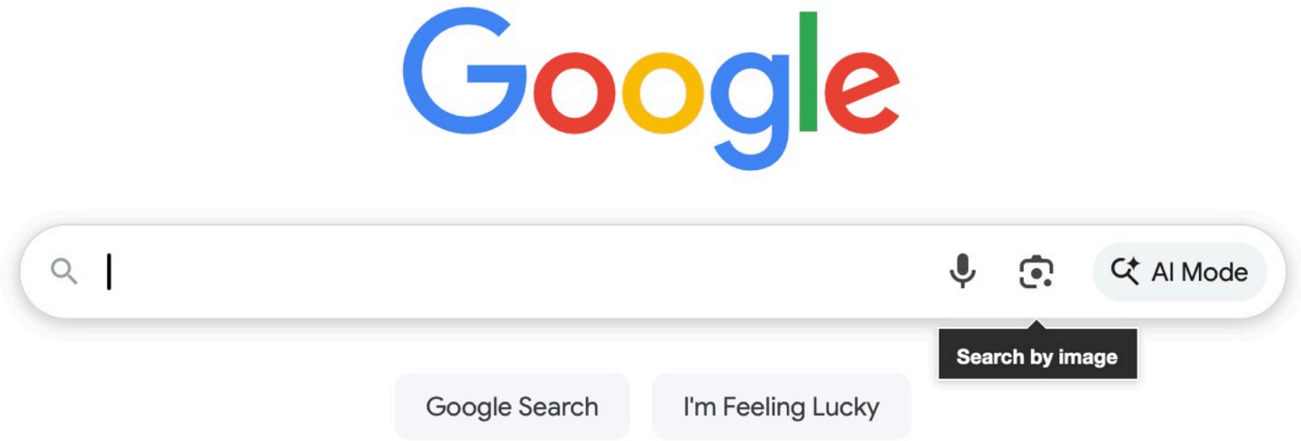
CS-461

Foundation Models and Generative AI

Architectures I:
Vision Foundation Models

Charlotte Bunne, Fall Semester 2025/26

Where do we have images?



Where do we have images?

Where do we have images?

Where do we have images?

Where do we have images?

Where do we have images?

Lin et al. (2023)

Where do we have images?

Labeled vs. Unlabeled Data

e.g.,

Open Images Dataset

15,851,536 boxes on 600 classes

2,785,498 instance segmentations on 350 classes

3,284,280 relationship annotations on 1,466 relationships

675,155 localized narratives

66,391,027 point-level annotations on 5,827 classes

61,404,966 image-level labels on 20,638 classes

ImageNet

1,281,167 training images

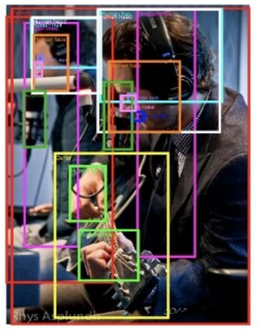
50,000 validation images

100,000 test images

labeled datasets

$\approx 10^6$–10^7 images

image labels
of 20k classes



bounding boxes
of 600 classes

relationships
of 1.4k types

segmentations
of 350 classes

localized narratives
(text caption, audio, and mouse trace)

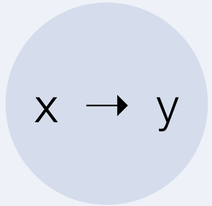
point labels
of 5.8k classes

Labeled vs. Unlabeled Data

unlabeled datasets

billions to trillions?

unsupervised or **self-supervised training**



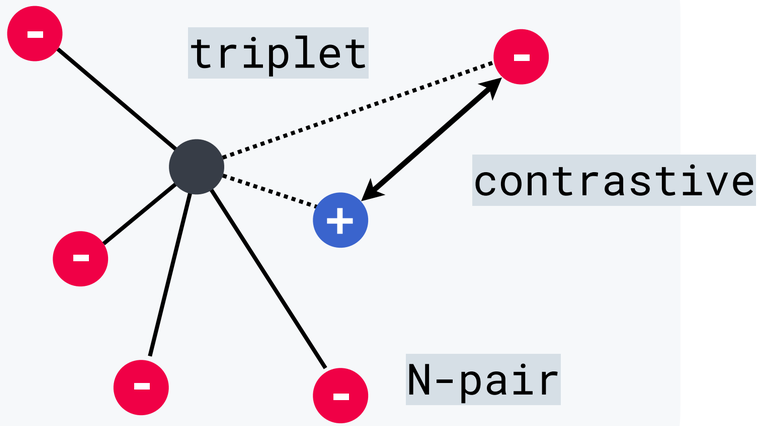
labeled datasets

$\approx 10^6$–10^7 images

supervised training

Self-Supervised Learning Concepts

Contrastive



Frames `N-pair` loss as mutual information maximization between positive pairs. `infoNCE`

Masking

$$\mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{M}\left[\frac{1}{|M|}\sum_{i\in M}\ell\left(x_i,\hat{x}_i\right)\right]$$

cross-entropy MSE

Masking methods approximate pseudo-likelihood optimization.

Autoregressive

$$-\mathbb{E}_{x\sim\mathcal{D}}\left[\sum_{t=1}^{T}\log p_\theta\left(x_t \mid x_{<t}\right)\right]$$

Autoregressive models optimize the exact likelihood.

Core Architectural Principles

Convolutional Neural Networks

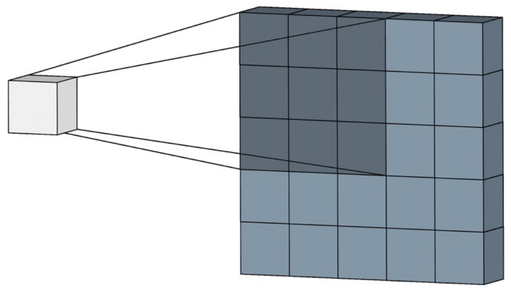
Vision Transformers

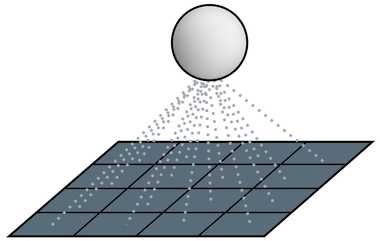
Whatever Comes Next ?

Hierarchical Vision Transformers

Operating on raw pixels.

Local receptive fields and weight sharing!
→ hierarchical, **translation-equivariant** features.

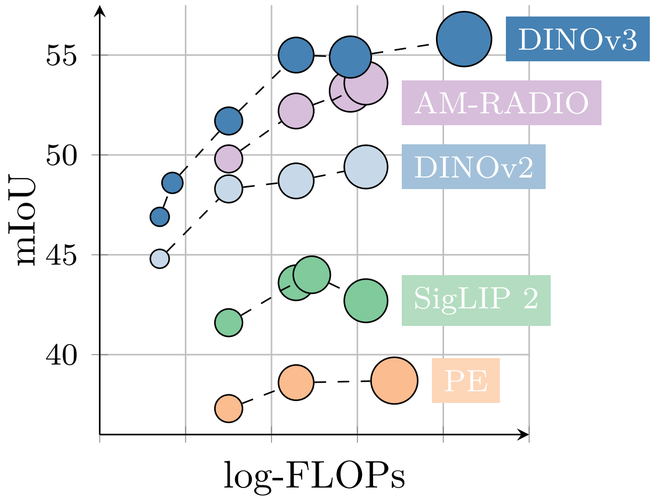




Tokenize images!

Global self-attention learns features with minimal inductive bias and scales with data.

