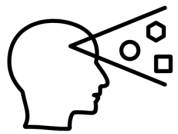
IMEN574/IDSC574 Programming for Data Science

17. **Vision Models I**

Spring 2026

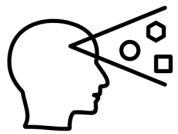
Instructor: **Sangwoo Mo**

FM Landscape

Perception	Generation	Action
		

FM Landscape

	Perception	Generation	Action
Vision			
Language			







Visual Perception

	Perception	Generation	Action
Vision			
Language			

Visual Perception

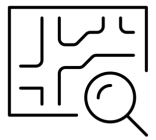
Abstract Perception

“what” (high-level, conceptual)



Dense Perception

“where” (low-level, spatial)



Abstract Perception

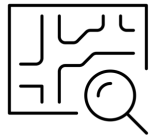
Abstract Perception

“what” (high-level, conceptual)

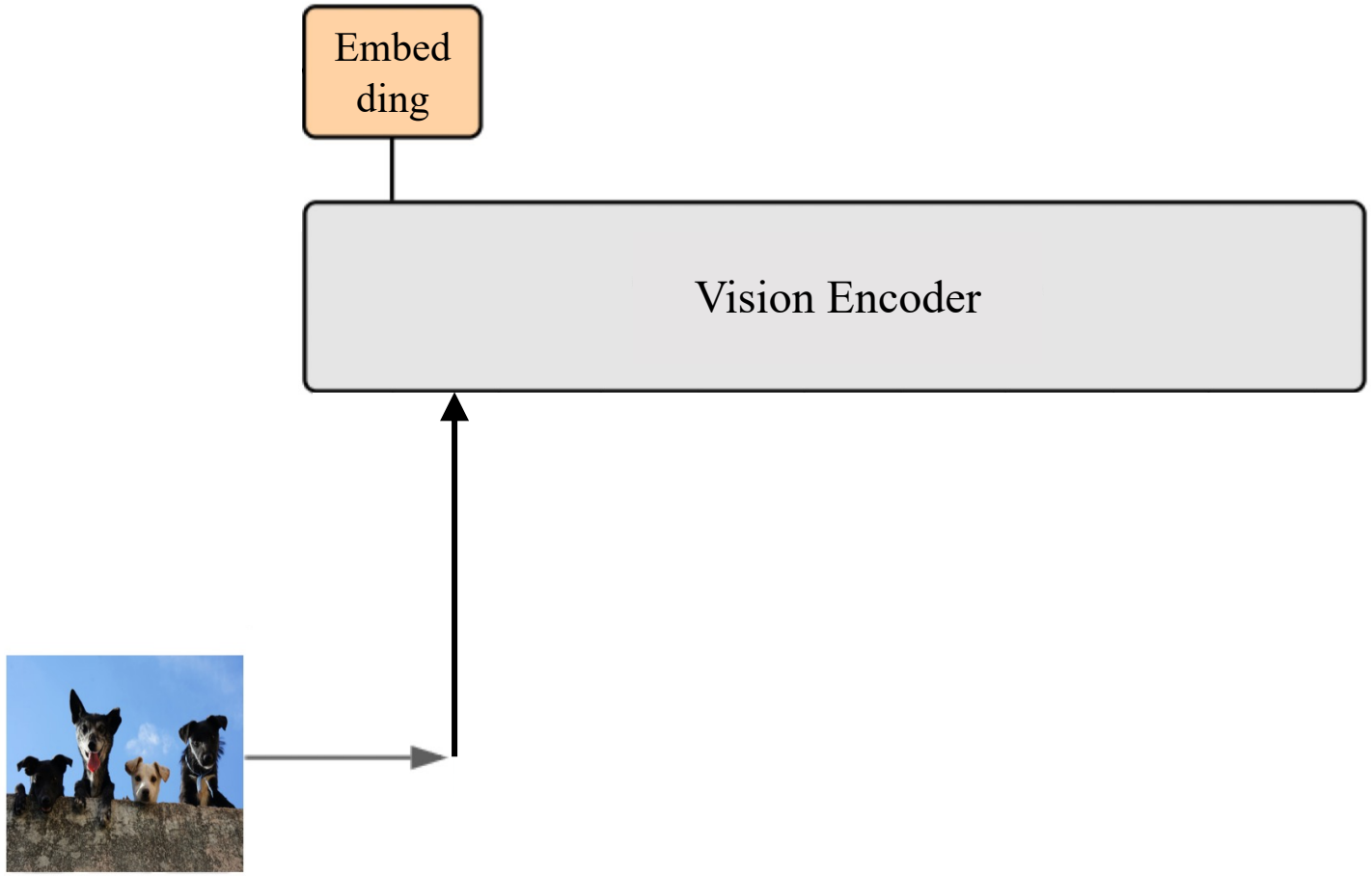


Dense Perception

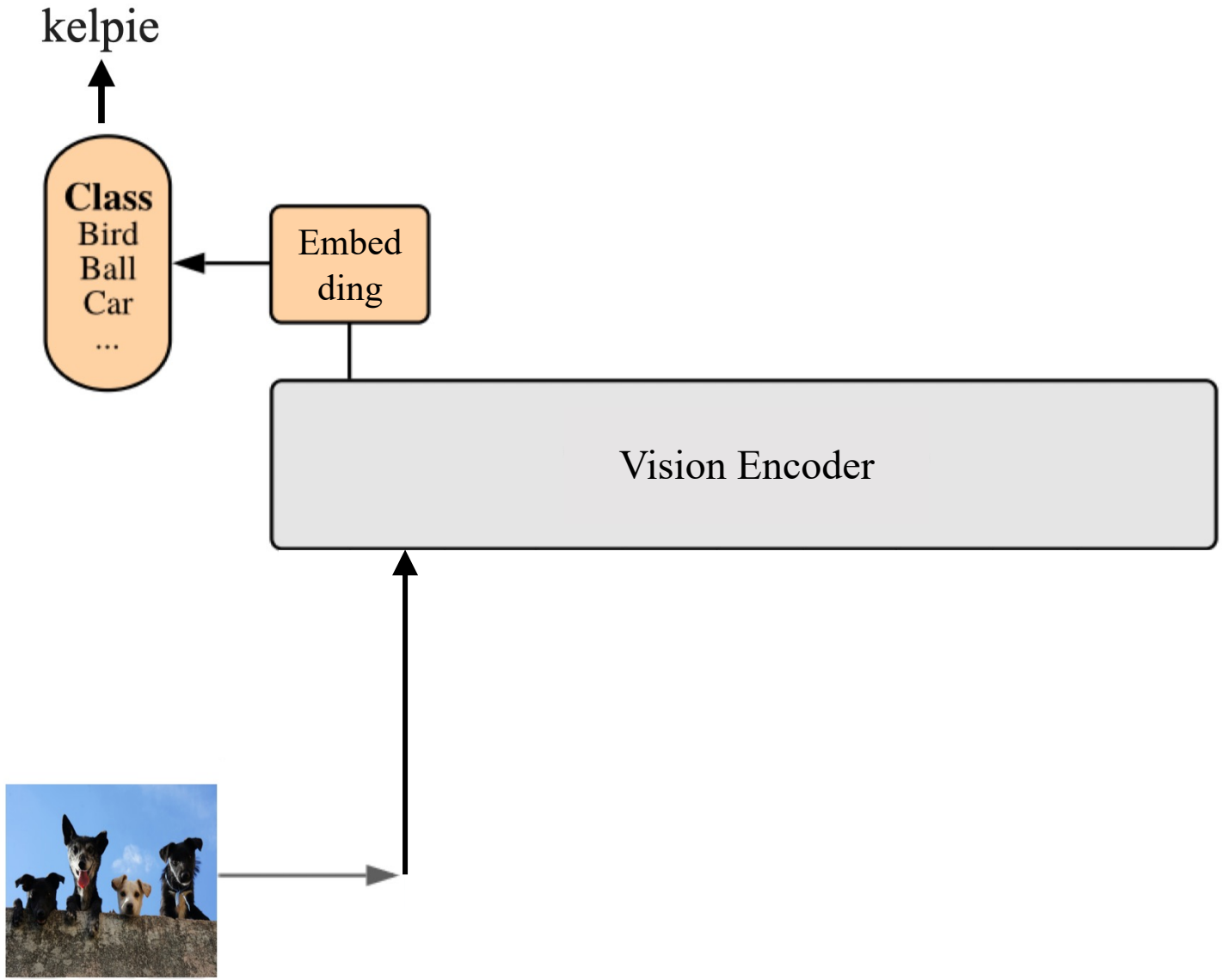
“where” (low-level, spatial)



