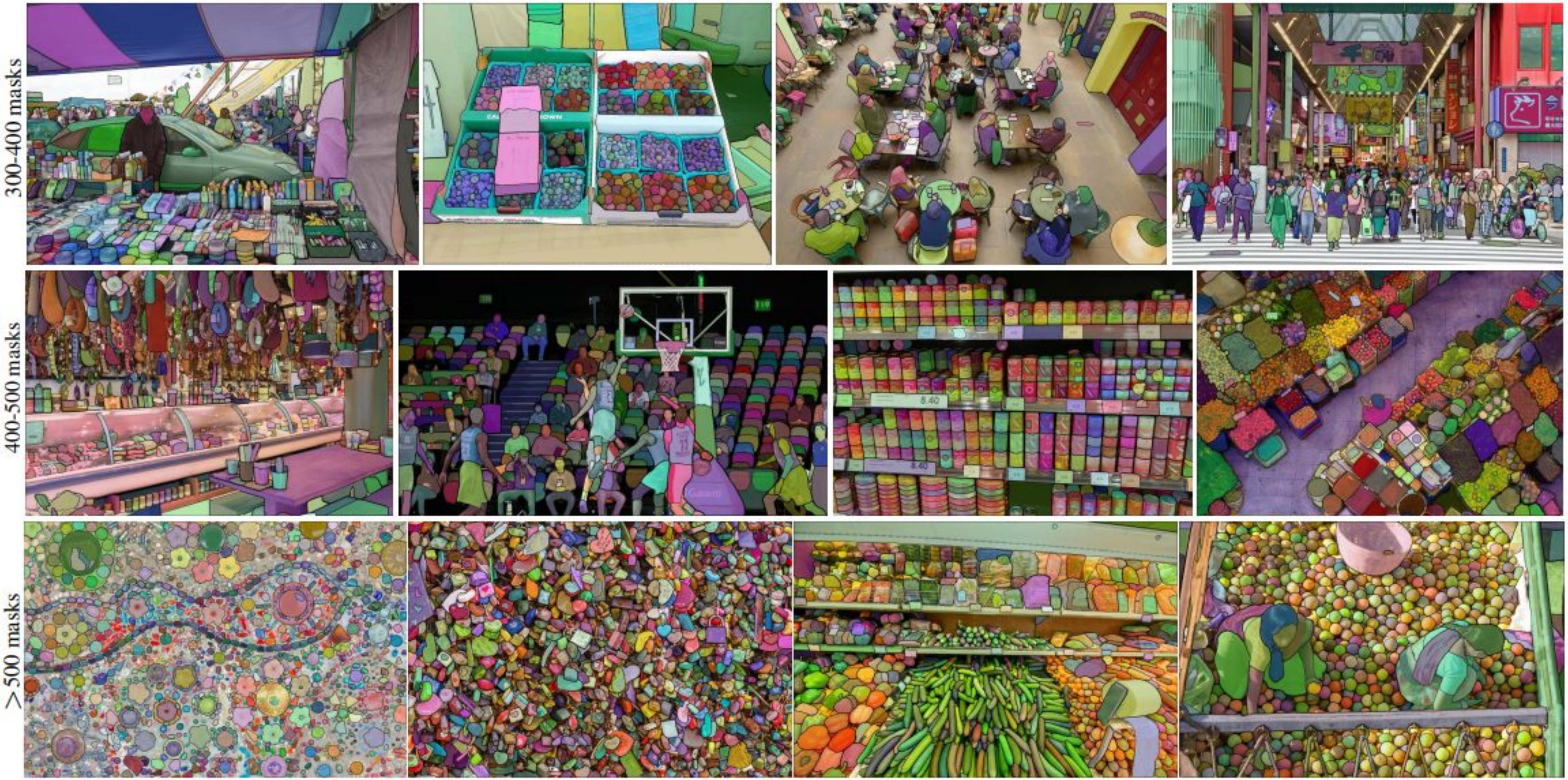
(April 2023)

Segment Anything

Alexander Kirillov[1,2,4] Eric Mintun[2] Nikhila Ravi[1,2] Hanzi Mao[2] Chloe Rolland[3] Laura Gustafson[3]
Tete Xiao[3] Spencer Whitehead Alexander C. Berg Wan-Yen Lo Piotr Dollár[4] Ross Girshick[4]

[1]project lead [2]joint first author [3]equal contribution [4]directional lead

Meta AI Research, FAIR

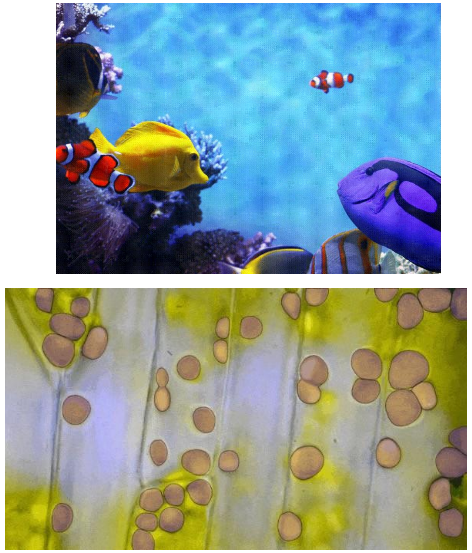

300-400 masks
400-500 masks
>500 masks

Segment Anything Models (SAM)

Segment Anything Model (SAM) [Kirillov et al., 2023]

- A foundation model for image segmentation, *i.e.*, predicting object masks
- SA-1B dataset
  - Web-scale **11M** photography and **1.1B** segmentation masks[1]
- Enables strong zero-shot transfer on new domains
  - e.g., segmenting underwater scenes, or microscopy

SA-1B examples



Zero-shot transfer with SAM

Segment Anything Models (SAM)

Segment Anything Model (SAM) [Kirillov et al., 2023]

- Promptable Segmentation via **points** and **boxes**
  - User can steer the image segmentation, like prompting MLs

- For example, user can prompt regions to be **included** & **excluded** by the model
  - Segmenting the whole image can be done by prompting a grid of points

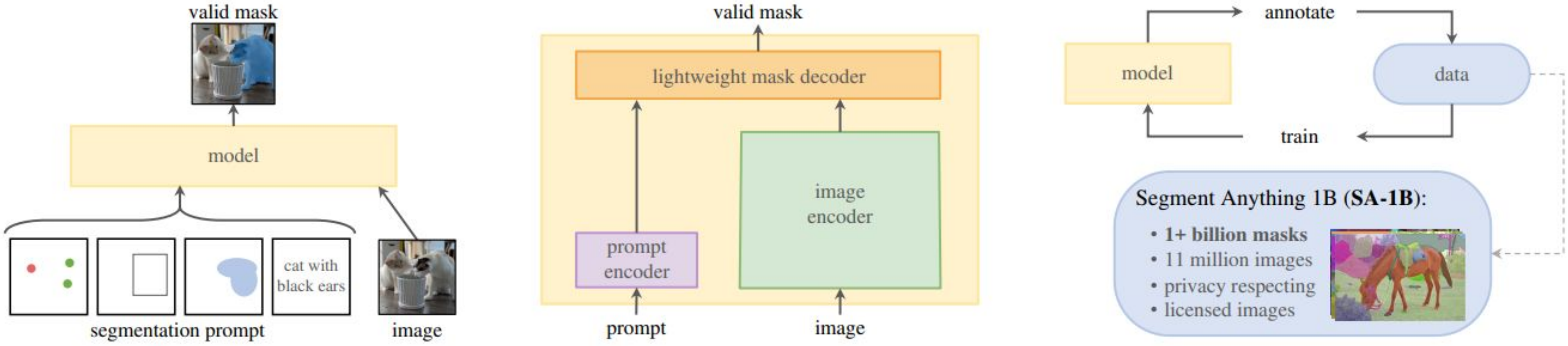
include

Prompt-based
Image Segmentation by SAM

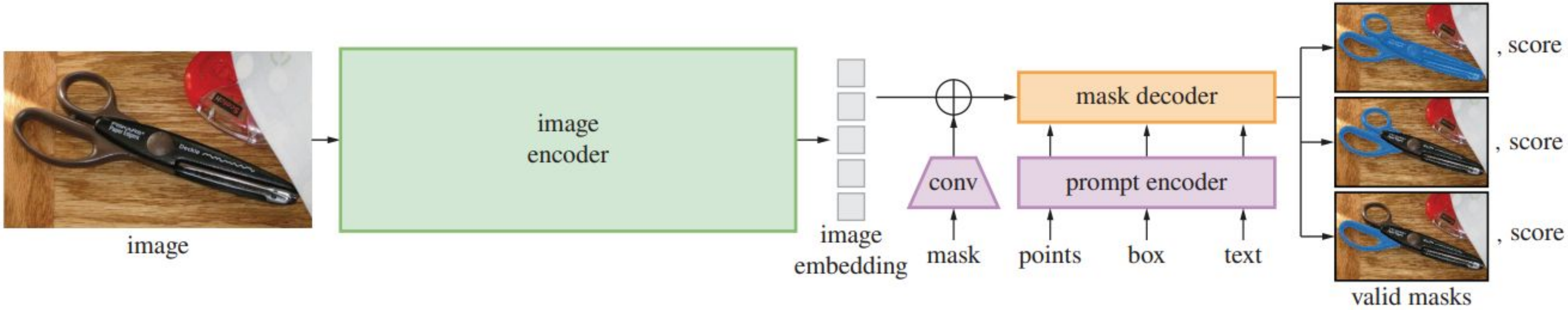
exclude

Segmenting the whole image by prompting a grid of points

SAM



(a) **Task**: promptable segmentation (b) **Model**: Segment Anything Model (**SAM**) (c) **Data**: data engine (top) & dataset (bottom)