So far, scaling laws apply …



1989 → 2019

2020 →

time

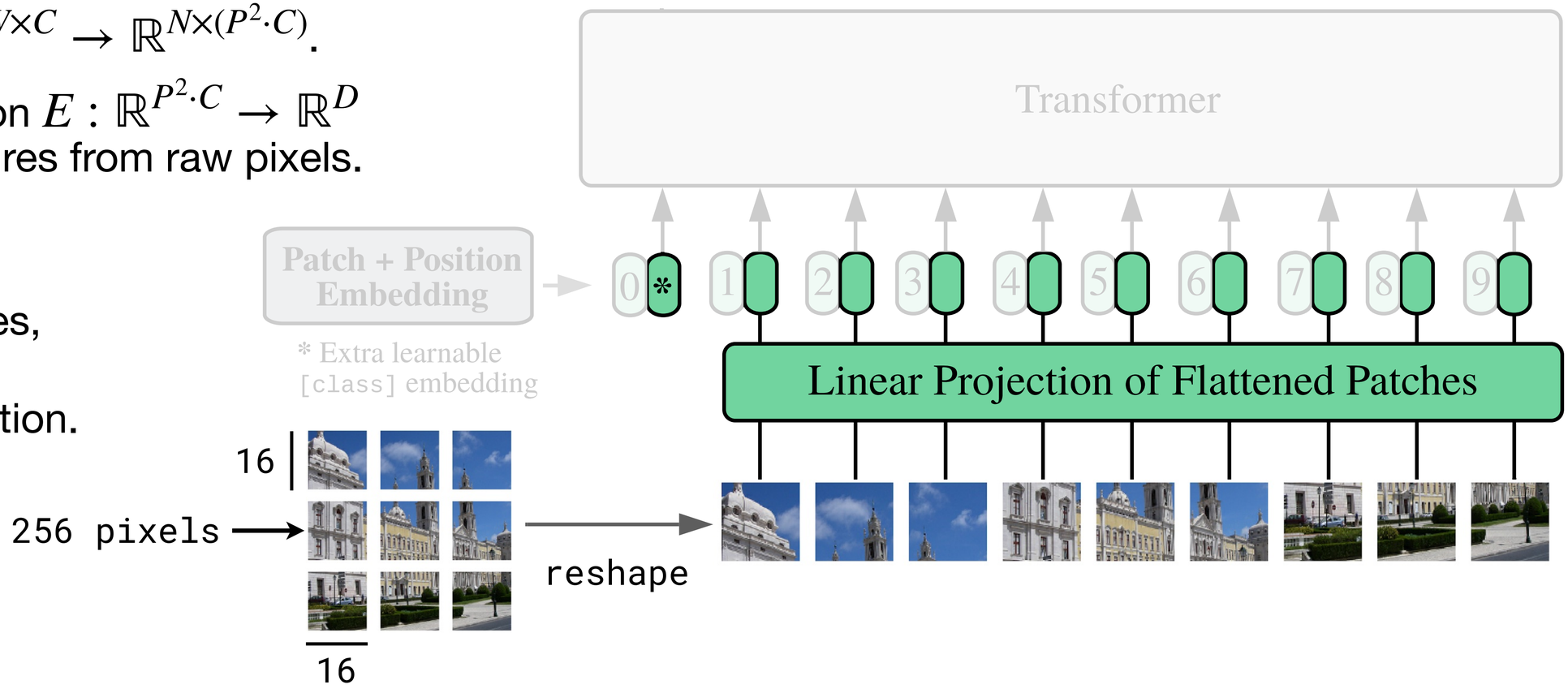
Vision Transformers: Fixed Patch Tokenization

Tokenize images to make them Transformer-compatible.

> Given image $I \in \mathbb{R}^{H \times W \times C}$.
>
> Reshape $\mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{N \times (P^2 \cdot C)}$.
>
> Linear projection $E : \mathbb{R}^{P^2 \cdot C} \rightarrow \mathbb{R}^{D}$ to extract features from raw pixels.

$N = \frac{HW}{P^2}$ patches,

with $\mathcal{O}(N^2)$ attention.



Vision Transformers: VQ-VAE Tokenization

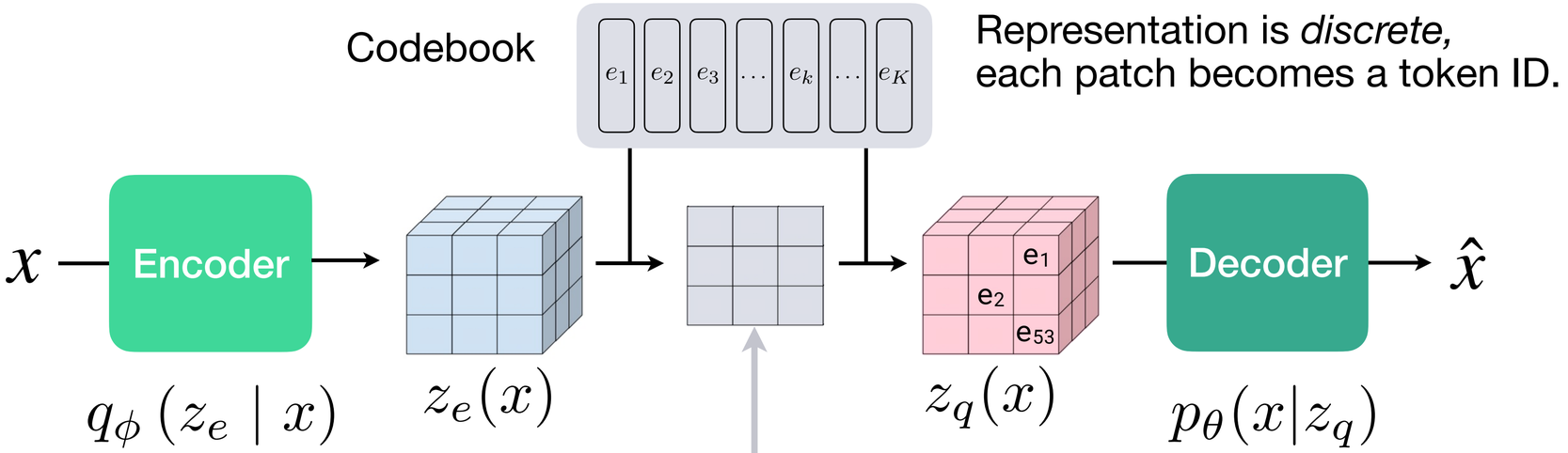
Vector-Quantized Variational Autoencoder (VQ-VAE) represents a fundamentally different approach:

Goal: Instead of fixed patches, compress a continuous image x into a sequence of discrete latent tokens, i.e., a **learned vocabulary of visual concepts**.

Encoder:
Maps x into continuous latent features z_e.

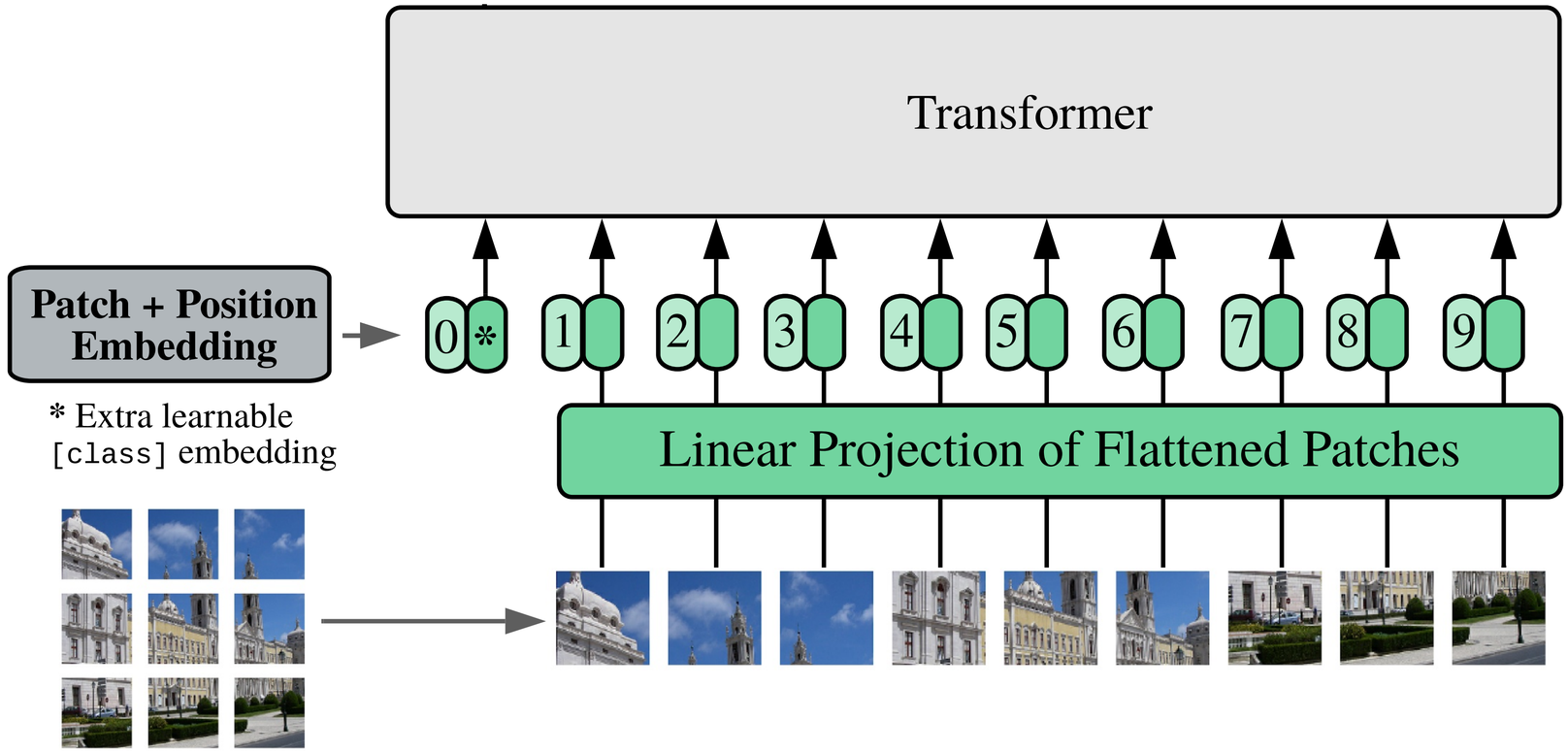
Vector Quantization:
z_e is replaced by its nearest codebook entry e_k.