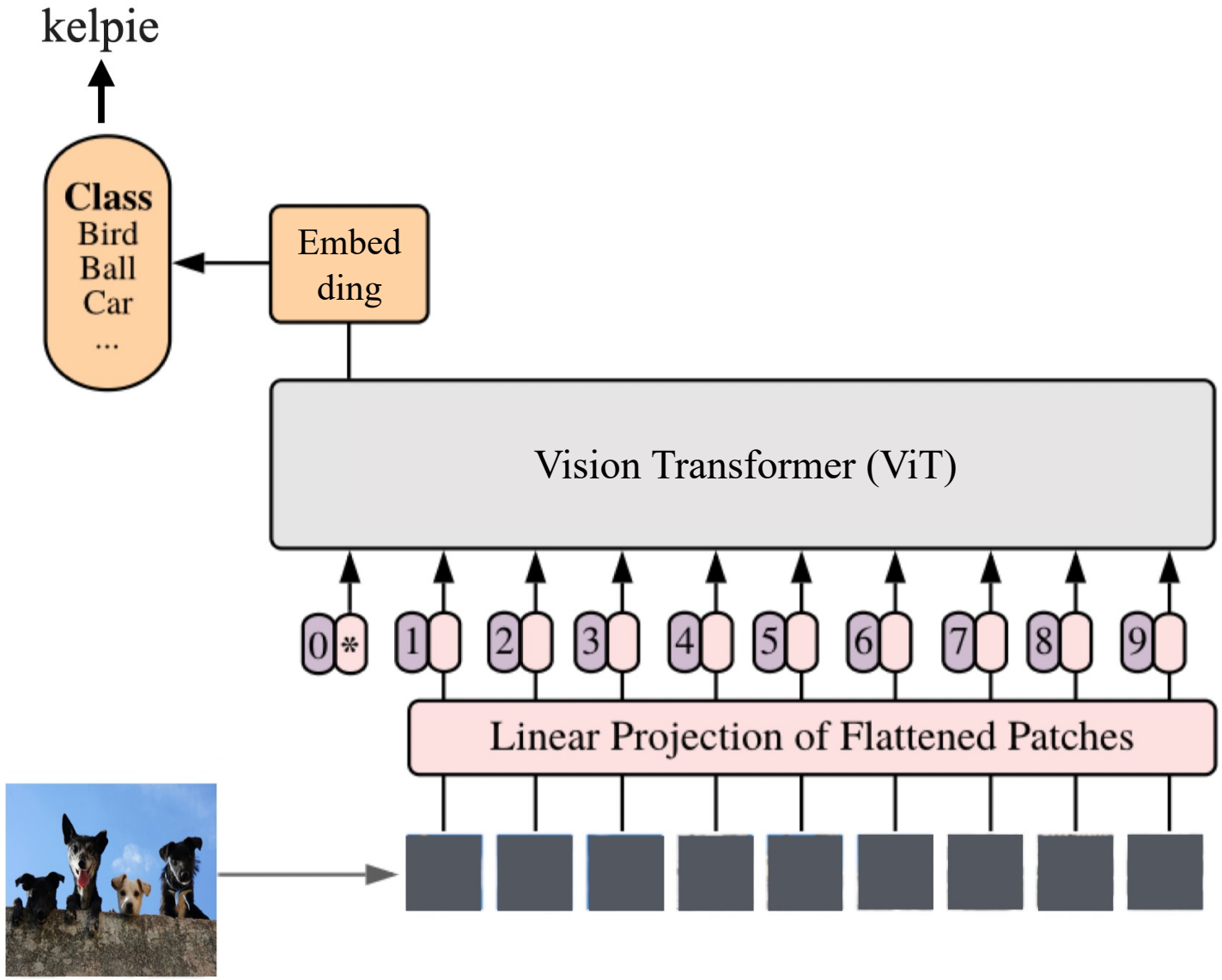
Abstract Perception



Abstract Perception



Abstract Perception



Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR 2021.

Vision Transformer (ViT)

Transformer

- Non-vision specific model
  - Typically applied to 1-D sequence data
- Transformer “encoder”
  - A stack of alternating self-attention and MLP blocks
  - Residuals and LayerNorm
- Transformer “decoder” (not shown)
  - A slightly more involved architecture useful when the output space is different from the input space (e.g. translation)

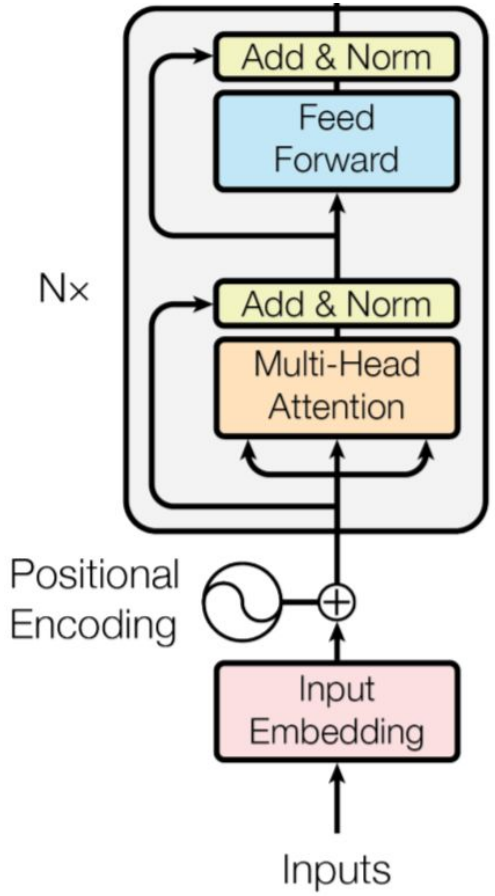
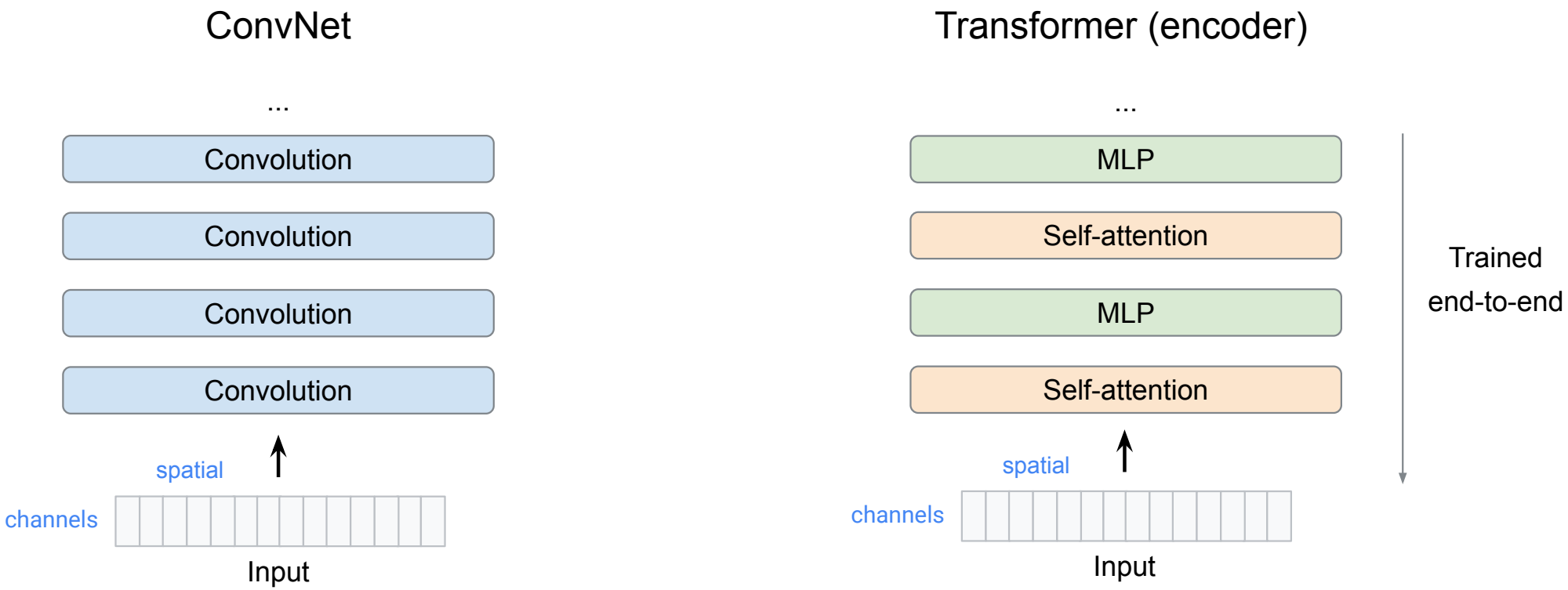


Figure from Vaswani et al.