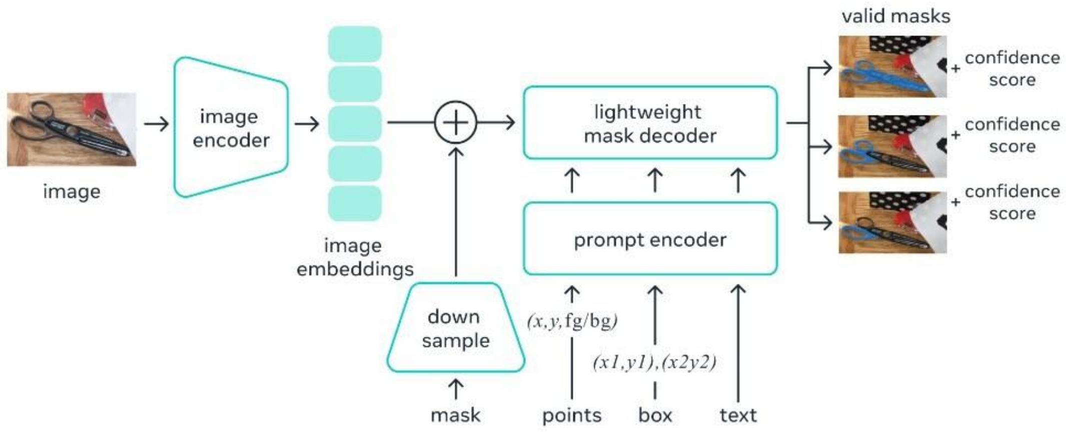
https://segment-anything.com/
https://docs.ultralytics.com/models/sam/
https://arxiv.org/abs/2304.02643
https://github.com/facebookresearch/segment-anything
https://github.com/Hedlen/awesome-segment-anything

Segment Anything Models (SAM)

Component of SAM model

- **Image Encoder**
  - A ViT model producing a one-time embedding for segmentation
  - The embedding can be shared for different prompts
- **Prompt Encoder**
  - Encodes point, box, or text[1] prompts into transformer tokens

- **Mask Decoder**
  - Prompt token and image embedding goes through a transformer decoder
  - Decoder predicts multiple candidates for segmentation mask and the confidence



Note: Text encoding function is not published.

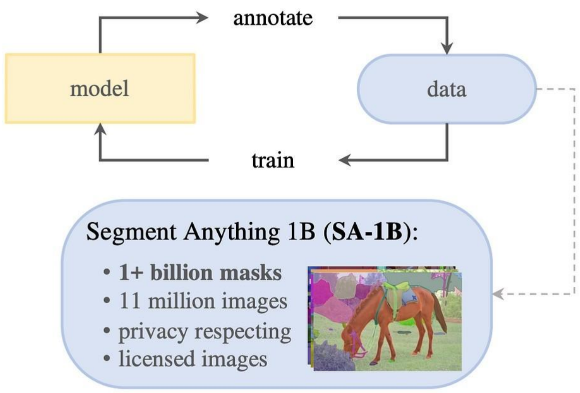
Segment Anything Models (SAM)

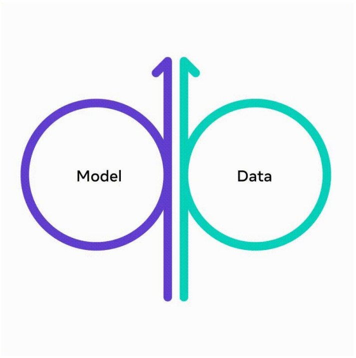
SA-1B dataset

- Web-scale 11M photography and **1.1B segmentation masks**
  - Challenge: manually annotating the images is too expensive

- **Model-in-the-loop design**
  1. The data annotators use and fix SAM's outputs to annotate images (semi- auto)
  2. Newley available annotations are then used to re-train SAM
  3. The process is repeated and SAM's performance is bootstrapped

- Finally, the **automatic annotator (a SAM)** creates the SA-1B dataset



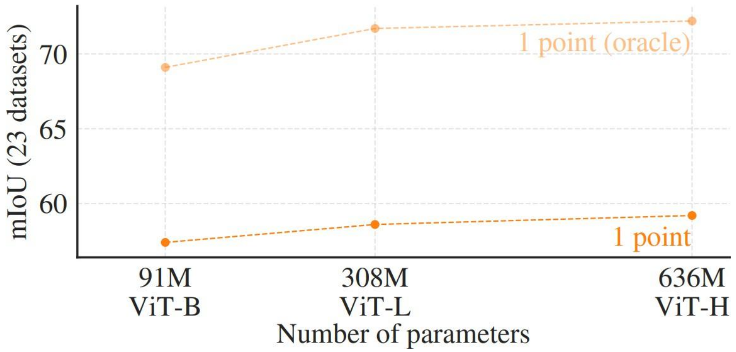


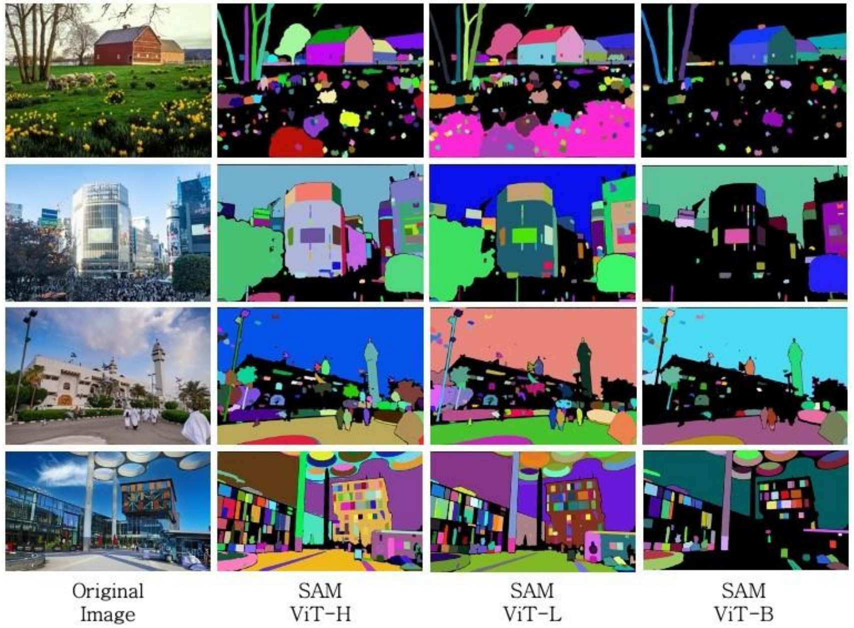
Model-in-the-loop process is repeated +10 times to get the final automatic annotation

Segment Anything Models (SAM)

SAM model variants

- Default variants by the original research paper
  - Considers different image enocders: ViT-B, ViT-L, ViT-H
  - A **direct trade-off** on performance vs. computation cost



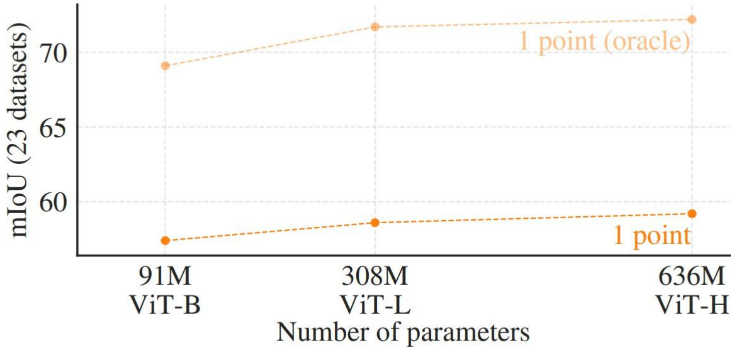


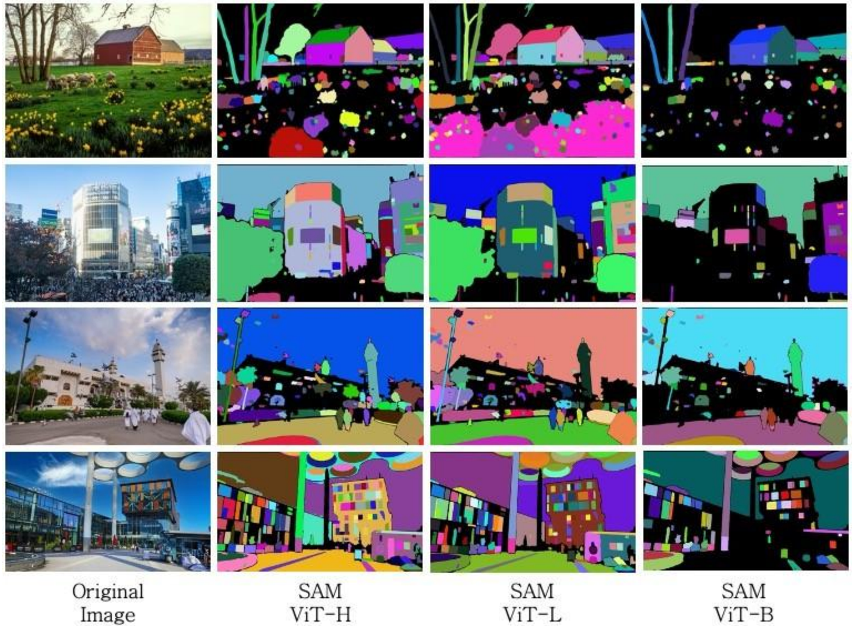
Segment Anything Models (SAM)

SAM model variants

- Default variants by the original research paper
  - Considers different image enocders: ViT-B, ViT-L, ViT-H
  - A **trivial trade-off** on segmentation accuracy vs. computation cost

- More effective way for the efficiency?