Decoder:
Reconstucts x from quantized representation z_q.



(van den Oord et al., 2017)

Vision Transformers



Vision Transformers: Positional Encodings

`learned` **Absolute,** i.e., 1D index.

A simple sequence index that self-attention lifts to 2D. Often used in contrastive FMs where we have a stable input. Requires interpolation when input size changes.

`learned` **Relative Position Bias**.

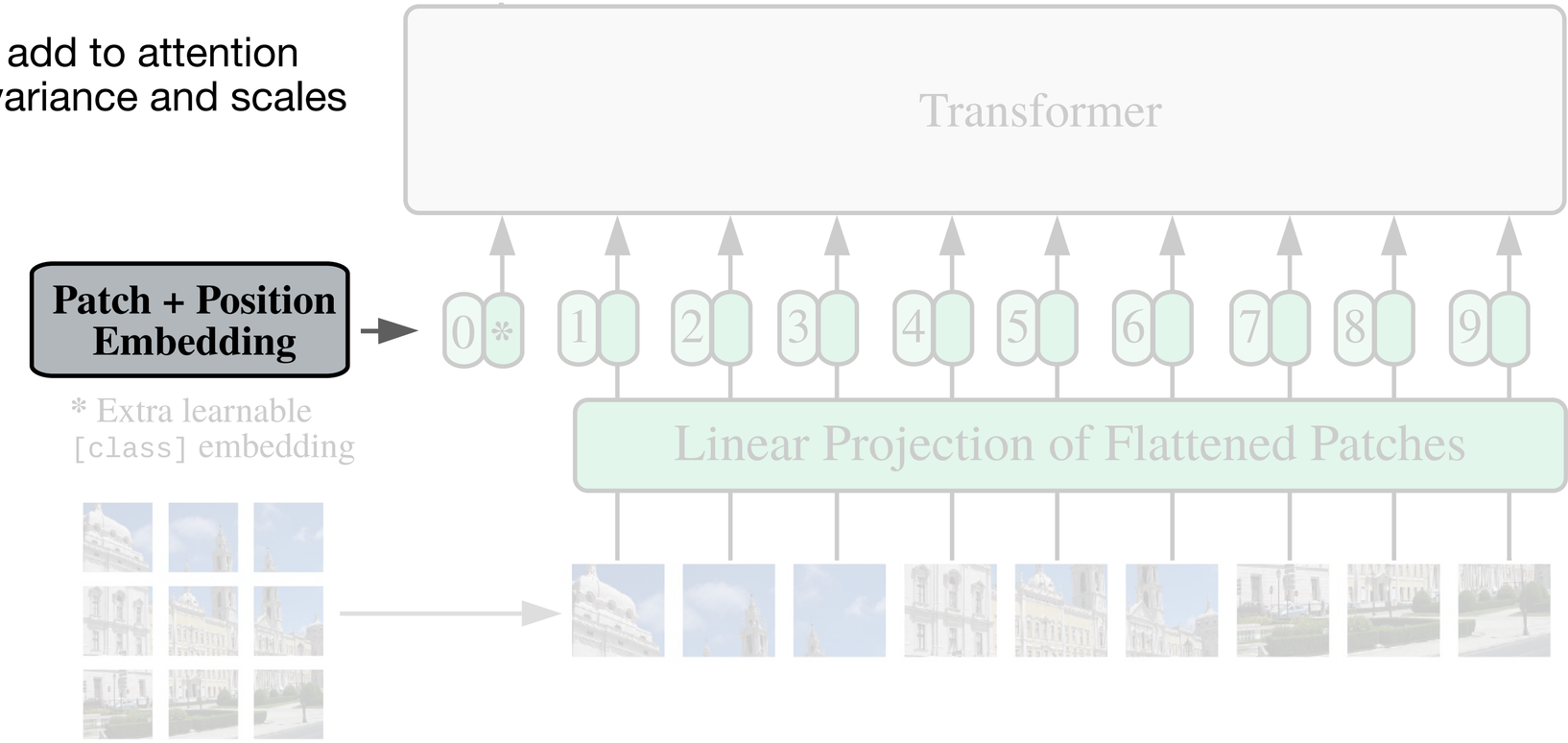
Learn parameters over (Δx, Δy) and add to attention logits; encourages translation-equivariance and scales better across image sizes.

Fixed `sinusoidal`.

Used in masked or generative objectives where many patches are missing or sequence length varies.

Relative `rotary`.

Naturally relative which improves shift robustness and length extrapolation; used with autoregressive models.



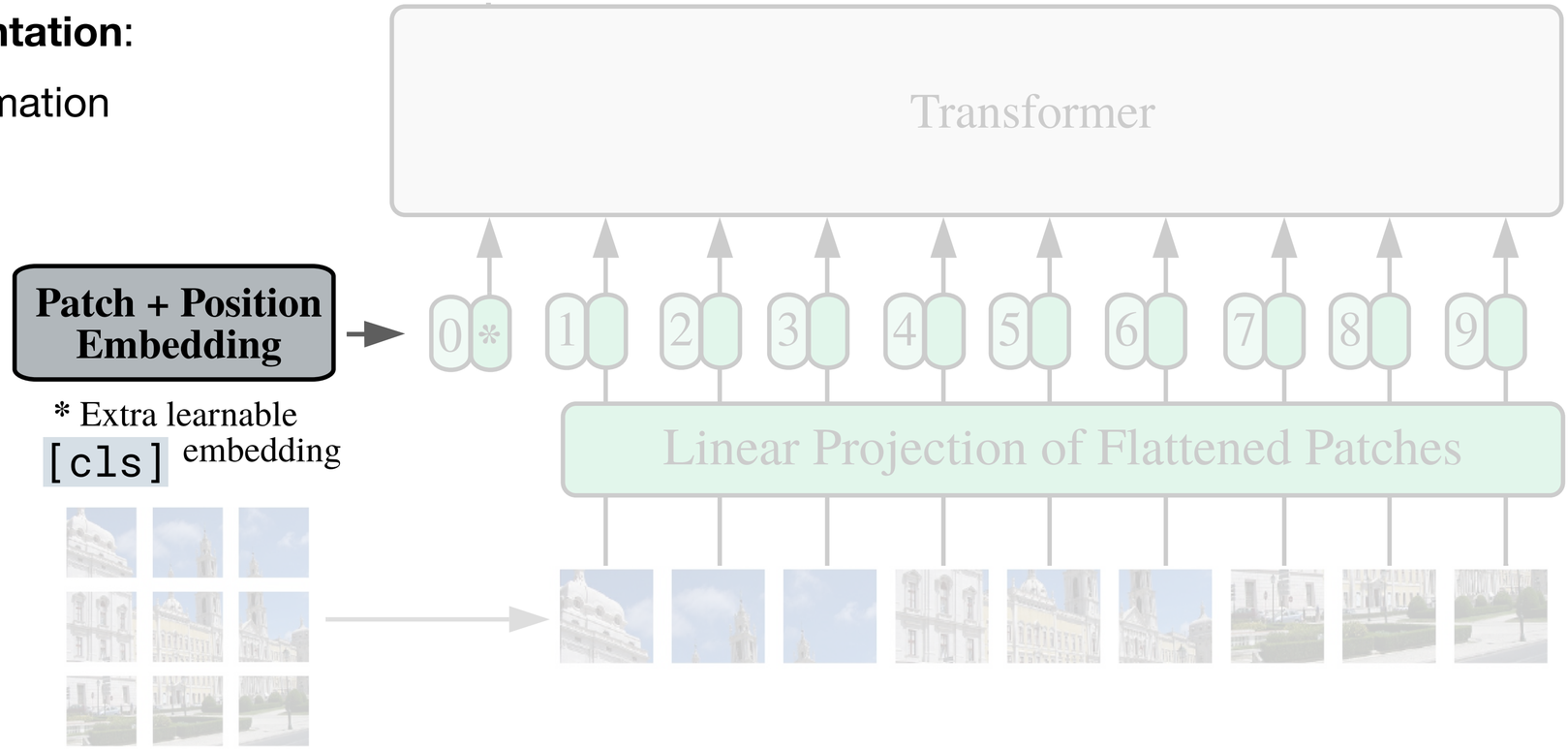
Vision Transformers: A Summary Token

Learnable `[cls]` token!