Vaswani et al., *Attention Is All You Need*, NeurIPS 2017

Vision Transformer (ViT)

ConvNet vs Transformer



Convolutions (with kernels > 1x1) mix both the channels and the spatial locations

MLPs (=1x1 convs) only mix the channels, per location

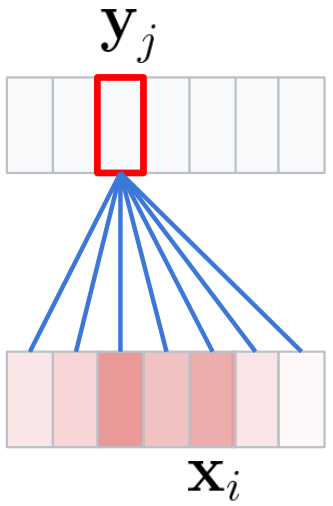
Self-attention mixes the spatial locations (and channels a bit)

*ResNets have grouped of 1x1 convolutions that are nearly identical to transformer's MLPs

Vision Transformer (ViT)

Self-attention

- Each of the tokens (=vectors) attends to all tokens
  - Extra tricks: learned key, query, and value projections, inverse-sqrt scaling in the softmax, and multi-headed attention (omit for simplicity)
- It's a set operation (permutation-invariant)
  - ...and hence need "position embeddings" to "remember" the spatial structure
- It's a global operation
  - Aggregates information from all tokens



$$\boldsymbol{\alpha}_j = softmax(\frac{K\mathbf{x}_1 \cdot Q\mathbf{x}_j}{\sqrt{d_K}}, \ldots, \frac{K\mathbf{x}_n \cdot Q\mathbf{x}_j}{\sqrt{d_K}})$$

$$\mathbf{y}_j = \sum_{i=1}^{n} \alpha_{ji} V\mathbf{x}_i$$

Simplified! Multi-headed attention not shown

Vaswani et al., *Attention Is All You Need*, NeurIPS 2017

Vision Transformer (ViT)

Self-Attention with Queries, Keys, Values

Make three version of each input embedding x(i)

- **Query** vector $\mathbf{q}^{(i)} = \mathbf{W}_q\mathbf{x}^{(i)}$
- **Key** vector: $\mathbf{k}^{(i)} = \mathbf{W}_k\mathbf{x}^{(i)}$
- **Value** vector: $\mathbf{v}^{(i)} = \mathbf{W}_v\mathbf{x}^{(i)}$

The **attention weight of the *j*-th position** to compute the **new output for the *i*-th position** depends on the **query of i** and the **key of j (scaled):**

$$w_j^{(i)} = \frac{\exp(\mathbf{q}^{(i)}\mathbf{k}^{(j)})/\sqrt{k}}{\sum_j(\exp(\mathbf{q}^{(i)}\mathbf{k}^{(j)})/\sqrt{k})}$$

The **new output vector for the i-th position** depends on the **attention weights** and **value** vectors of all **input positions j**:

$$\mathbf{y}^{(i)} = \sum_{j=1..T} w_j^{(i)}\mathbf{v}^{(j)}$$

[Julia Hockenmaier, Lecture 9, UIUC]

Vision Transformer (ViT)

Transformer self-attention layer

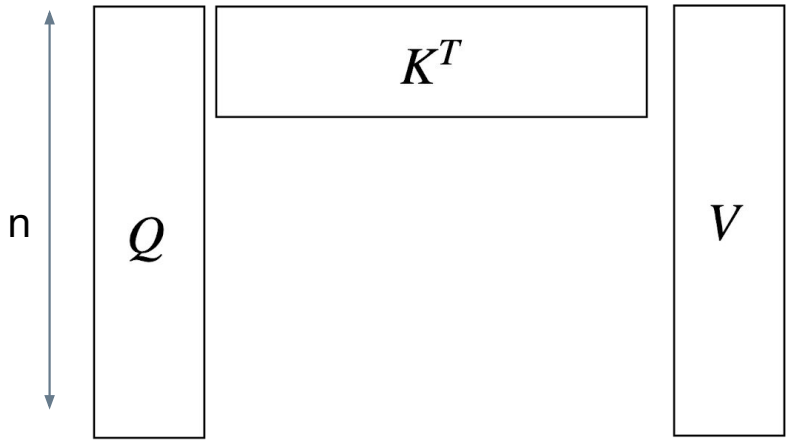
Input: X (matrix of n embedding vectors, each dim m)

Parameters (learned): W_Q, W_K, W_V

Compute:

$$Q = XW_Q$$
$$K = XW_K$$
$$V = XW_V$$

Self-attention Parameters

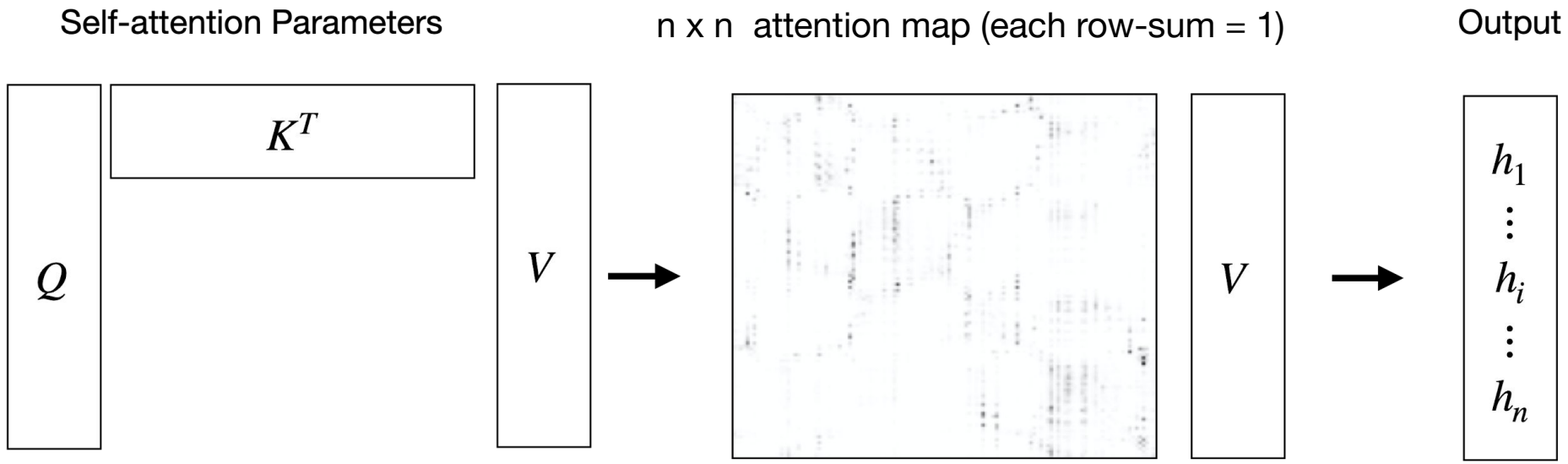


$$Q, K, V \in \mathbb{R}^{n \times m}$$

$$QK^T \in \mathbb{R}^{n \times n}$$

Vision Transformer (ViT)

Transformer self-attention



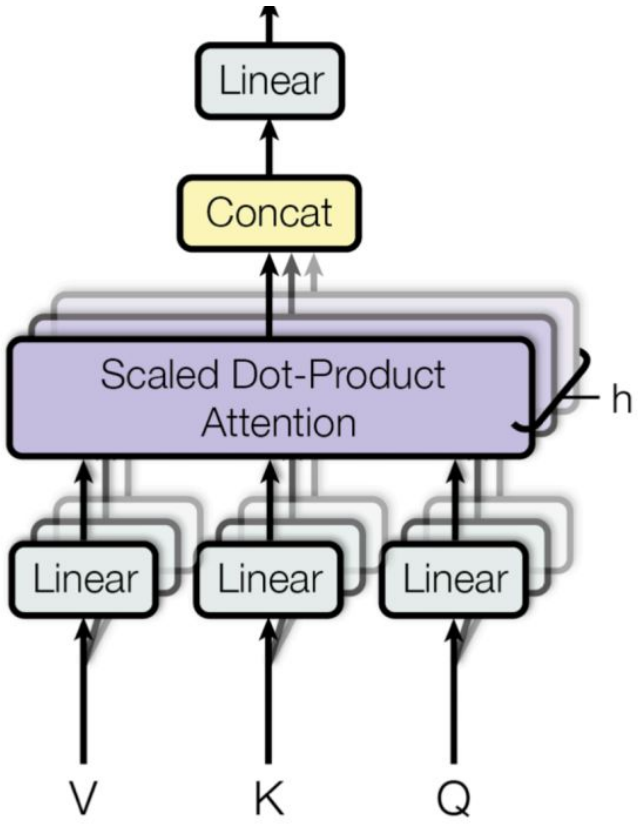
Output matrix H = $\text{softmax}(\frac{1}{\sqrt{d}} QK^T) \cdot V$

Self-attention explicitly models interactions between all pairs of input embeddings

Vision Transformer (ViT)

Multi-Head attention

- Learn *h* different linear projections of Q,K,V
- Compute attention separately on each of these *h* versions
- Concatenate and project the resultant vectors to a lower dimensionality.
- Each attention head can use low dimensionality



$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,...,\text{head}_\text{h})W^O$$

$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

[Julia Hockenmaier, Lecture 9, UIUC]

Vision Transformer (ViT)

Positional Encoding (1-D)

How to capture sequence order?