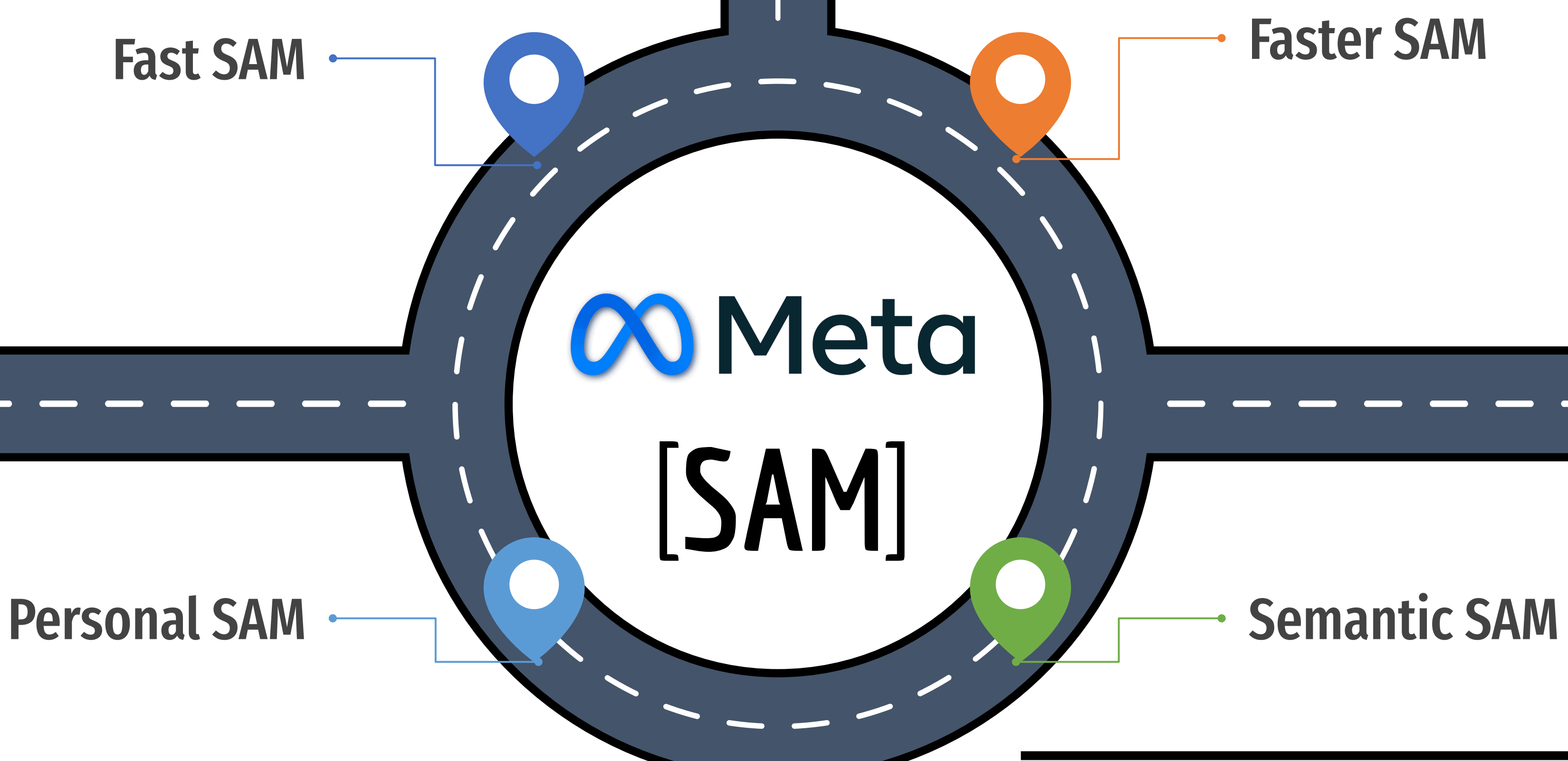
Fast Segment Anything

Xu Zhao [1,3] Wenchao Ding [1,2] Yongqi An [1,2] Yinglong Du [1,2]
Tao Yu [1,2] Min Li [1,2] Ming Tang [1,2] Jinqiao Wang [1,2,3,4]
Institute of Automation, Chinese Academy of Sciences, Beijing, China[1]
School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China[2]
Objecteye Inc., Beijing, China[3]
Wuhan AI Research, Wuhan, China[4]

FASTER SEGMENT ANYTHING: TOWARDS LIGHTWEIGHT SAM FOR MOBILE APPLICATIONS

A PREPRINT

Chaoning Zhang* Kyung Hee University
Dongshen Han Kyung Hee University
Yu Qiao Kyung Hee University
Jung Uk Kim Kyung Hee University
Sung-Ho Bae Kyung Hee University
Seungkyu Lee Kyung Hee University
Choong Seon Hong Kyung Hee University



Personalize Segment Anything Model with One Shot

Renrui Zhang[1,2], Zhengkai Jiang[3], Ziyu Guo[1], Shilin Yan[1], Junting Pan[2]
Hao Dong[4], Peng Gao[1], Hongsheng Li[2]
[1]Shanghai Artificial Intelligence Laboratory [2]CUHK MMLab
[3]Tencent Youtu Lab [4]CFCS, School of CS, Peking University

Semantic-SAM: Segment and Recognize Anything at Any Granularity

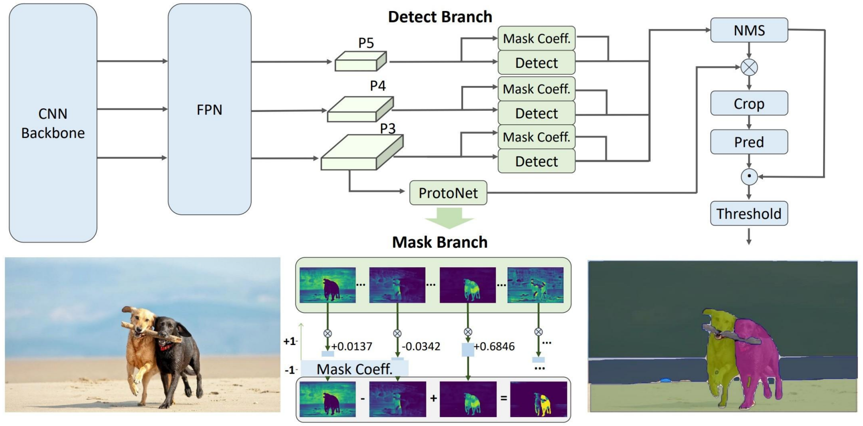
Feng Li♠*, Hao Zhang♠*, Peize Sun♯, Xueyan Zou§, Shilong Liu¶, Chunyuan Li‡
Jianwei Yang‡1, Lei Zhang†2, Jianfeng Gao‡2
♠ HKUST ‡ Microsoft Research, Redmond † IDEA ♯ HKU § UW-Madison ¶ Tsinghua
* Equal Contribution 1. Project Lead 2. Equal Advisory Contribution

186

Segment Anything Models (SAM)

FastSAM [Zhao et al., 2023]

- Trains SA-1B on a CNN-based architecture for image segmentation (YoLo v7)
- Predicts all possible masks at once, without conditioning on prompts
  - (+) Better parallization on the GPUs (Running time is independent to the number of points)
  - (−) Does not learn to utilize user prompts, e.g., points, boxes



YoLo architecture predicts all image segmentations at once

method	params	Running Speed under Different Point Prompt Numbers (ms)					
		1	10	100	E(16×16)	E(32×32*)	E(64×64)
SAM-H [20]	0.6G	446	464	627	852	2099	6972
SAM-B [20]	136M	110	125	230	432	1383	5417
FastSAM (Ours)	68M	40					

Segment Anything Models (SAM)

MobileSAM [Zhang et al., 2023]

- Downsizing the image encoder through **Knowledge Distillation** [Hinton et al., 2015]
- Parameters: 611M (ViT-H) → 5M (tiny transformer)

- Image embedding space tends to be similar after knowledge distillation
  - Can perform well close to the original SAM
  - Realtime inference 452ms (Original SAM) → 8ms (MobileSAM)