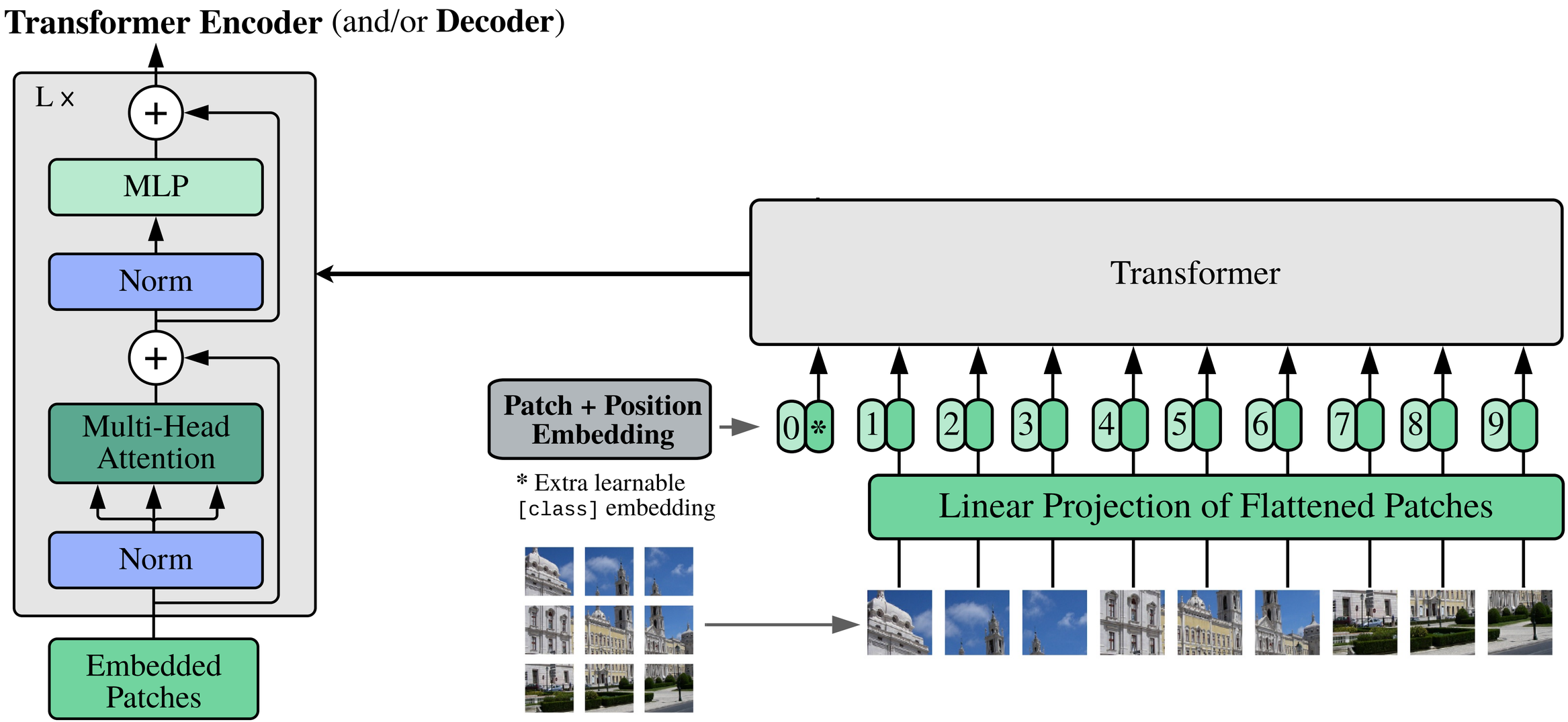
Special learnable token prepended at position 0:
Does not correspond to any image patch; randomly initialized.

Aggregates global image representation:

- Through self-attention, pulls information from all patches for classification.
- Patch tokens contain spatial info while `[CLS]` learns task-specific features.



Vision Transformers

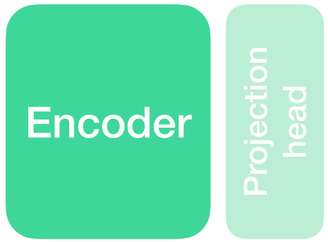


From Learning Principle to Architecture

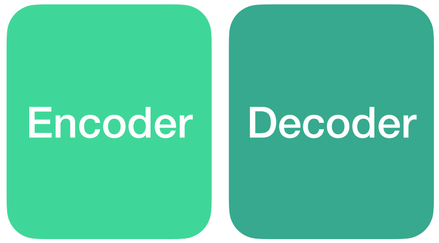
Important!

The **self-supervision objective** is *not* a training detail but the fundamental constraint that **shapes many architectural choices** from attention patterns to encoder-decoder asymmetry.

1. **Contrastive**



2. **Masked**



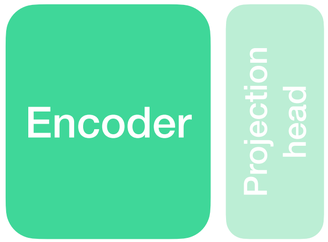
3. **Autoregressive**

From Learning Principle to Architecture

Important!

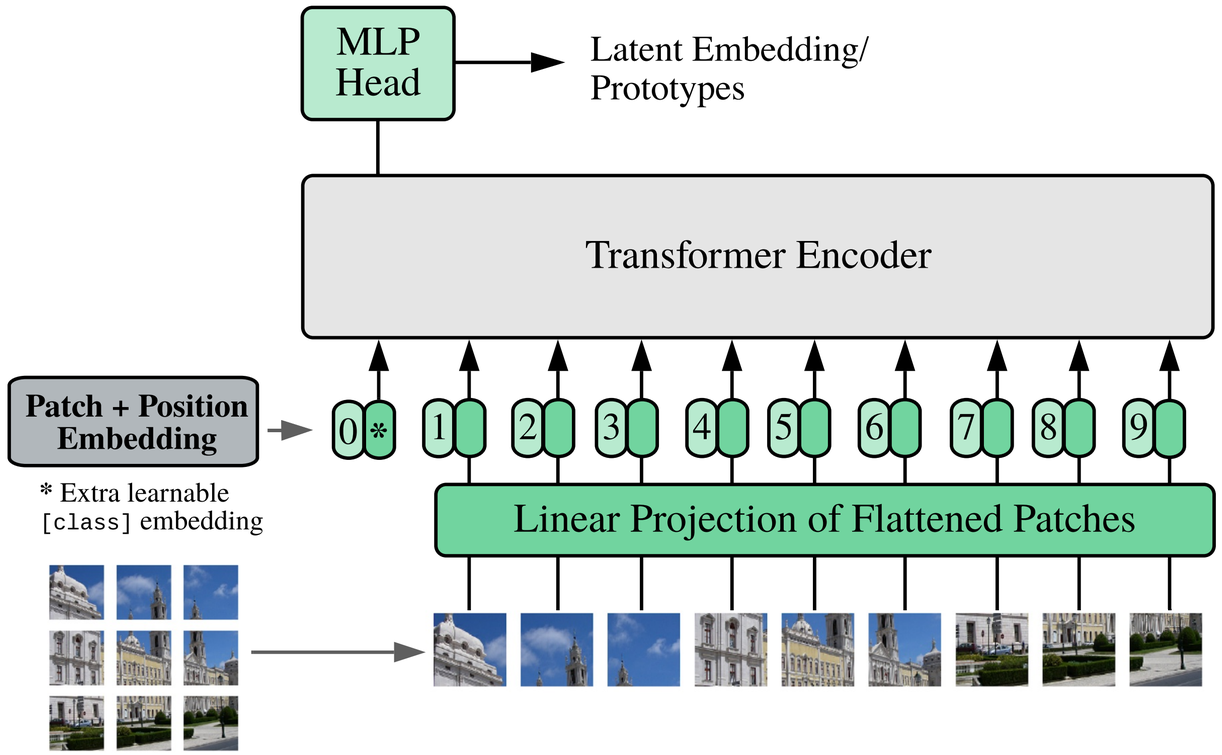
The **self-supervision objective** is *not* a training detail but the fundamental constraint that **shapes many architectural choices** from attention patterns to encoder-decoder asymmetry.