Add positional embeddings to input embeddings

- Same dimension
- Can be learned or fixed

Fixed encoding: sin / cos of different frequencies:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

Abstract Perception

	Abstract Perception “what” (high-level, conceptual)	**Dense Perception** “where” (low-level, spatial)
Visual		
Semantic		

Abstract Perception – Visual Cue

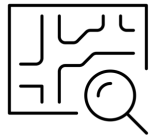
Abstract Perception

“what” (high-level, conceptual)



Dense Perception

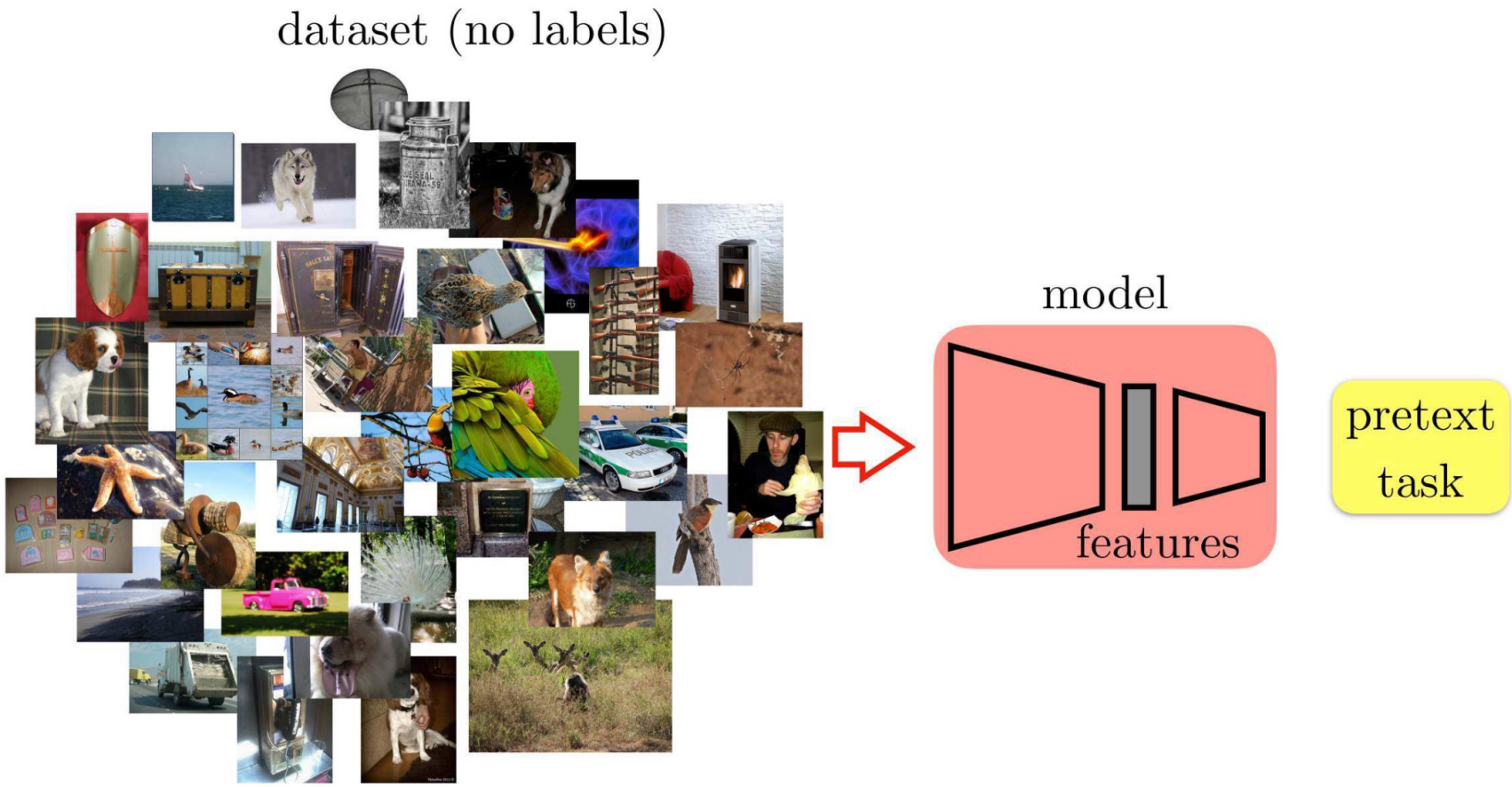
“where” (low-level, spatial)



Visual

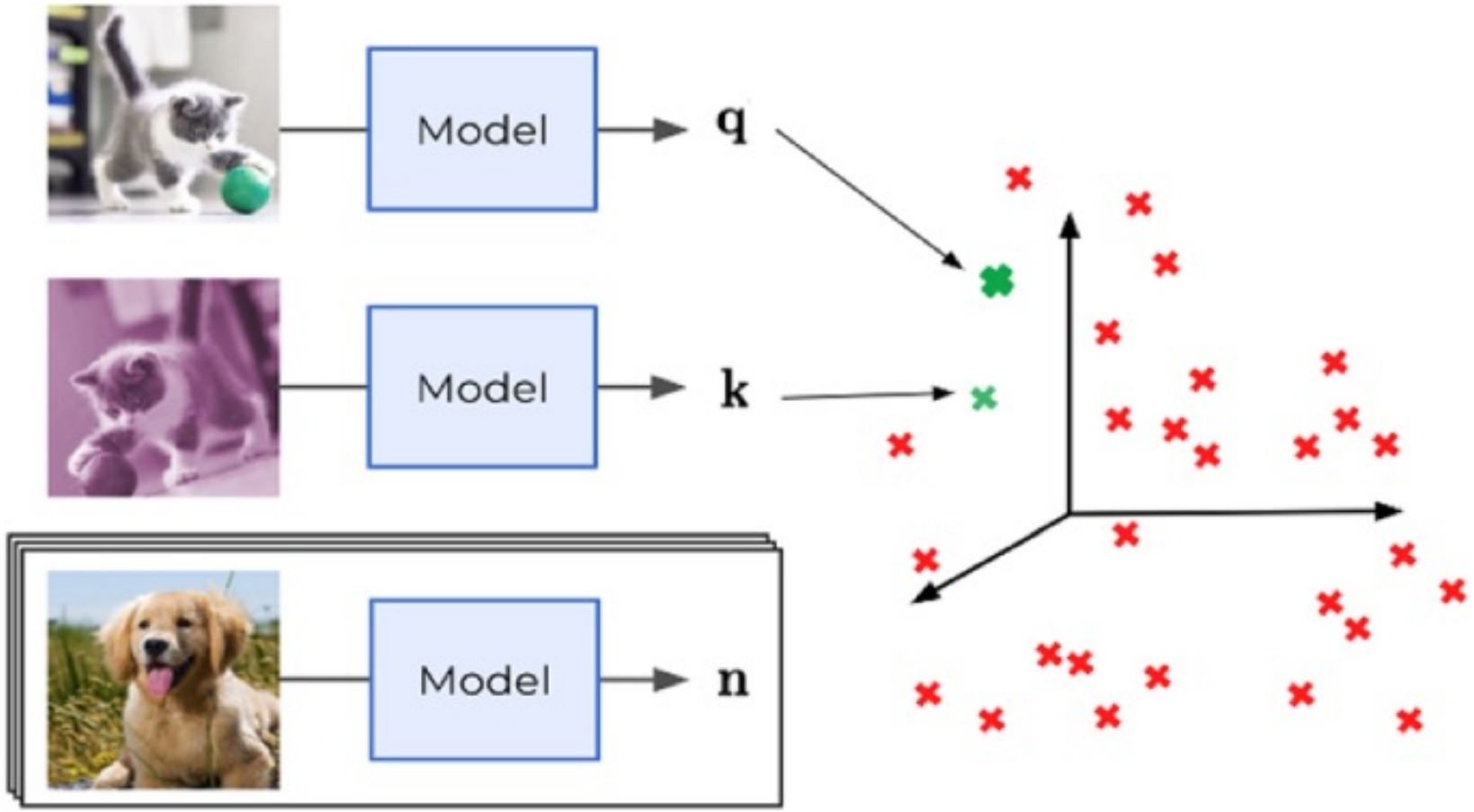
Semantic

Visual Cue: Self-Supervised Learning (e.g., DINO)



Visual Cue: Self-Supervised Learning (e.g., DINO)