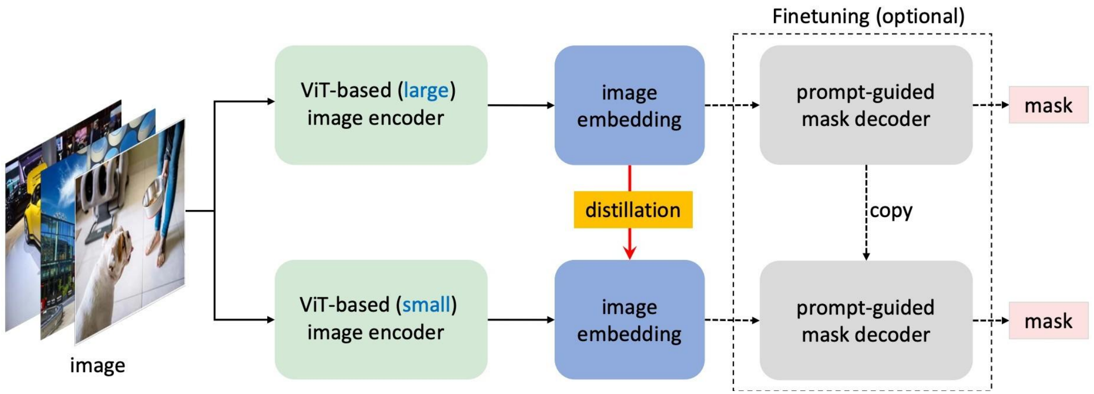


Image encoder is distillated, with a frozen mask decoder

(a) Image (b) MobileSAM (c) Original SAM

Segment Anything Models (SAM)

SAM-HQ [Ke et al., 2023]

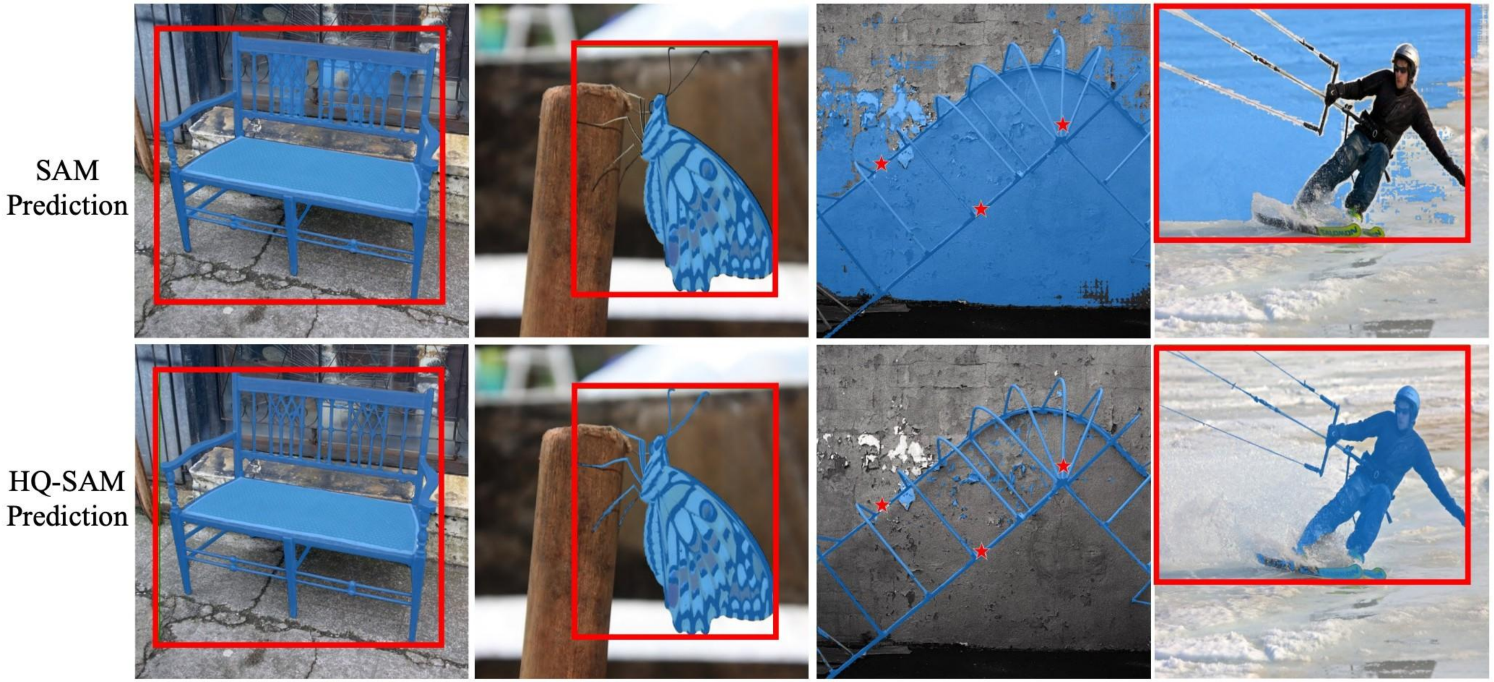
- Identifies the weakness of SAM and SA-1B dataset
  - Failures on objects with intricate structures (e.g., grate patterns)

SAM has weakness on intricate structures, which gets fixed by HQ-SAM [Ke et al., 2023]

Segment Anything Models (SAM)

SAM-HQ [Ke et al., 2023]

- SAM-HQ introduces fine-tuning to mitigate the failure cases (**HQSeg-44K dataset**)
  - Custom collection of 44K images, with extremely intricate segmentation annotations



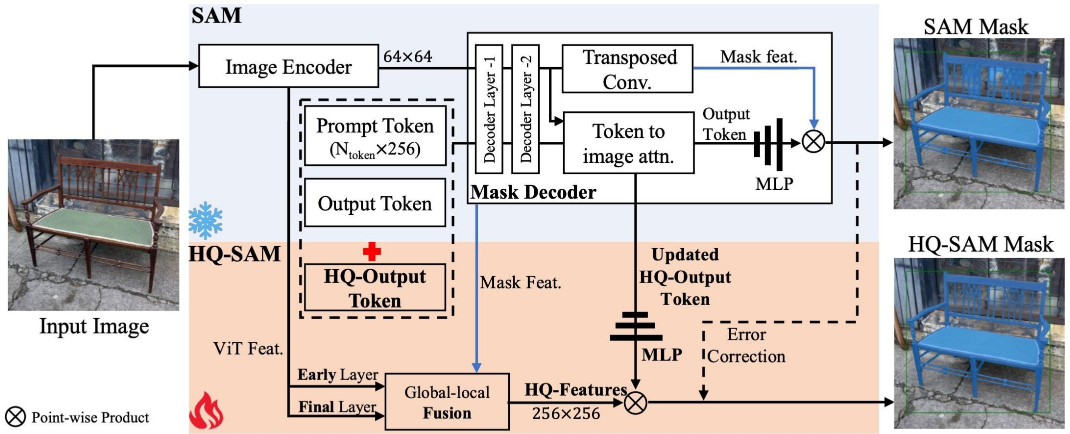
SAM vs. HQ-SAM on HQSeq-44k samples

Segment Anything Models (SAM)

SAM-HQ [Ke et al., 2023]

- The pretrained SAM parameters remain frozen
  - Prevents model overfitting or catastrophic forgetting by a small HQSeg-44K dataset

- SAM-HQ only introduces a tunable prompt token and MLPs for fusion
  - Requires training only 5.1M additional parameters (0.5% of the SAM's parameters)



HQ-SAM architecture

Method	Learnable Params (M)	Training # GPU	Training Batch Size	Training Time (h)	Inference FPS	Inference Mem.
SAM [21]	1191	128	128	N/A	5.0	7.6G
HQ-SAM	**5.1**	**8**	**32**	**4**	**4.8**	**7.6G**

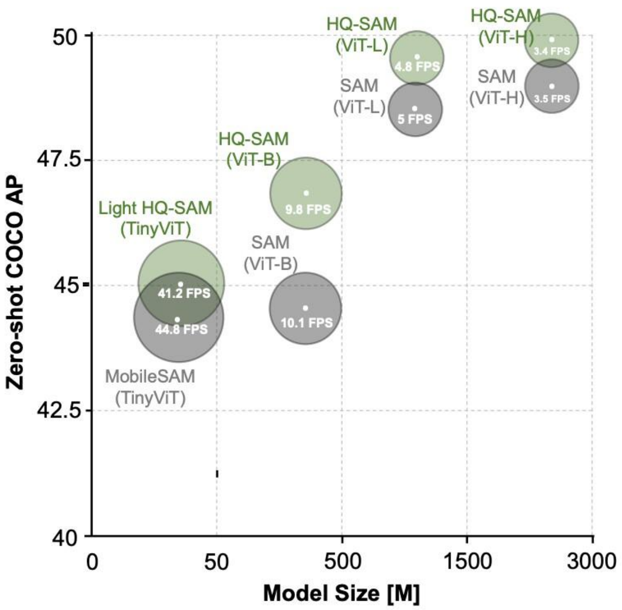
Training cost of HQ-SAM

Segment Anything Models (SAM)

SAM-HQ [Ke et al., 2023]

- The pretrained SAM parameters remain frozen
    - Prevents model overfitting or catastrophic forgetting by a small HQSeg-44K dataset

- SAM-HQ only introduces a tunable prompt token and MLPs for fusion
    - Brings simple and effective performance boosts on all existing SAM variants
    - Including VIT-H, ViT-L, ViT-B and MobileSAM [Zhang et al., 2023]



Zero-shot performances on MS-COCO

Segment Anything Models (SAM)