1. **Contrastive**



2. **Masked**

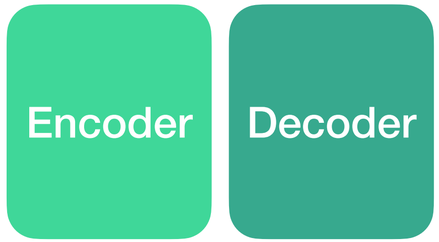
3. **Autoregressive**



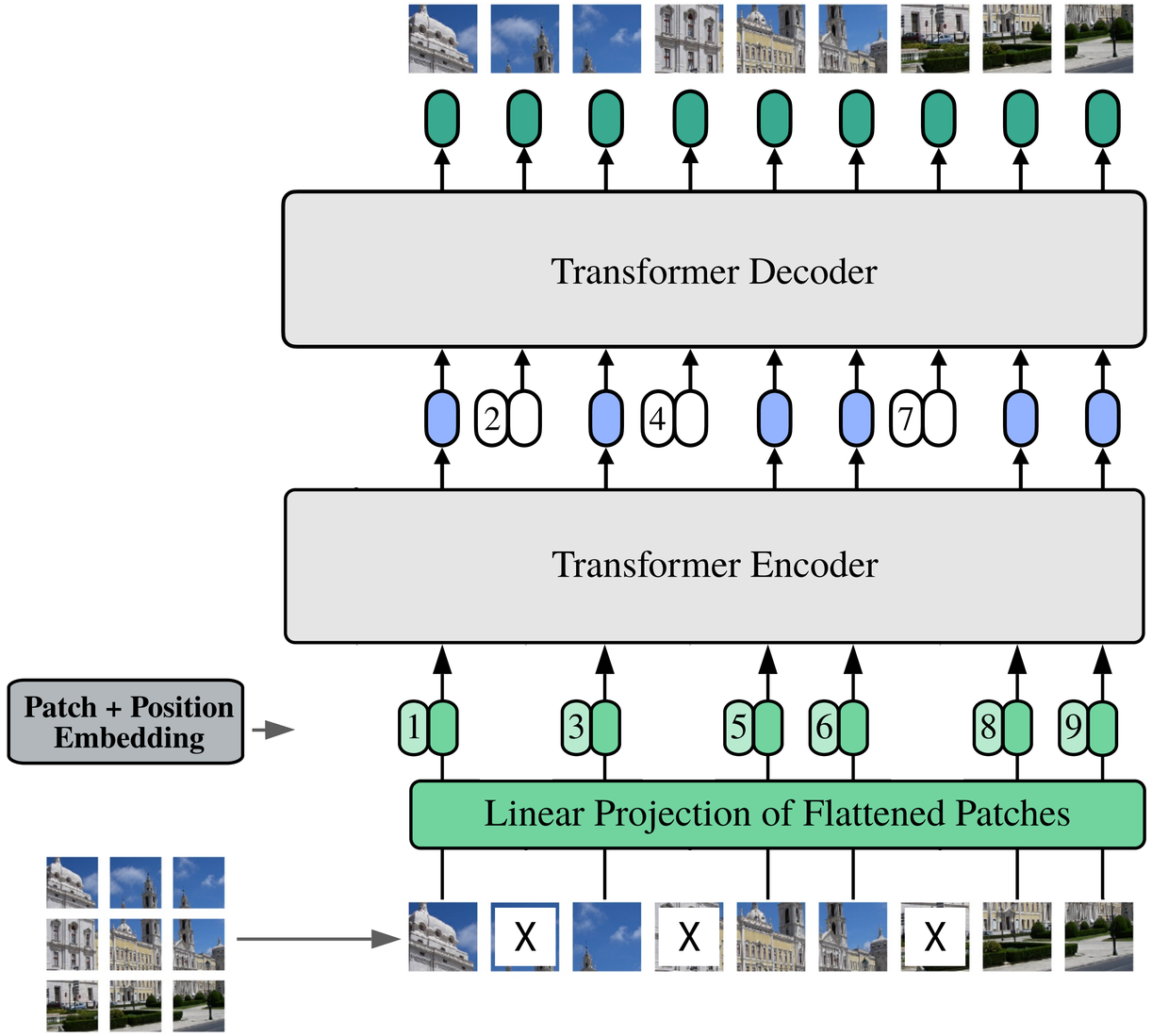
From Learning Principle to Architecture

1. Contrastive

2. **Masked**



3. Autoregressive

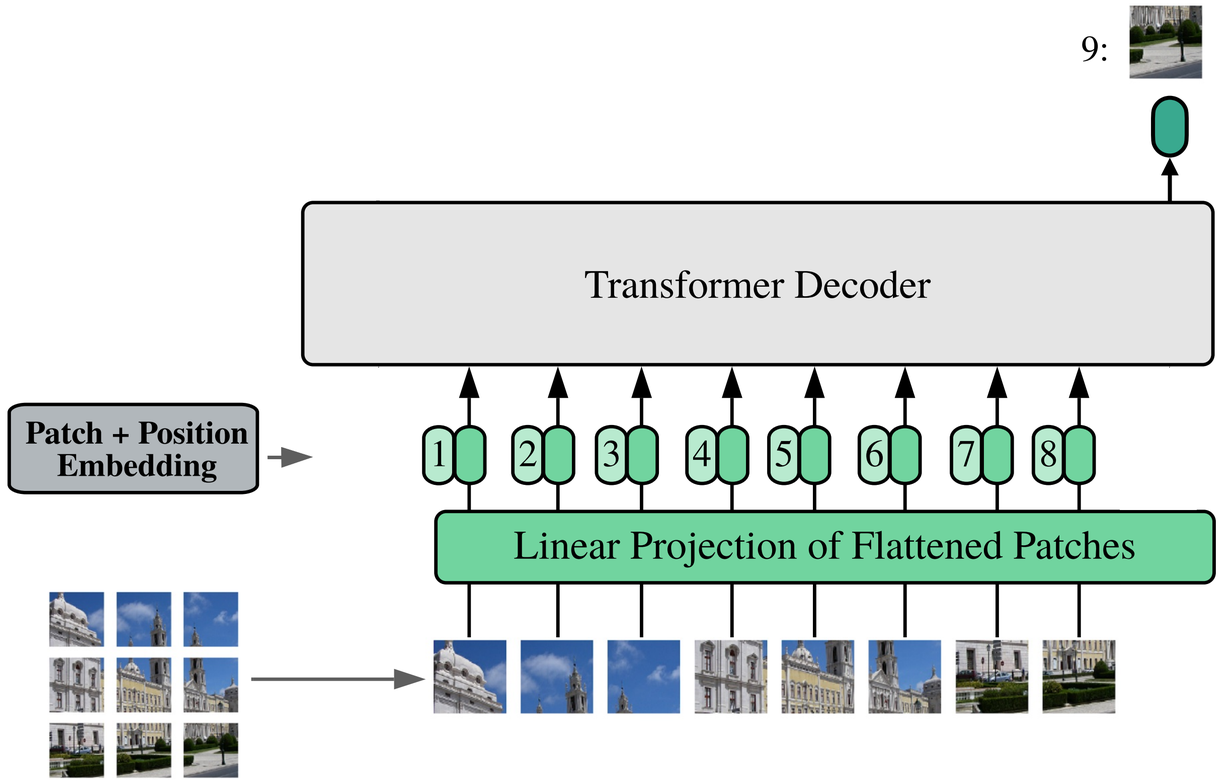


From Learning Principle to Architecture

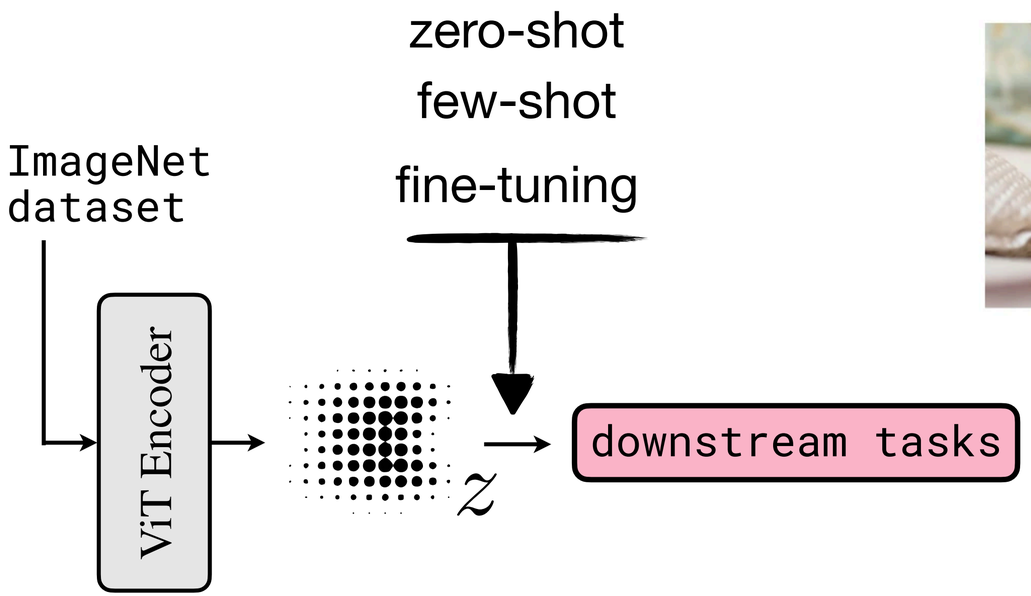
1. Contrastive

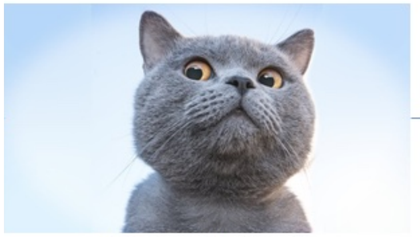