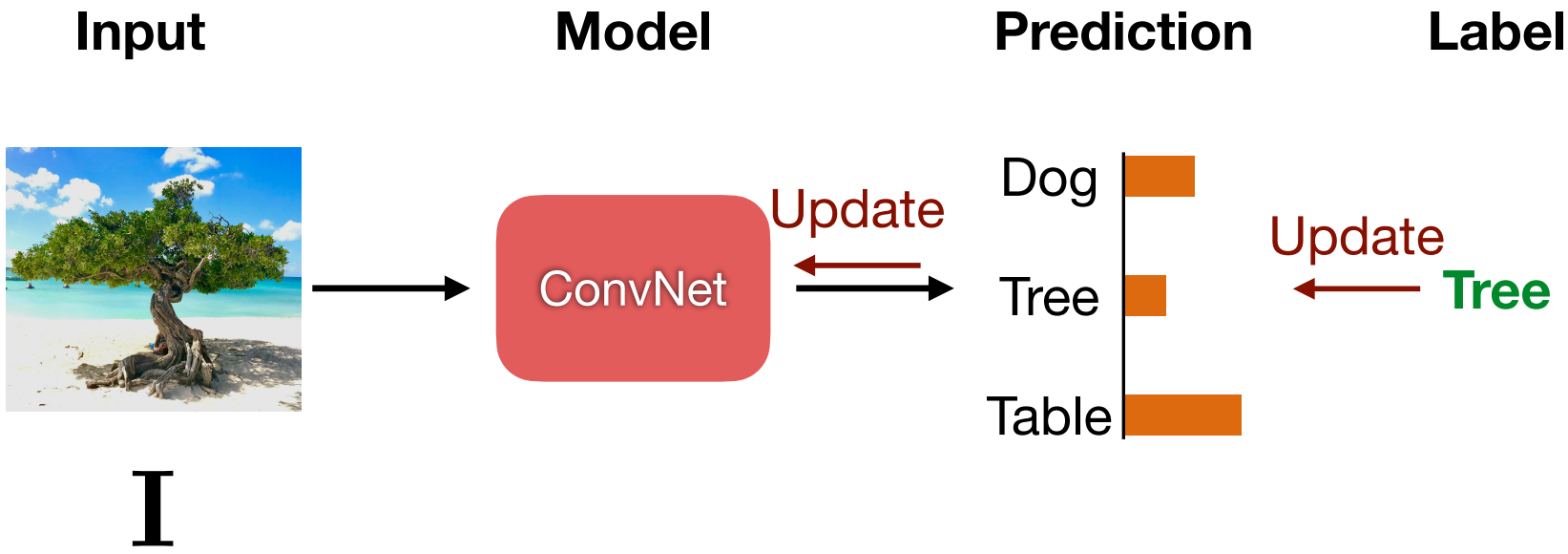
Standard for Image Encoder: Contrastive Learning



Siméoni et al. DINOv3. arXiv 2025.

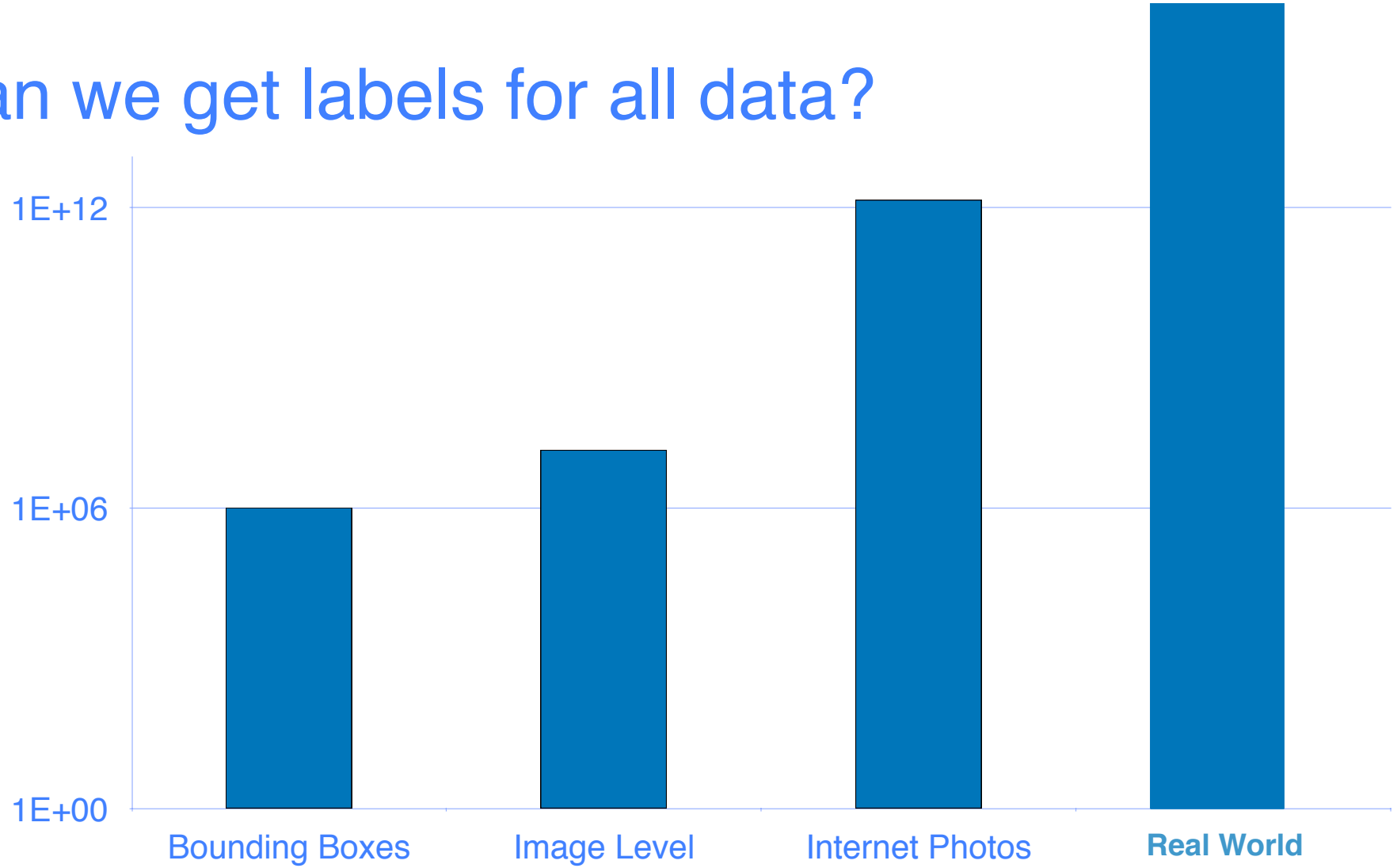
Contrastive Learning

Supervised Learning



Contrastive Learning

Can we get labels for all data?

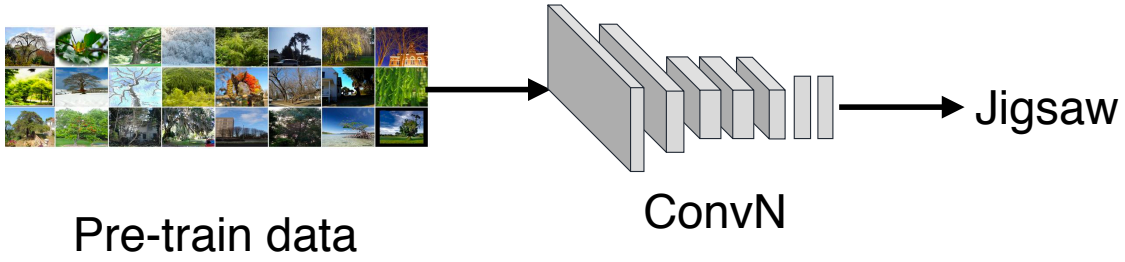


ImageNet (14 million images) needed 22 human years to label

5

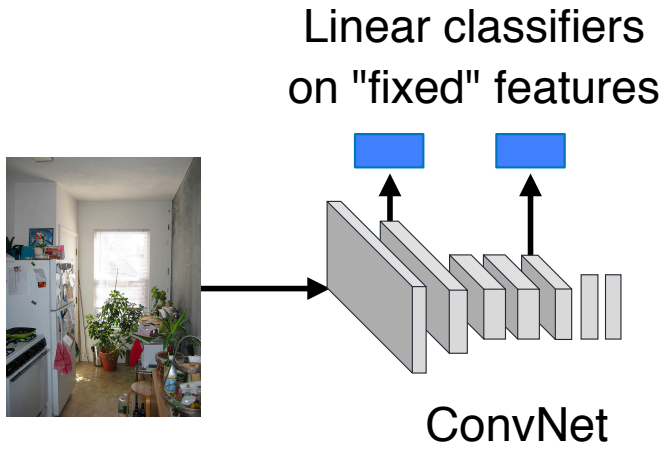
Contrastive Learning

The hope of generalization ... ?