Notable Applications of SAM

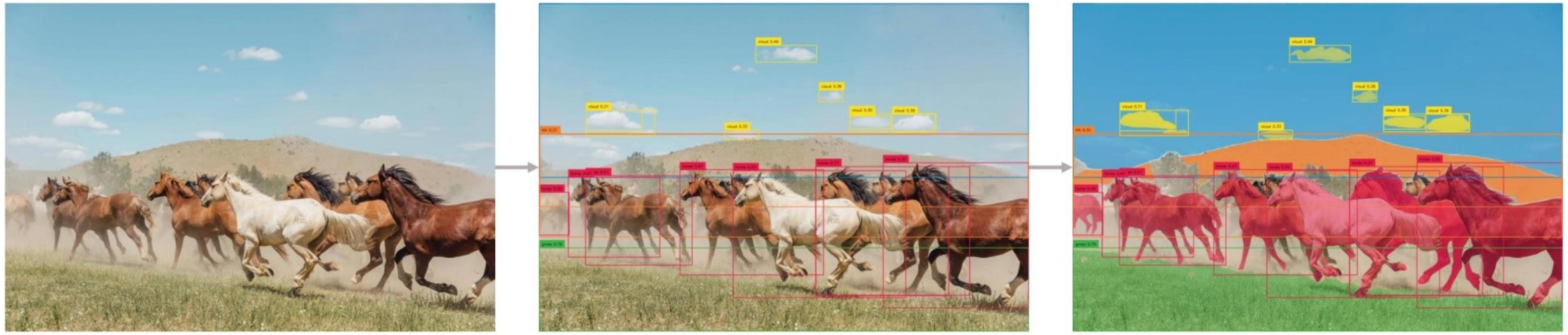
- Open-Vocabulary Semantic Segmentation (*e.g.,* Grounded SAM [Liu et al., 2023])

  Basic Idea: **prompting SAM** with boxes, via **text-prompted box predictors**

  - Recent **vision-language models** can make **zero-shot box predictions** at ease

    e.g., GroundingDINO [Liu et al., 2023], ViLD [Gu et al., 2022]

  - However, **zero-shot semantic segmentation** has remained **challenging**
    - SAM directly **escalates** the semantic box predictions → segmentation masks
    - A break-through in the zero-shot, open vocabulary, semantic segmentation task



Text Prompt: **"Horse. Clouds. Grasses. Sky. Hill."** | **Grounding DINO:** **Detect Everything** | **Grounded-SAM:** **Detect and Segment Everything**

Recognize Anything: A Strong Image Tagging Model

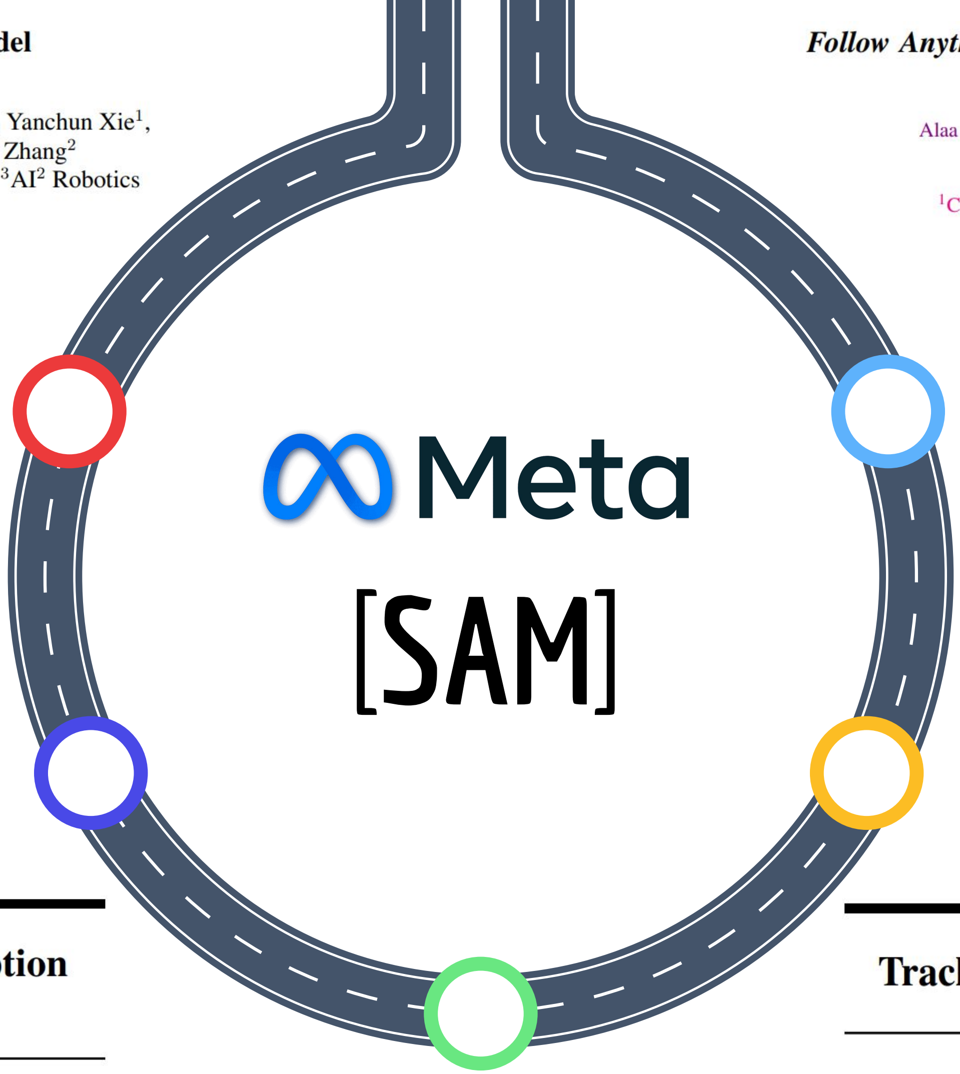
Youcai Zhang*[1], Xinyu Huang*[1], Jinyu Ma*[1], Zhaoyang Li*[1], Zhaochuan Luo[1], Yanchun Xie[1],
Yuzhuo Qin[1], Tong Luo[1], Yaqian Li[1], Shilong Liu[2], Yandong Guo[3], Lei Zhang[2]
[1]OPPO Research Institute, [2]International Digital Economy Academy (IDEA), [3]AI² Robotics
*Equal Contribution

***Follow Anything*: Open-set detection, tracking, and following in real-time**

Alaa Maalouf[1,*], Ninad Jadhav[2], Krishna Murthy Jatavallabhula[1], Makram Chahine[1],
Daniel M.Vogt[2], Robert J. Wood[2], Antonio Torralba[1], and Daniela Rus[1]
[1]CSAIL, MIT [2]SEAS, Harvard University *Correspondence: alaam@mit.edu

RAM

FAN



CAM

TAM

Caption Anything: Interactive Image Description with Diverse Multimodal Controls

Teng Wang*†, Jinrui Zhang*, Junjie Fei*, Hao Zheng, Yunlong Tang, Zhe Li,
Mingqi Gao, Shanshan Zhao
SUSTech VIP Lab

Track Anything: Segment Anything Meets Videos

Jinyu Yang*, Mingqi Gao*, Zhe Li*, Shang Gao, Fangjing Wang, Feng Zheng†
SUSTech VIP Lab

MAM

Matting Anything

Jiachen Li[1], Jitesh Jain[1], Humphrey Shi[1,2]
[1]SHI Labs @ UIUC & Oregon, [2]Picsart AI Research (PAIR)

194

Segment Anything Models (SAM)

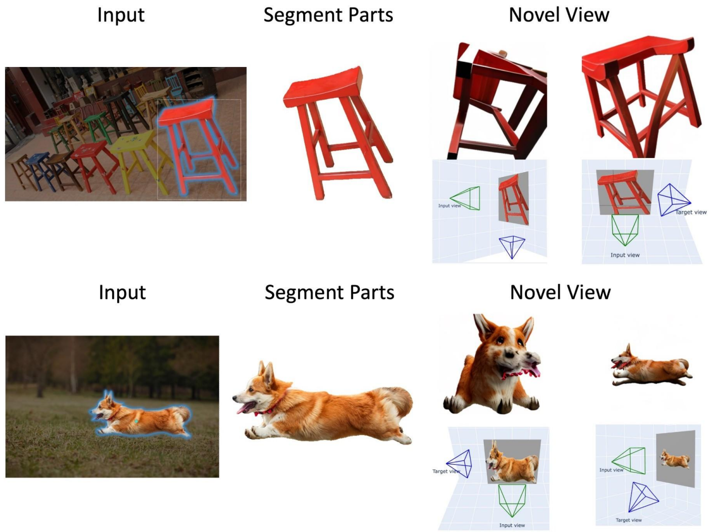
Conclusion

Segment Anything Model, a **foundation model** in Vision AI

- Trained on a **web-scale** dataset of 11M images & 1B+ masks
- **Adaptable** to wide range of image domains & tasks via user prompts

Foundation Model = *scale & flexibility*

SA-1B



Anything 3D [NUS team, 2023]

Table of Contents

1. **Problem**

2. **Introduction**
   - Foundation models in vision tasks

3. **Self-supervised Learning**
   - **Discriminative Visual Foundation Models**
     - Contrastive learning [SimCLR, MoCo, …]
     - Self Distillation[BYOL, DINO, …]

   - **Generative Visual foundation models**
     - Mask Auto-Encoder [**MAE**]

   - **Evaluation**

4. **Multi-Modal Self-supervised Learning [CLIP]**
   - Image-text Contrastive Learning

5. **Segment Anything [SAM]**

Conclusion

Foundation Models?