2. Masked

3. **Autoregressive**



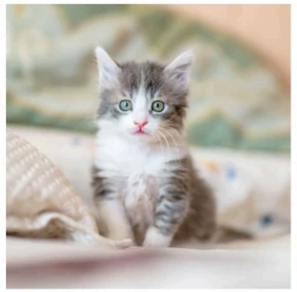
Why are they Foundation Models?





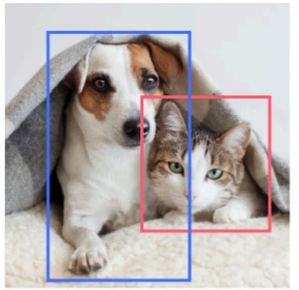
"Query" Image

Classification



Cat

Object Detection

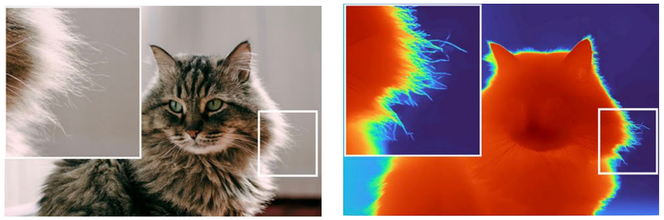


Cat, Dog

Instance Segmentation

Cat, Dog

Depth Estimation

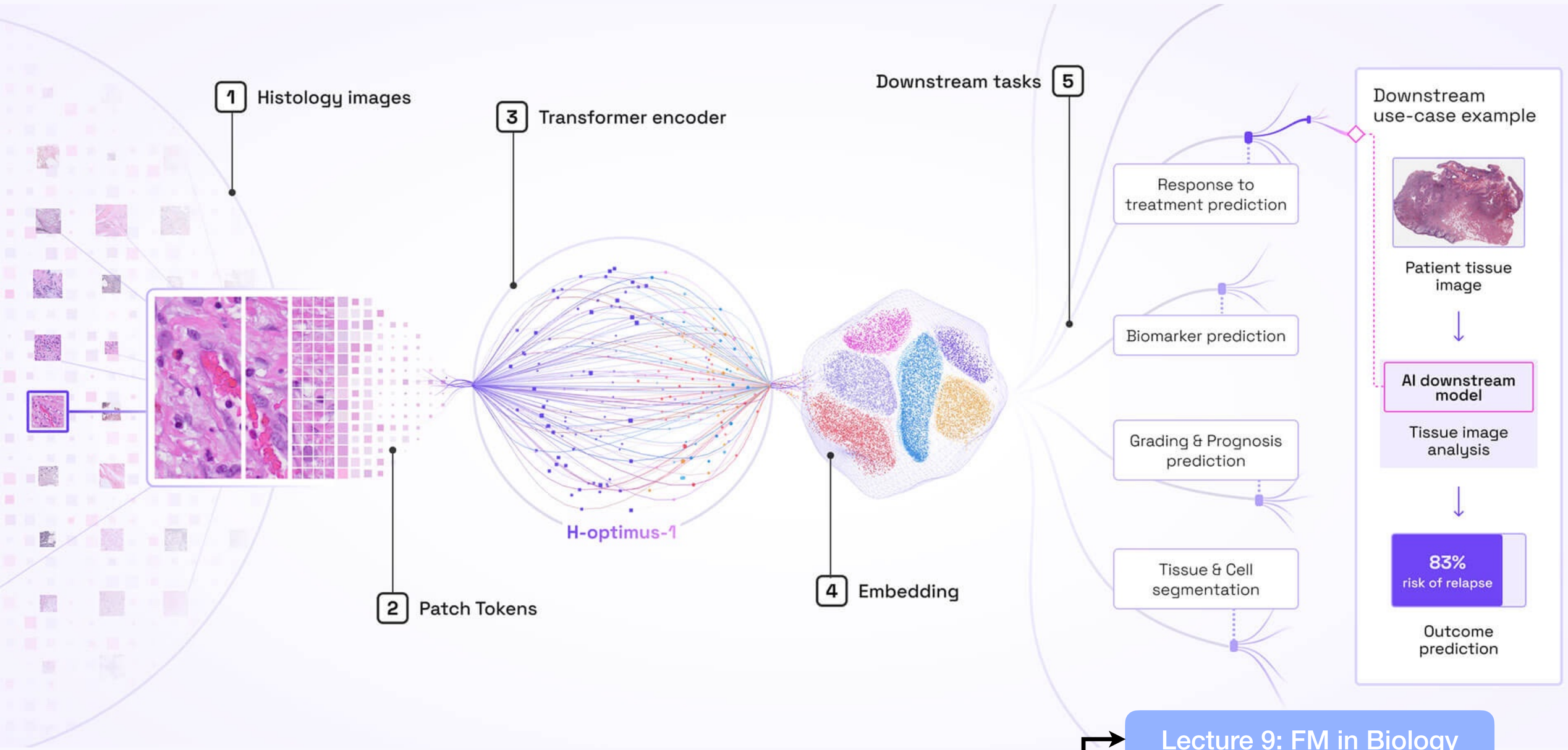


Segmentation

… and more!