Pre-training

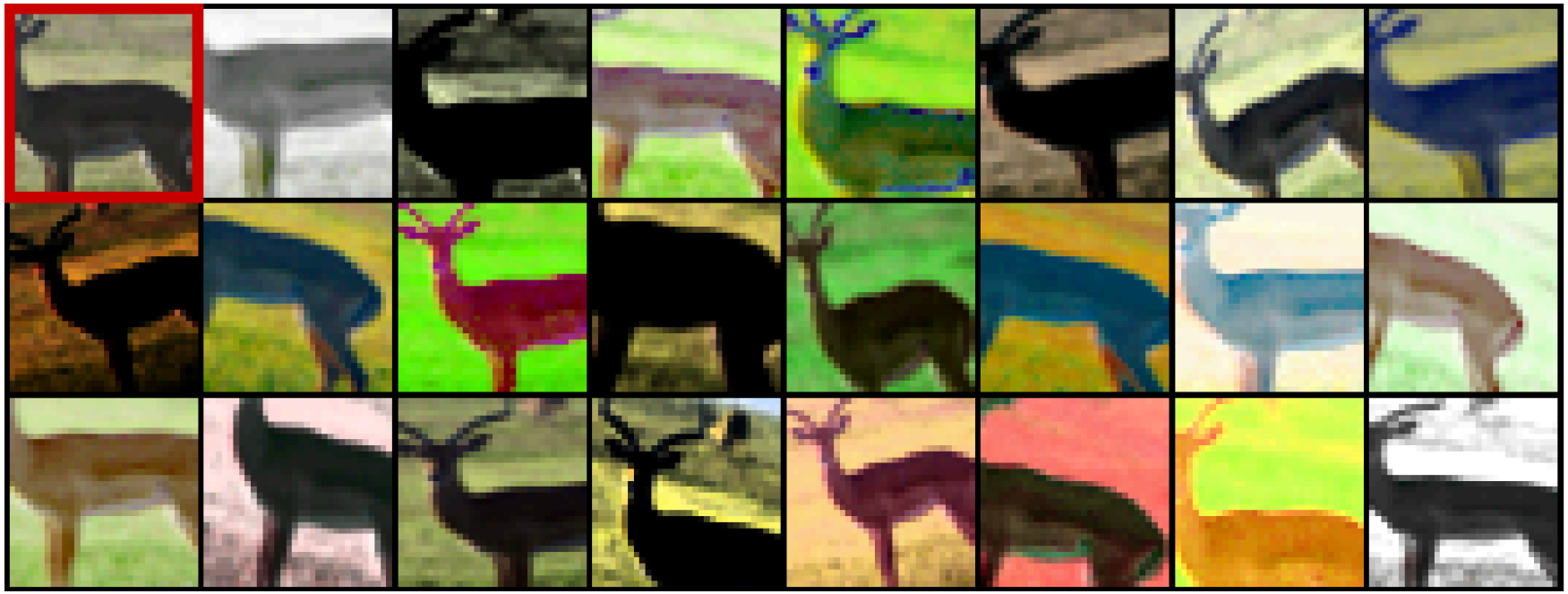
Weak or self-supervised



Transfer

Contrastive Learning

Popular & Common principle for most methods



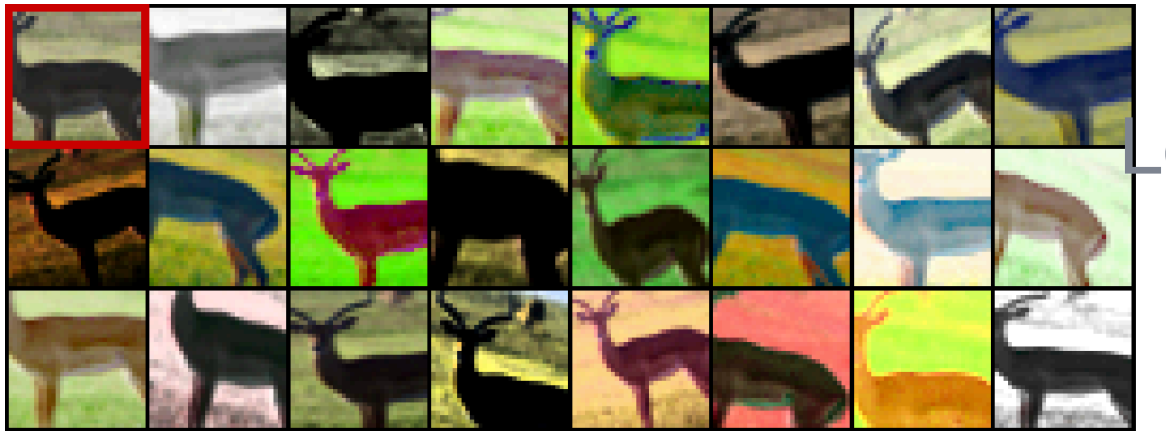
Learn features such that:

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

Figure from Dosovitskiy et al., 2014

Contrastive Learning

Why is it useful?



Learn features such that:

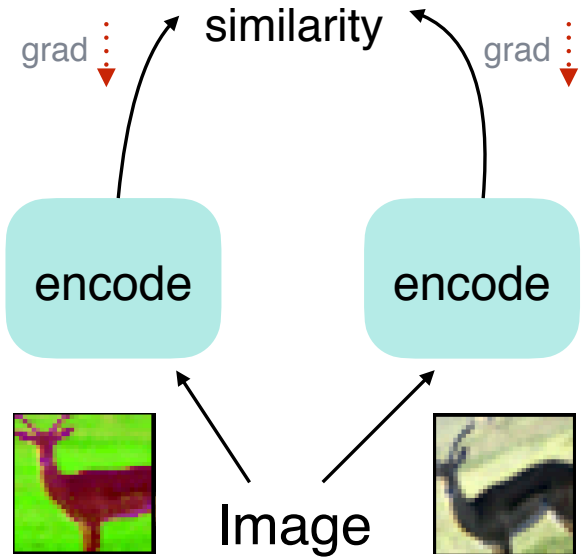
$$f_\theta(I) = f_\theta(\text{augment}(I))$$

Learned features are invariant to "nuisance factors"
or data augmentation

Contrastive Learning

Can it work?

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

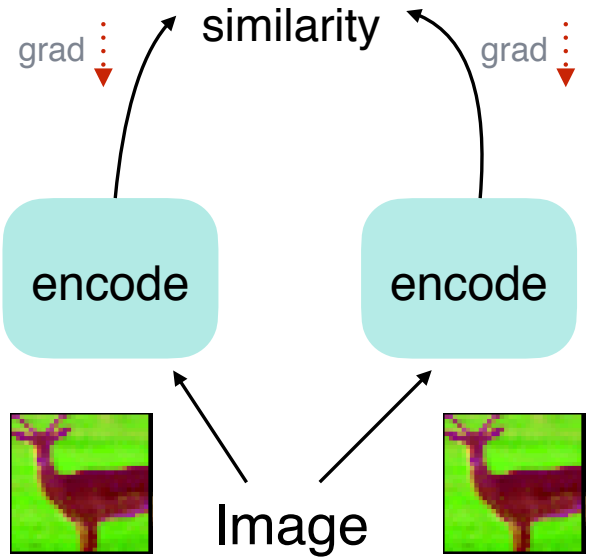


Contrastive Learning

Trivial Solutions

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

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



Satisfies the invariance property, but not useful

Contrastive Learning

Many ways to avoid trivial solutions

Similarity Maximization Objective

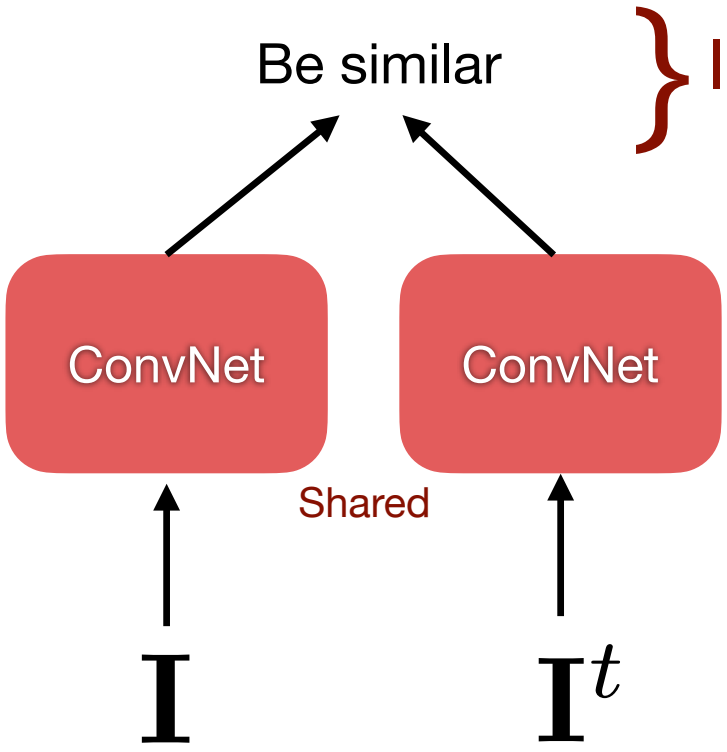
- Contrastive learning
  - MoCo, PIRL, SimCLR
- Clustering
  - DeepCluster, SeLA, SwAV
- Distillation
  - BYOL, SimSiam, DINO

Redundancy Reduction Objective

- Redundancy Reduction
  - Barlow Twins, VICReg

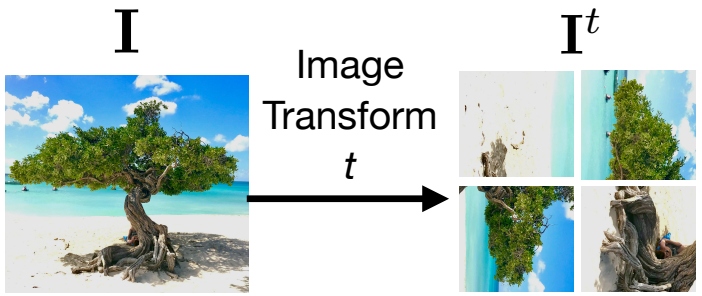
Contrastive Learning

PIRL



} Invariant to Pretext transform

$$L_{\text{contrastive}}(\mathbf{v}_{\mathbf{I}}, \mathbf{v}_{\mathbf{I}^t})$$



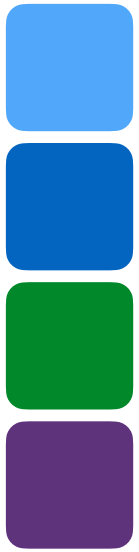
Invariance to

- Data Augmentations
- Multiple views created by pretext task (Jigsaw/Rotation)