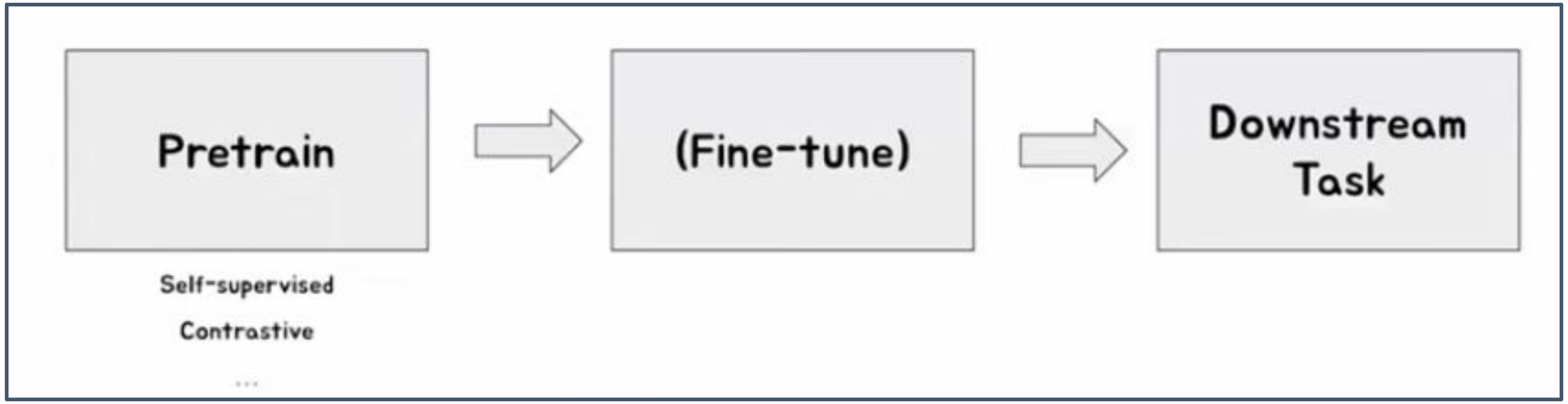
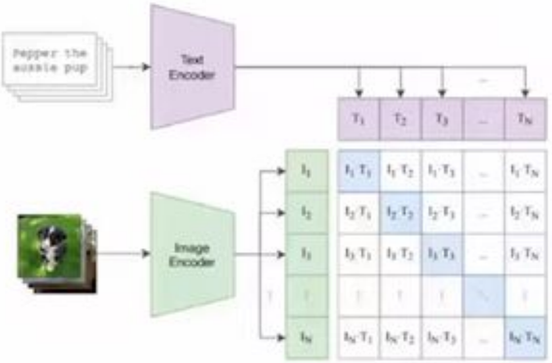
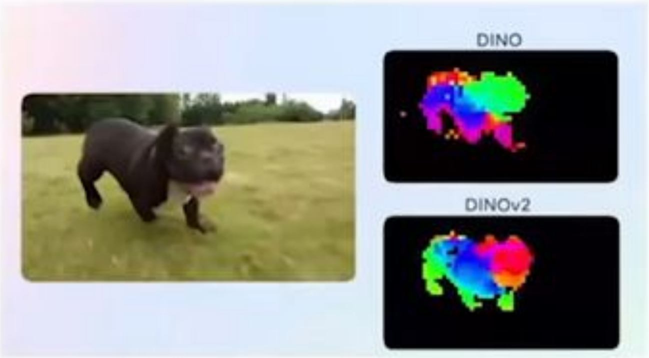


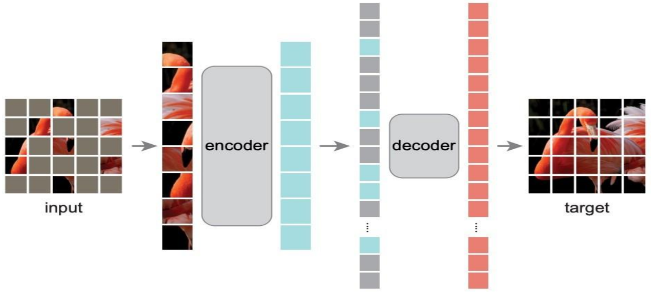
Image Foundation Models



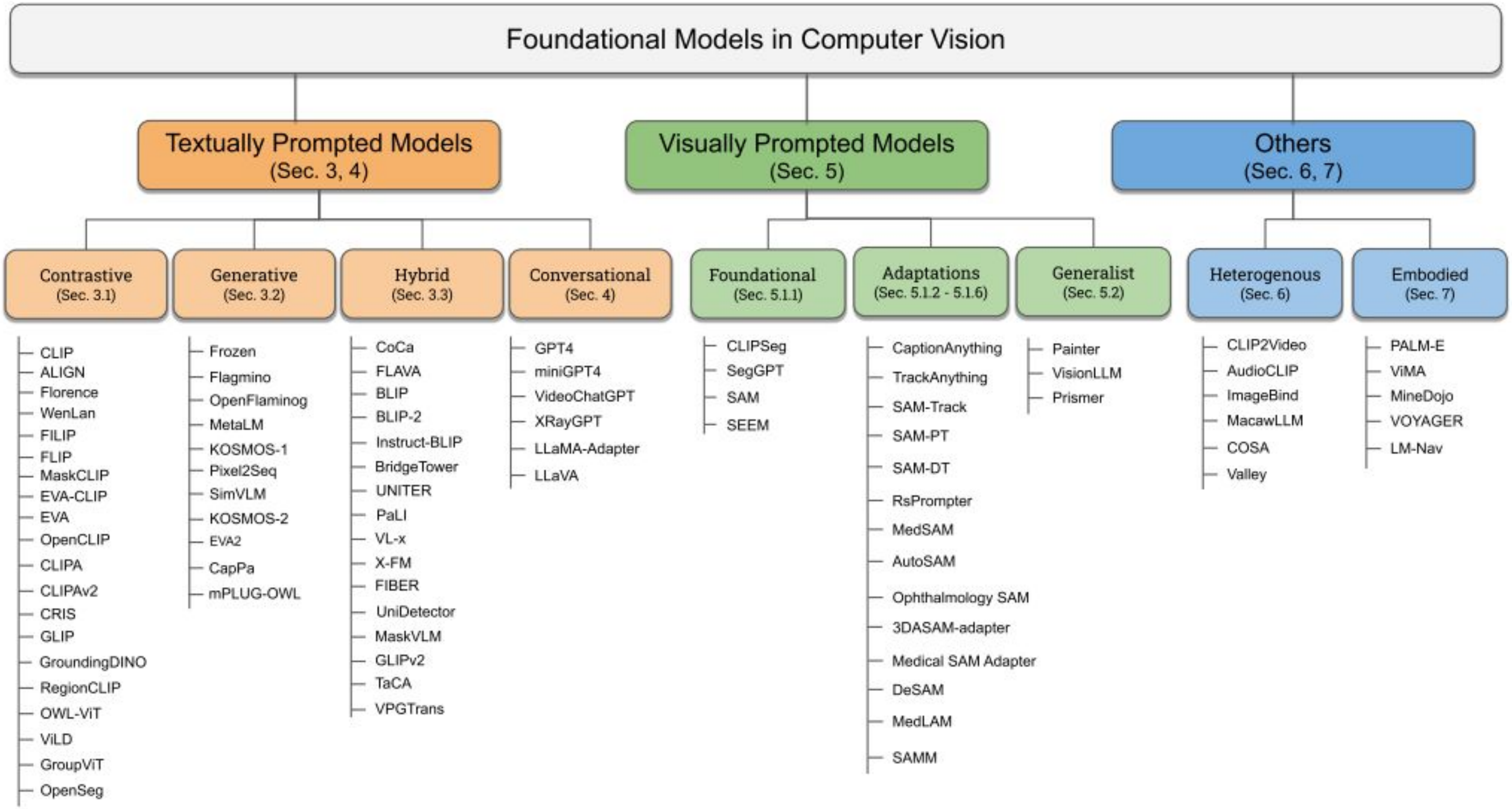
CLIP



DINO v2



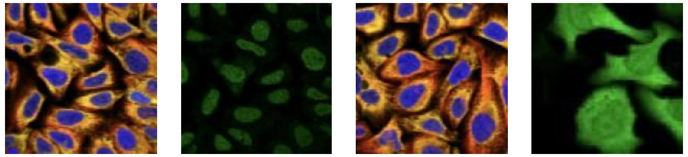
MAE


Foundational Models in Computer Vision
Textually Prompted Models (Sec. 3, 4)
Visually Prompted Models (Sec. 5)
Others (Sec. 6, 7)
Contrastive (Sec. 3.1)
Generative (Sec. 3.2)
Hybrid (Sec. 3.3)
Conversational (Sec. 4)
Foundational (Sec. 5.1.1)
Adaptations (Sec. 5.1.2 - 5.1.6)
Generalist (Sec. 5.2)
Heterogenous (Sec. 6)
Embodied (Sec. 7)
CLIP
ALIGN
Florence
WenLan
FILIP
FLIP
MaskCLIP
EVA-CLIP
EVA
OpenCLIP
CLIPA
CLIPAv2
CRIS
GLIP
GroundingDINO
RegionCLIP
OWL-ViT
ViLD
GroupViT
OpenSeg
Frozen
Flagmino
OpenFlaminog
MetaLM
KOSMOS-1
Pixel2Seq
SimVLM
KOSMOS-2
EVA2
CapPa
mPLUG-OWL
CoCa
FLAVA
BLIP
BLIP-2
Instruct-BLIP
BridgeTower
UNITER
PaLI
VL-x
X-FM
FIBER
UniDetector
MaskVLM
GLIPv2
TaCA
VPGTrans
GPT4
miniGPT4
VideoChatGPT
XRayGPT
LLaMA-Adapter
LLaVA
CLIPSeg
SegGPT
SAM
SEEM
CaptionAnything
TrackAnything
SAM-Track
SAM-PT
SAM-DT
RsPrompter
MedSAM
AutoSAM
Ophthalmology SAM
3DASAM-adapter
Medical SAM Adapter
DeSAM
MedLAM
SAMM
Painter
VisionLLM
Prismer
CLIP2Video
AudioCLIP
ImageBind
MacawLLM
COSA
Valley
PALM-E
ViMA
MineDojo
VOYAGER
LM-Nav