Lecture 10: Emergent Behaviors

Foundation Models for Medicine



Lecture 9: FM in Biology

Masked Vision Models

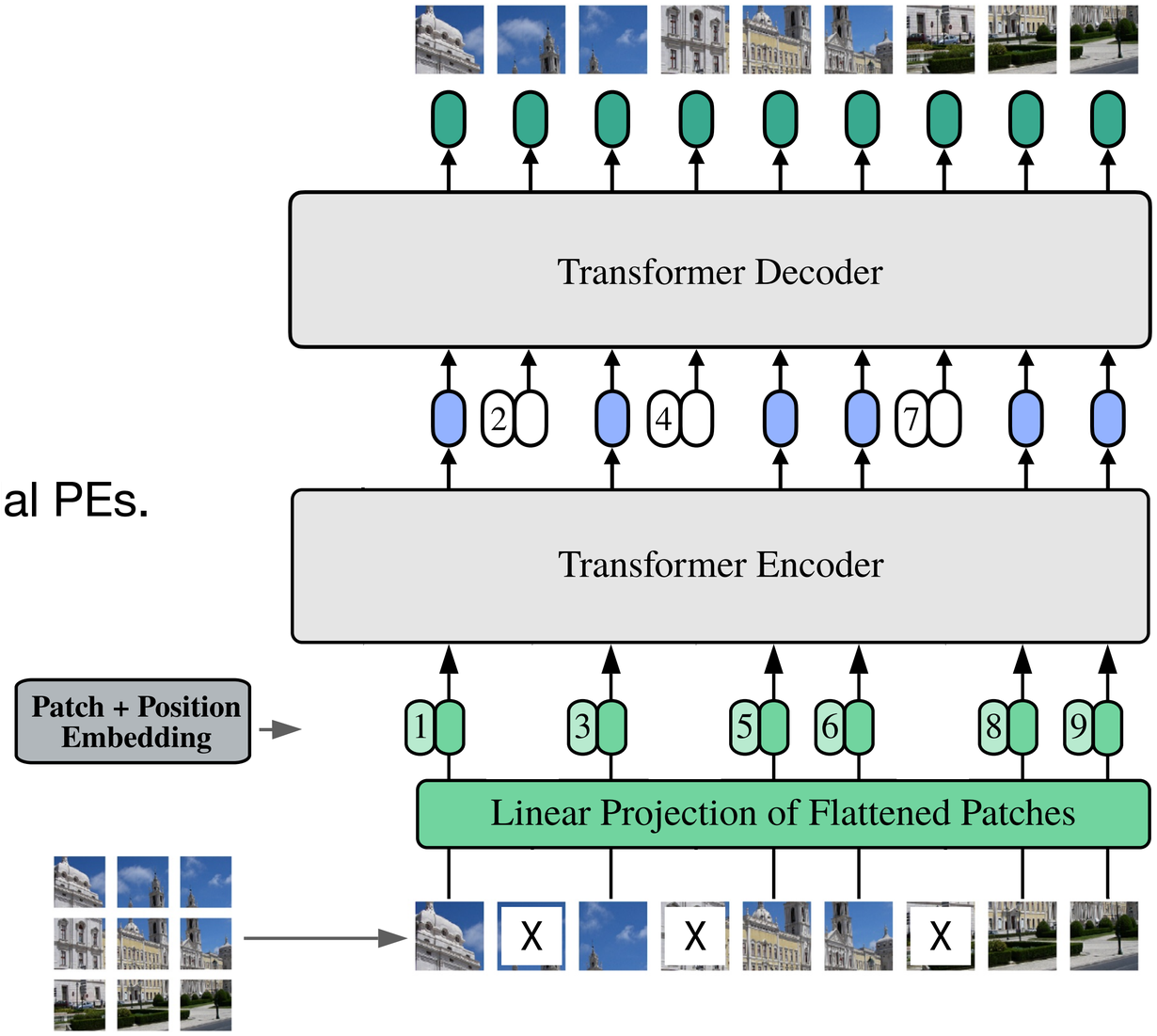
MAE: **Masked Autoencoders**

Goal: Learn strong representations
by reconstructing masked patches.

Encoder:
Operates on visible tokens and fixed 2D sinusoidal PEs.
No `[CLS]` summary token.

Decoder (lightweight):
Receives encoded visible tokens *and*
mask tokens (with positions)
and predicts the missing patches.

Loss: MSE on masked patches only.

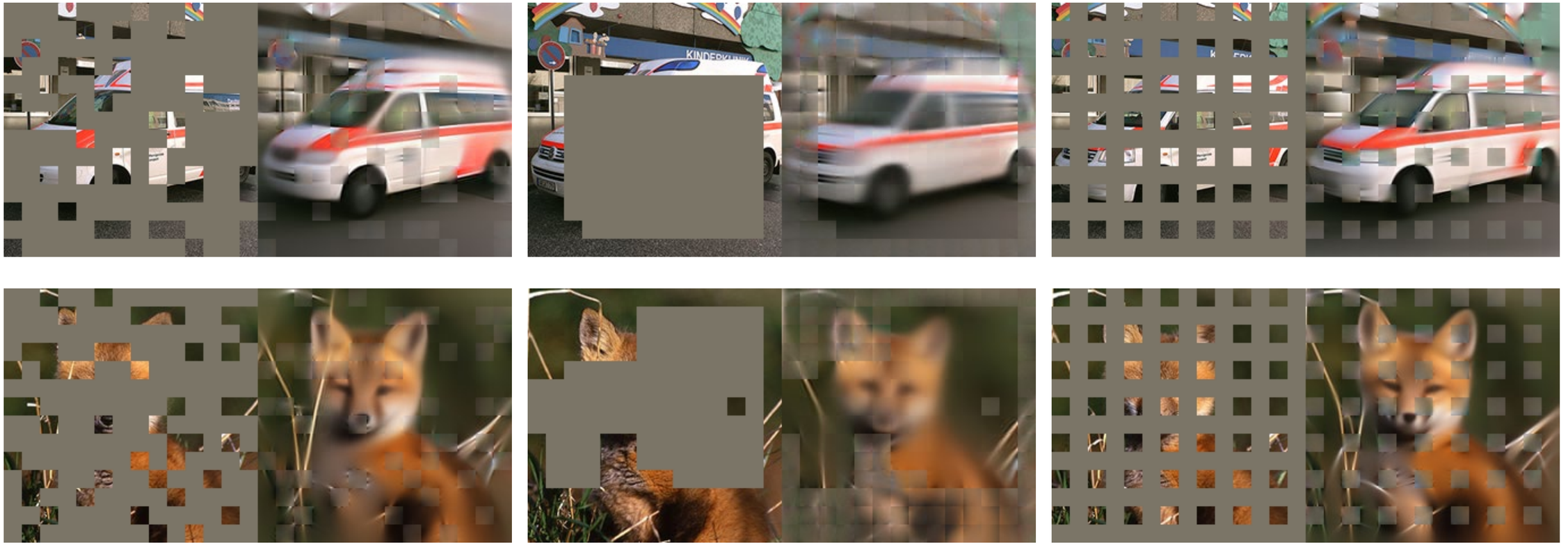


MAE: **Masked Autoencoders**

He et al., (2022)

Why does this work? High masking forces semantic, global understanding.

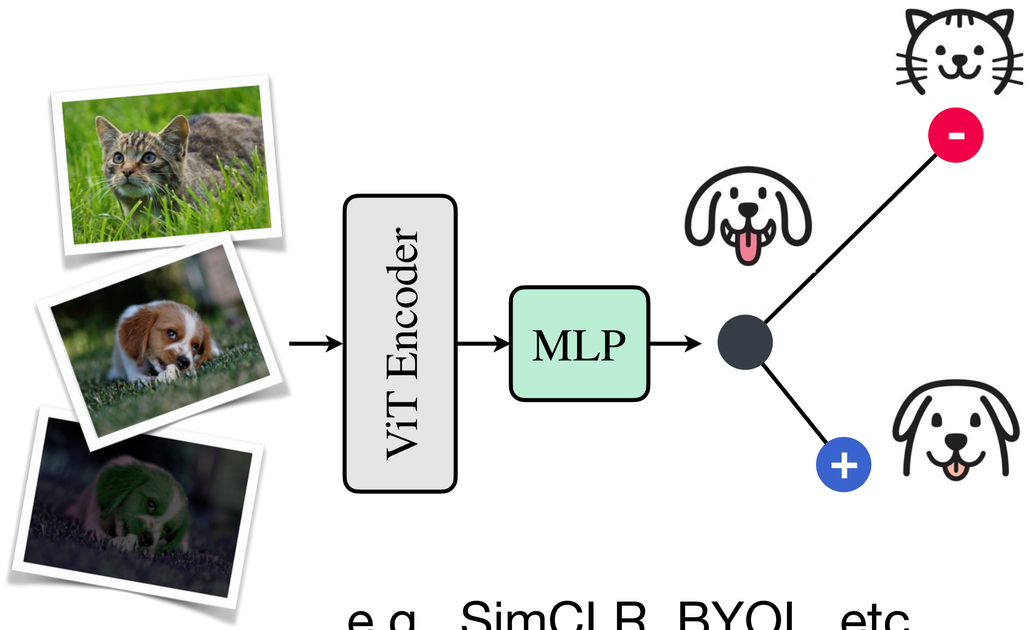
Mask sampling strategies determine the pretext task difficulty, influencing reconstruction quality and representations.



random 75% block 50% grid 75%

Contrastive Vision Models

Idea: Learn features that stay stable under natural image changes (crop, scale, color, viewpoint) by aligning representations of augmented versions of the same image; without labels.



e.g., SimCLR, BYOL, etc.