Contrastive Learning

Contrastive Learning in PIRL

Dataset



Loss Function

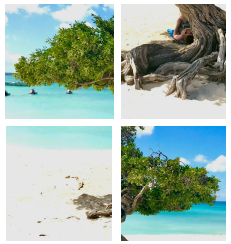


Image Feature & Patch Features

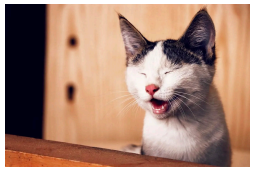
Random Images

$\mathbf{I}$ $\mathbf{I}^t$



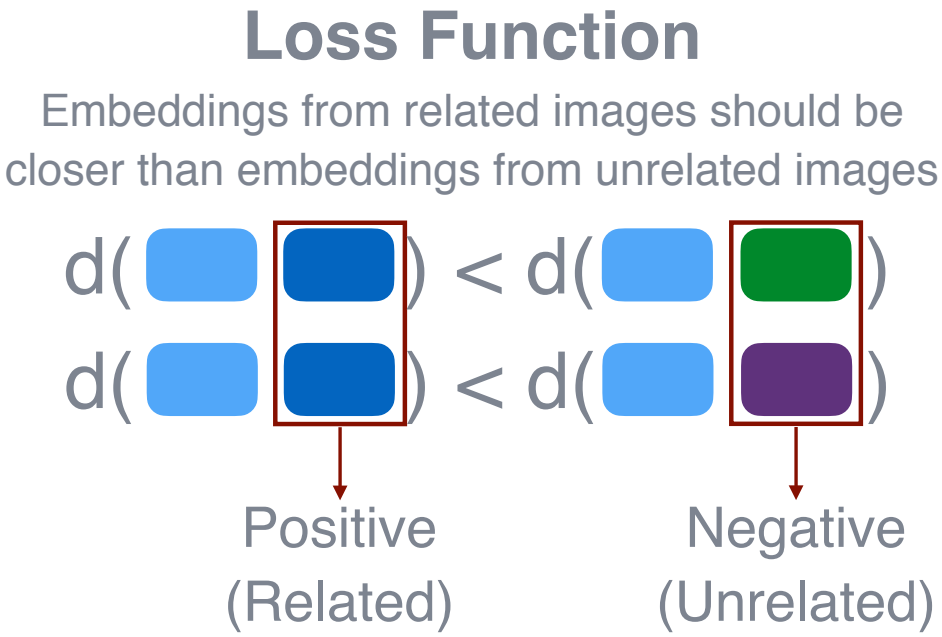






Contrastive Learning

Good negatives are necessary



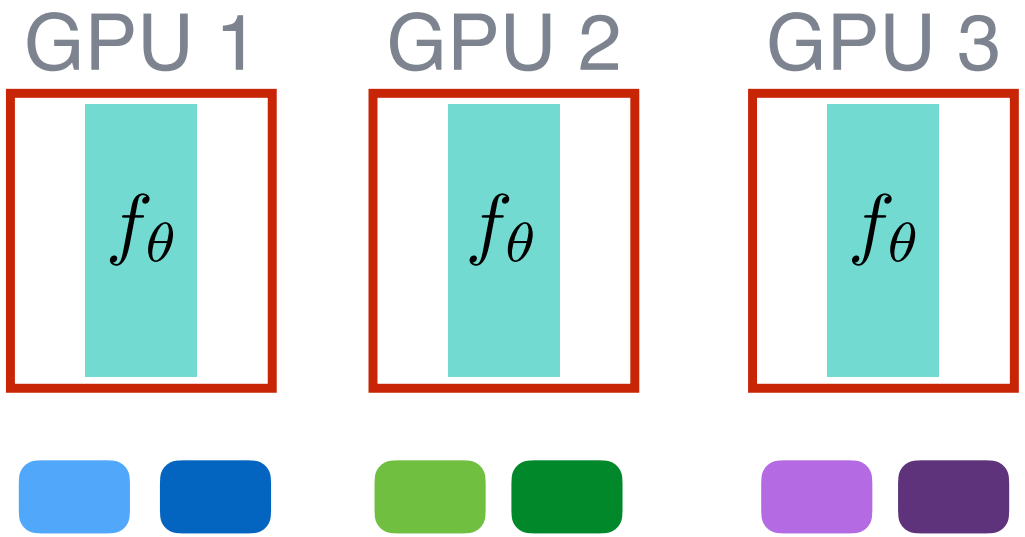
Good negatives are *very* important in contrastive learning

Hadsell et al., 2005, DrLim

Contrastive Learning

SimCLR

- Large batch size - e.g. in SimCLR
- Pros - Simple to implement
- Cons - Large batch size

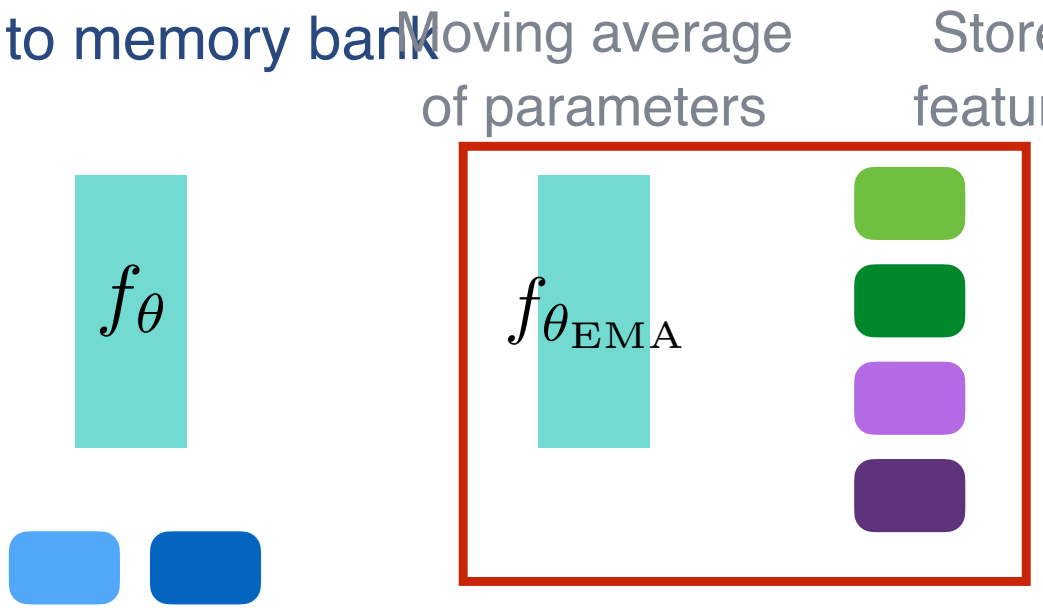


A simple framework for contrastive learning - Chen et al., 20

Contrastive Learning

MoCo

- Maintain "momentum" network - MoCo
- Pros - online
- Cons - extra memory for parameters/stored features, extra fwd pass compared to memory bank