Domain gap…

- For example, open source SSL models is pre-trained on natural looking images:

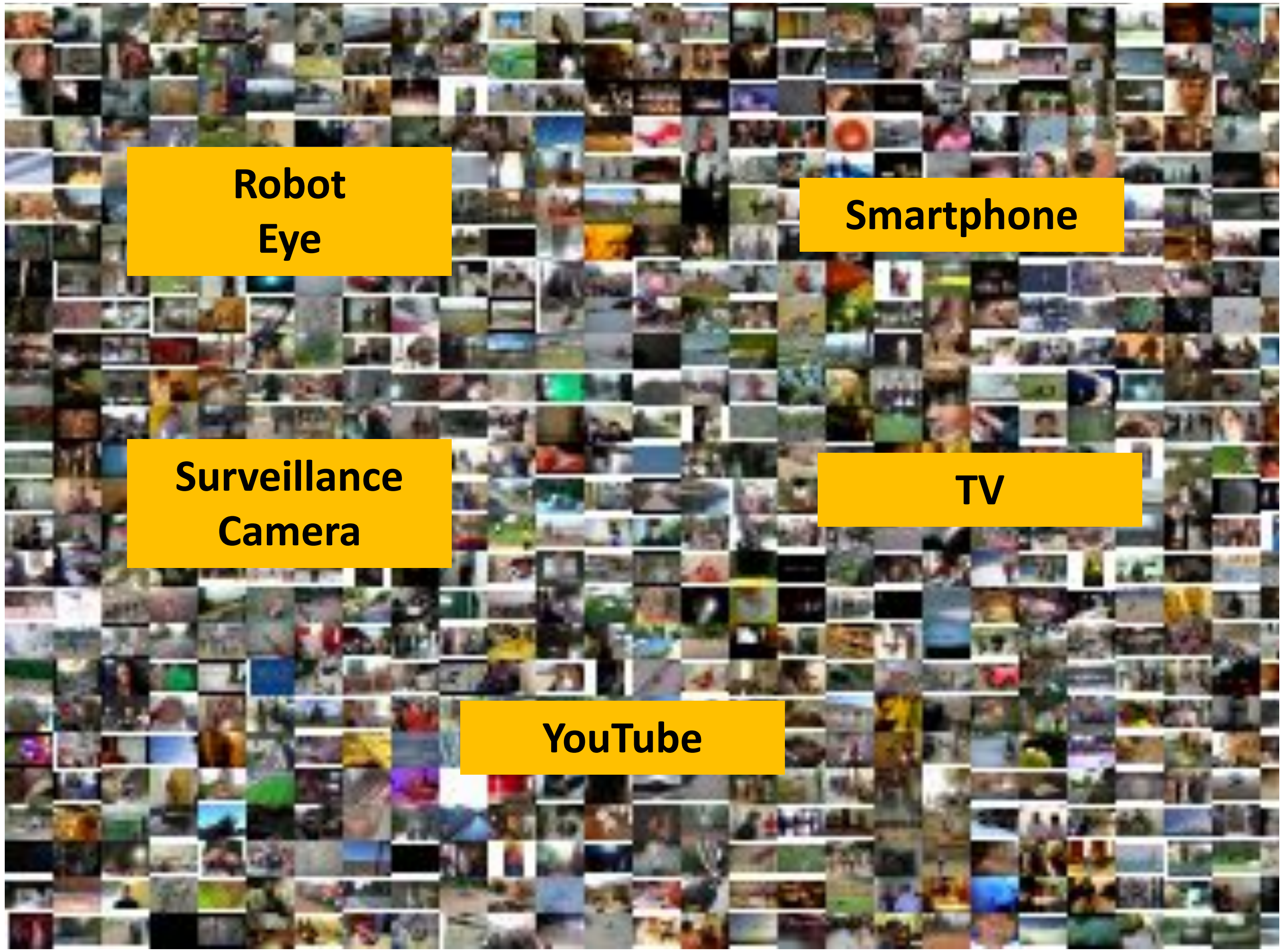
- **But**, your data looks like this:



Solution: Fine-tune SSL pretrained model using on your data

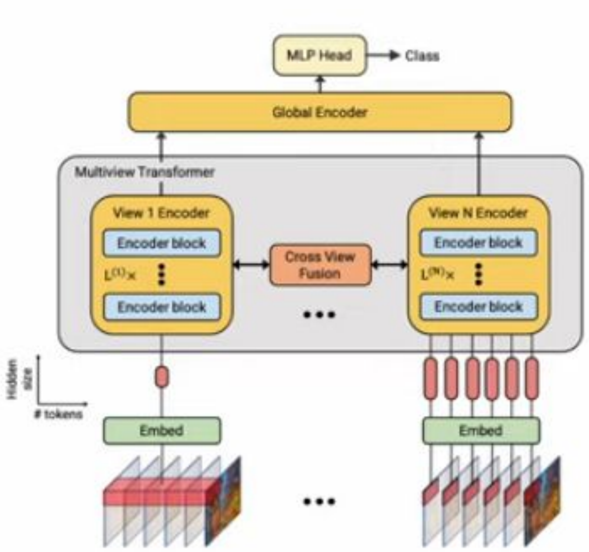
Videos

Tremendous videos and media contents can be obtained from:

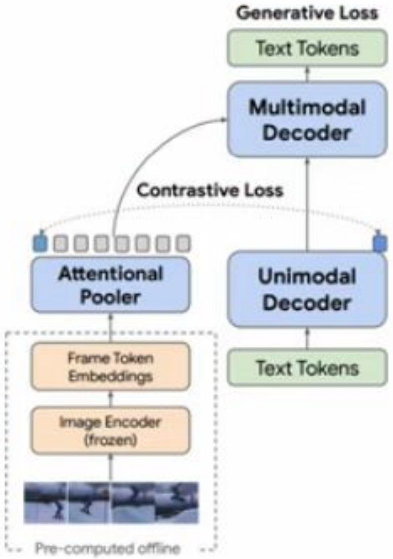


Video understanding is an important research topic.

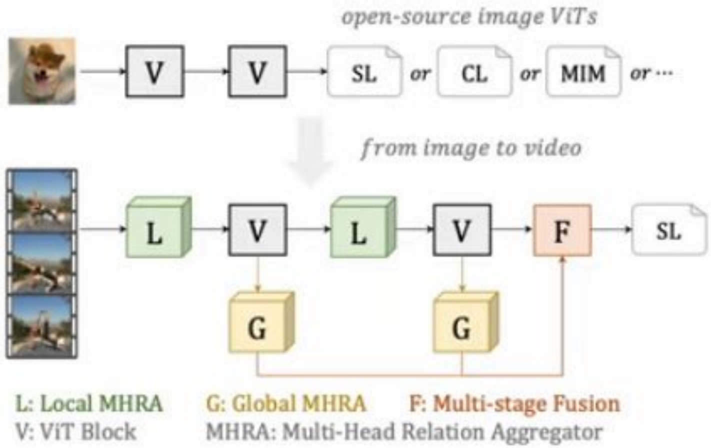
Build VFM on the top of IFM



MTV
Google CVPR2022

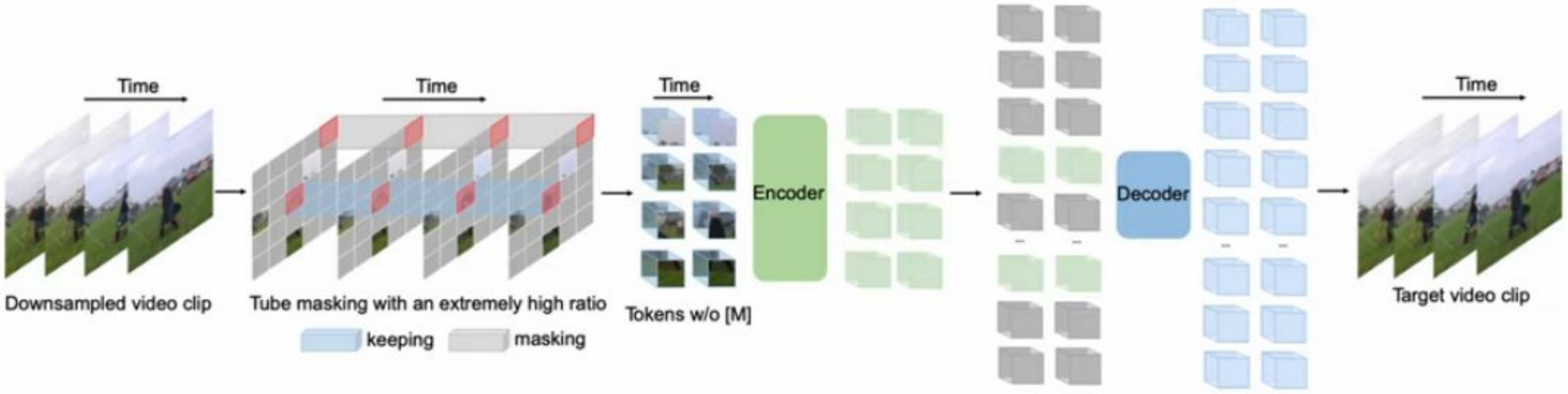


VideoCoCa
Google Arxiv2023



UniFormerV2
Shanghai AI Lab ICCV2023

Build VFM by learning from scratch with MAE



VideoMAE
NJU NeurIPS2022

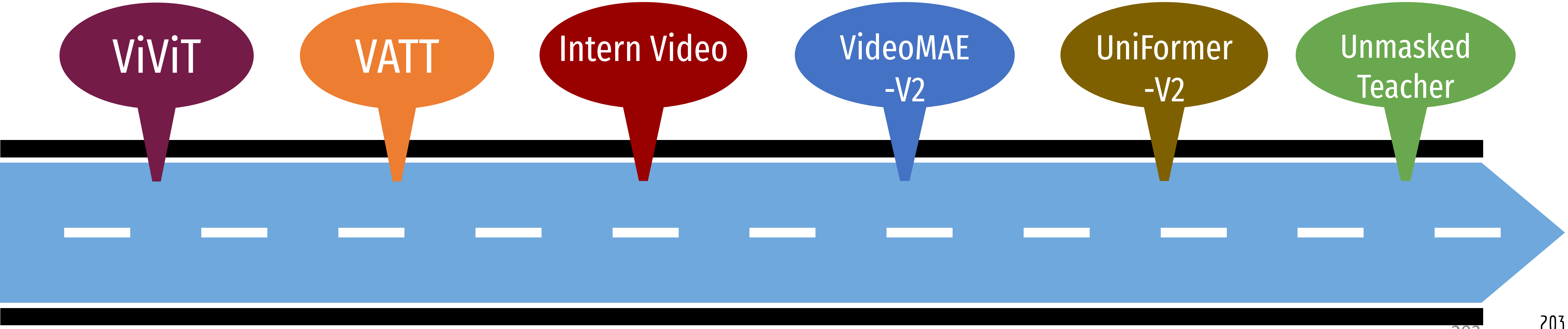
ST-MAE
Meta NeurIPS2022

Video Foundation Models

Unmasked Teacher: Towards Training-Efficient Video Foundation Models

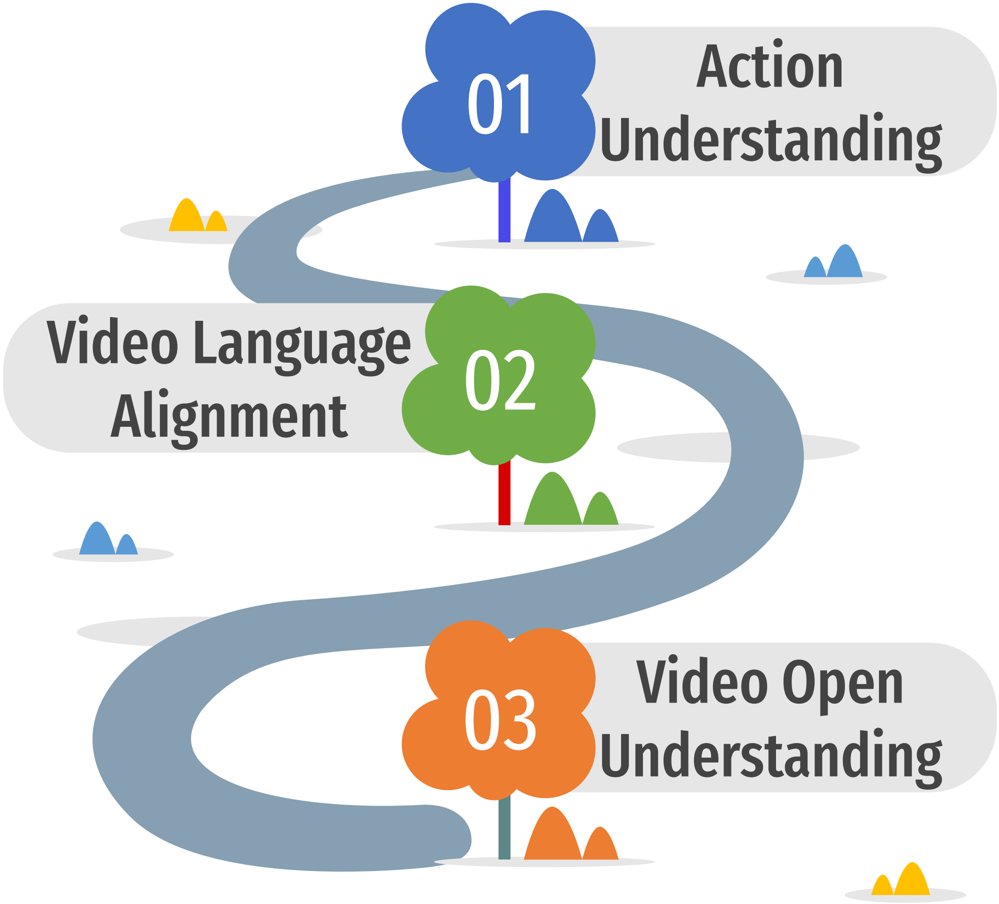
Kunchang Li[1,2,3*] Yali Wang[1,3†] Yizhuo Li[4,3*] Yi Wang[3] Yinan He[3]
Limin Wang[5,3] Yu Qiao[3,1†]

[1]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences [3]Shanghai AI Laboratory [4]The University of Hong Kong
[5]State Key Laboratory for Novel Software Technology, Nanjing University



203

Downstream Tasks



1. Action Recognition
2. Temporal Action Localization
3. Spatiotemporal Action Localization

1. Video Retrieval
2. Video Question Answering
3. Visual Language Navigation

1. Zero-shot Action Recognition
2. Zero-shot Multiple Choice
3. Open-set Action Recognition

References

[He et al., 2020] Momentum Contrast for Unsupervised Visual Representation Learning, CVPR 2020 [Chen et al., 2020] A Simple Framework for Contrastive Learning of Visual Representations, ICML 2020 [Grill et al., 2020] Bootstrap your own latent: A new approach to self-supervised Learning, NeurIPS 2020 [Caron et al., 2021] Emerging Properties in Self-Supervised Vision Transformers, ICCV 2021

[Bao et al., 2022] BEiT: BERT Pre-Training of Image Transformers, ICLR 2022
[He et al., 2022] Masked Autoencoders Are Scalable Vision Learners, CVPR 2022 [Zhou et al., 2022] ibot: Image bert pre-training with online tokenizer, ICLR 2022

[Baevski et al., 2022] data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language, 2022

[Oquab et al., 2023] DINOv2: Learning Robust Visual Features without Supervision, 2023

[Radford et al., 2021] Learning Transferable Visual Models From Natural Language Supervision, ICML 2021

[Schuhmann et al., 2022] Laion-5b: An open large-scale dataset for training next generation image-text models, NeurIPS 2022

[Fang et al., 2022] Data Determines Distributional Robustness in Contrastive Language Image Pre-training (CLIP), 2022

[Zhai et al., 2023] Sigmoid Loss for Language Image Pre-Training, ICCV 2023

[Mehdi, et al. 2023] Reproducible scaling laws for contrastive language-image learning., CVPR 2023

References

[Hertz et al., 2022] Prompt-to-Prompt Image Editing with Cross Attention Control, ICLR 2023 [Brooks et al., 2022] InstructPix2Pix:

Learning to Follow Image Editing Instructions, CVPR 2023 [Zhang et al., 2023] Adding Conditional Control to Text-to-Image Diffusion

Models, ICCV 2023

[Gal et al., 2022] An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion

[Ruiz et al., 2022] DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation, CVPR 2023

[Ruiz et al., 2023] HyperDreamBooth: HyperNetworks for Fast Personalization of Text-to-Image Models, 2023 [Poole et al., 2022] DreamFusion:

Text-to-3D using 2D Diffusion, ICLR 2023

[Kirillov et al., 2023] Segment Anything.

[Yang et al., 2023] SAM3D: Segment Anything in 3D Scenes.

[Liu et al., 2023] Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection [Gu et al., 2022] Open-vocabulary

Object Detection via Vision and Language Knowledge Distillation. ICLR 2022 [NUS team, 2023] Anything-3D: Towards Single-view Anything

Reconstruction in the Wild.

References

1. https://www.slideshare.net/slideshow/yurii-pashchenko-zeroshot-learning-capabilities-of-clip-model-from-openai/250528753
2. https://www.slideshare.net/slideshow/selfsupervised-learning-lecture-note/251837047
3. https://www.slideshare.net/slideshow/lecture16selfsupervisedlearningpptx/257422258
4. https://www.slideshare.net/slideshow/introduction-to-self-supervised-learning-kuliah-machine-learning-stei-itb/273221174
5. https://alinlab.kaist.ac.kr/ai602_2023.html

Thank you!