Momentum Contrast - He et al., 2019

Contrastive Learning

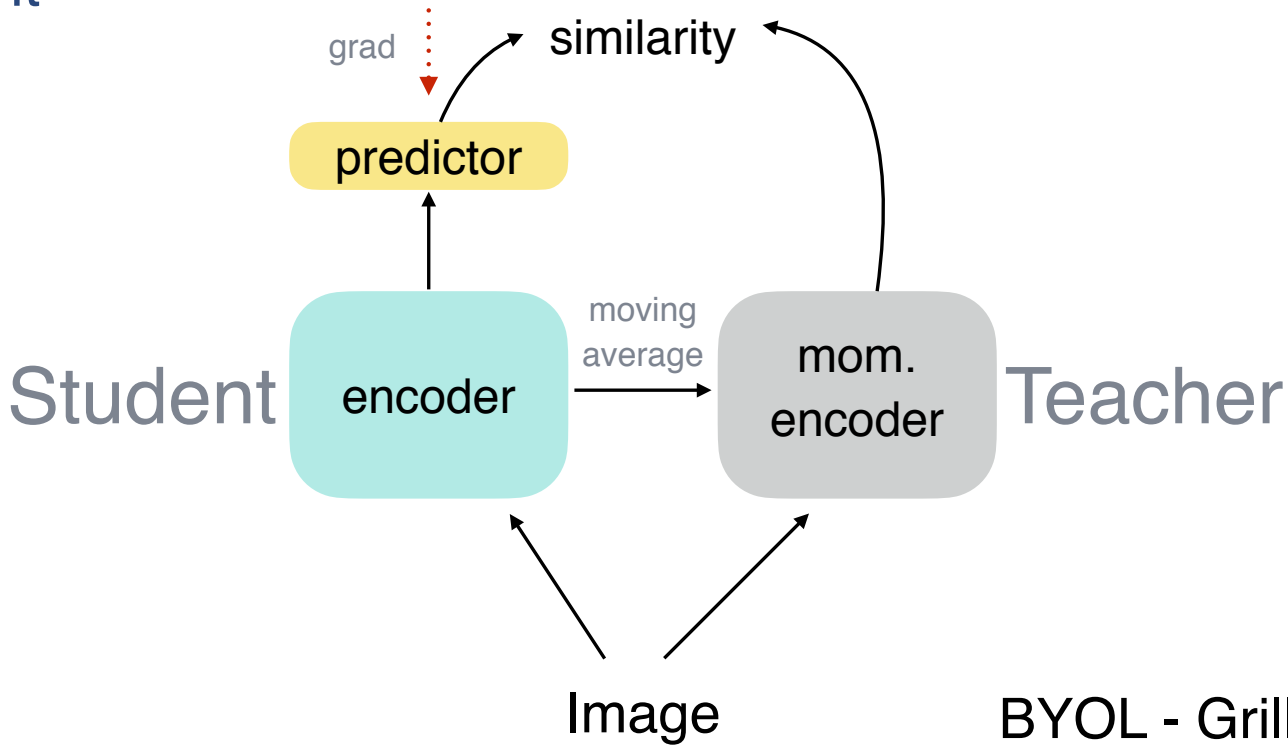
Distillation

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

Contrastive Learning

BYOL

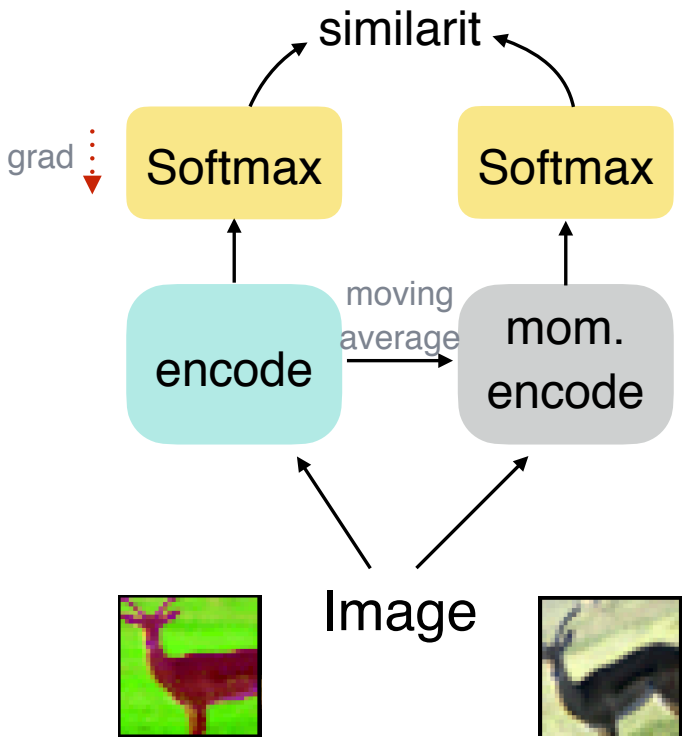
- What we want $f_\theta(I) = f_\theta(\text{augment}(I))$
- How we do it $f_\theta^{\text{student}}(I) = f_\theta^{\text{teacher}}(\text{augment}(I))$



BYOL - Grill et al., 2020

Contrastive Learning

DINO - Main idea



Contrastive Learning

[Submitted on 13 Aug 2025]

DINOv3

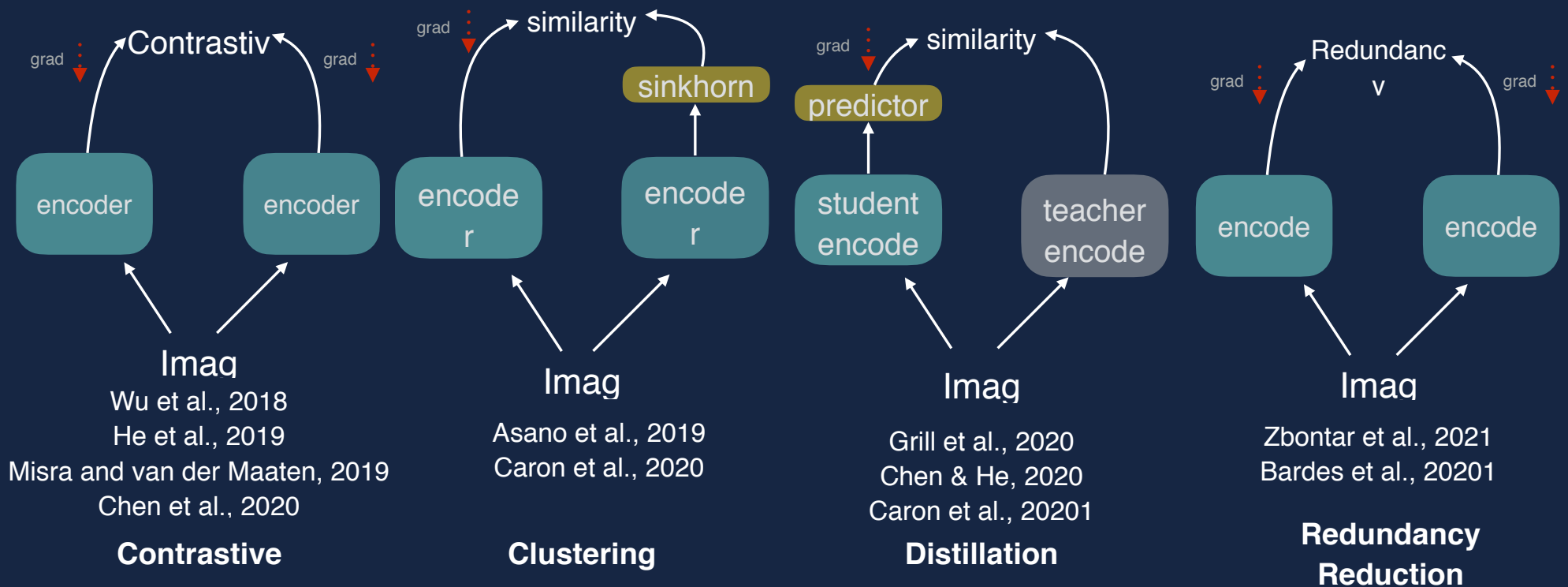
Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, Piotr Bojanowski

Self-supervised learning holds the promise of eliminating the need for manual data annotation, enabling models to scale effortlessly to massive datasets and larger architectures. By not being tailored to specific tasks or domains, this training paradigm has the potential to learn visual representations from diverse sources, ranging from natural to aerial images -- using a single algorithm. This technical report introduces DINOv3, a major milestone toward realizing this vision by leveraging simple yet effective strategies. First, we leverage the benefit of scaling both dataset and model size by careful data preparation, design, and optimization. Second, we introduce a new method called Gram anchoring, which effectively addresses the known yet unsolved issue of dense feature maps degrading during long training schedules. Finally, we apply post-hoc strategies that further enhance our models' flexibility with respect to resolution, model size, and alignment with text. As a result, we present a versatile vision foundation model that outperforms the specialized state of the art across a broad range of settings, without fine-tuning. DINOv3 produces high-quality dense features that achieve outstanding performance on various vision tasks, significantly surpassing previous self- and weakly-supervised foundation models. We also share the DINOv3 suite of vision models, designed to advance the state of the art on a wide spectrum of tasks and data by providing scalable solutions for diverse resource constraints and deployment scenarios.

Subjects: **Computer Vision and Pattern Recognition (cs.CV)**; Machine Learning (cs.LG)
Cite as: arXiv:2508.10104 **[cs.CV]**
(or arXiv:2508.10104v1 **[cs.CV]** for this version)
https://doi.org/10.48550/arXiv.2508.10104

Contrastive Learning

Thanks!



Wu et al., 2018
He et al., 2019
Misra and van der Maaten, 2019
Chen et al., 2020

Contrastive

Asano et al., 2019
Caron et al., 2020

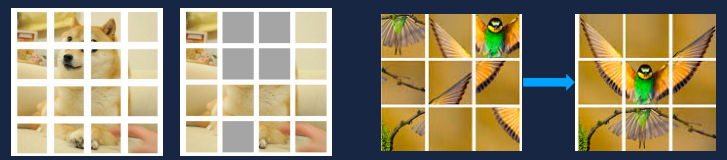
Clustering

Grill et al., 2020
Chen & He, 2020
Caron et al., 20201

Distillation

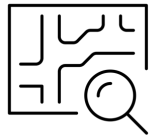
Zbontar et al., 2021
Bardes et al., 20201

Redundancy Reduction



Pretext tasks

Progress So Far – Part I

	Abstract Perception “what” (high-level, conceptual)	**Dense Perception** “where” (low-level, spatial)
		
Visual	Self-Supervised Learning (e.g., DINO)	
Semantic		