Problem!

Signal:
Optimizing *closeness* gave weak, uncalibrated targets → limited semantic pressure.

Stability:
Collapse control relied on fragile tricks; no mechanism to keep targets informative and balanced.

Data Views:
Two-crop setups underused scale/position variation → fewer supervision signals per image.

Scalability:
Dependence on negatives or huge batches increased brittleness and compute.

Contrastive Vision Models

Problem? If a network learns to match its *own* outputs on augmented views, it can drift or collapse to trivial solutions.

Idea!

Knowledge Distillation

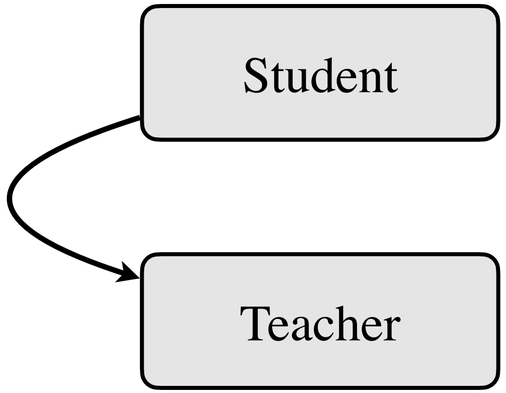
Train a student to imitate a teacher's soft predictions (probabilities, not hard labels).

→ Soft targets provide graded hints about which categories are close vs. far, which shapes a richer information than 0/1 labels.

Self-Distillation

Use no external teacher. Instead, make the teacher a slow, time-averaged copy of the student.

→ Think of it as an ensemble of the student's recent selves that produces steadier, higher-quality targets.



Exponential Moving Average (EMA) of Student

Contrastive Vision Models

The DINO Family!

= **S**elf-**D**istillation with **No** Labels

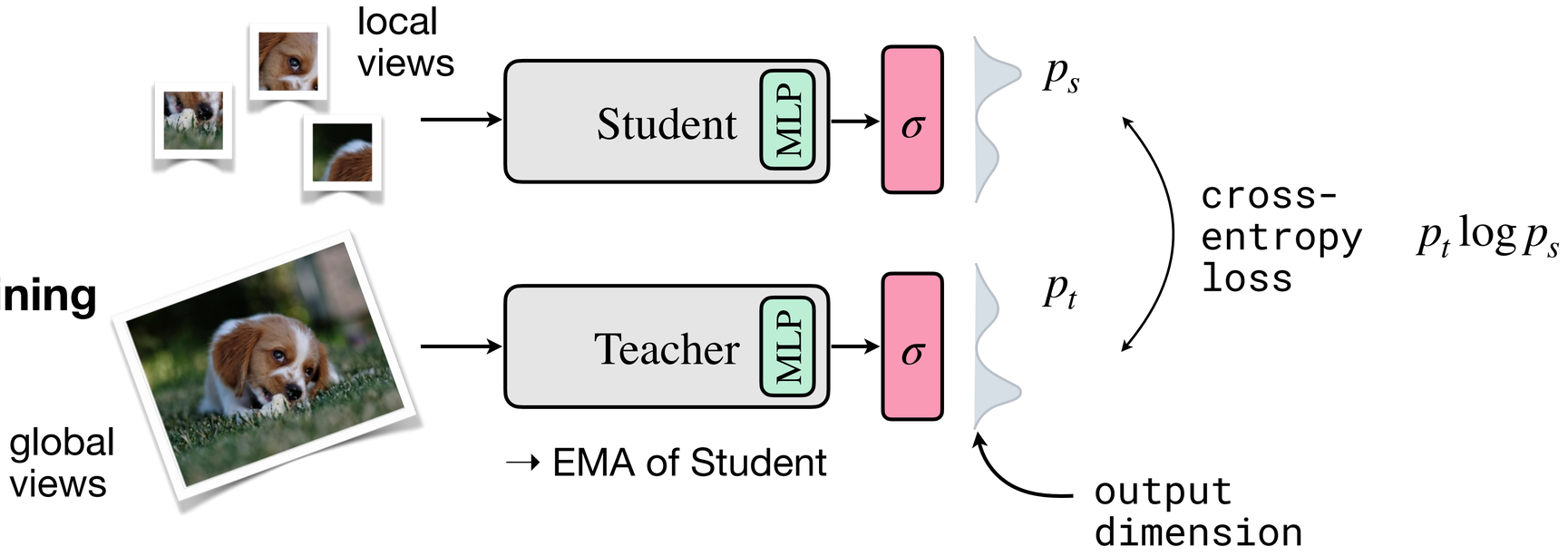
How do we train the teacher?

→ update with student weights.

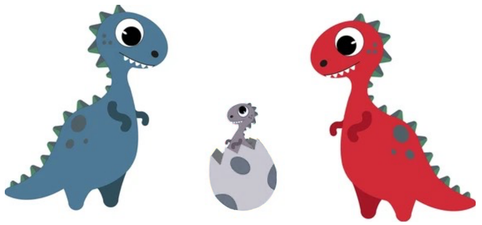
slow! $\theta_t \leftarrow \lambda\theta_t + (1-\lambda)\theta_s$

Why should the teacher be better?

→ **Multi-Crop Training**



→ student uses local and global, teacher only global views.

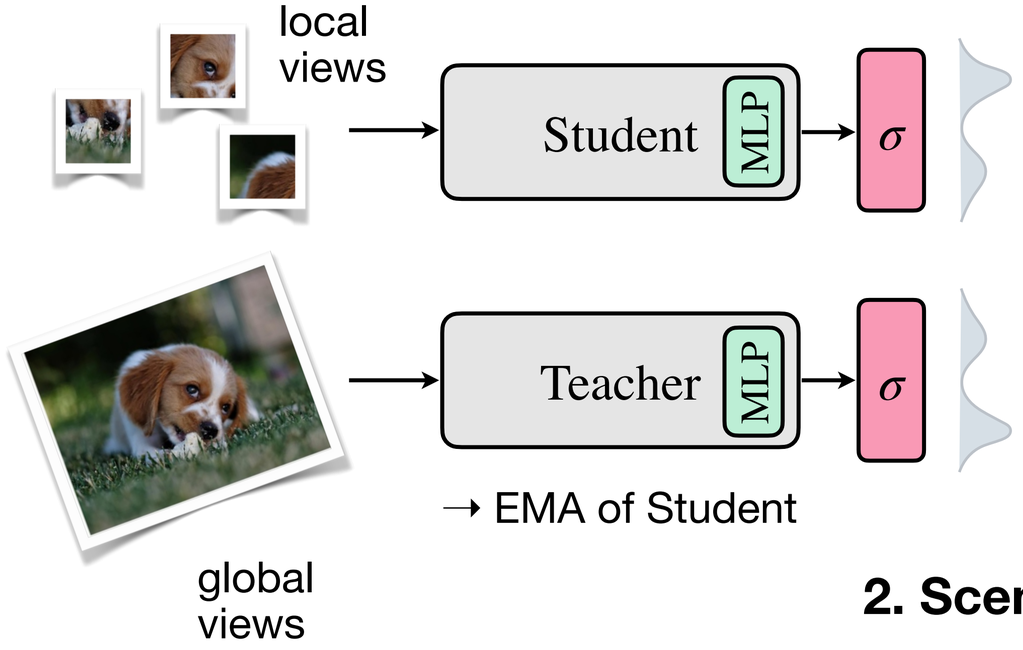


Contrastive Vision Models

How to avoid collapse?

DINOv1

Caron et al., (2021)



1. Scenario: p_s and p_t are uniform.

$$\sigma = P_s(x)^i = \frac{\exp\left(f_{\theta_s}(x)^i / \tau_s\right)}{\sum_{k=1}^{K} \exp\left(f_{\theta_s}(x)^k / \tau_s\right)}$$

→ temperature controls sharpness of the output distribution

2. Scenario: p_s and p_t are dominated by one dimension.

Centering $f_t(x) \leftarrow f_t(x) + c$ prevents this!

Contrastive Vision Models

DINOv1

Caron et al., (2021)

Supervised

DINO

Stable Teacher:
EMA teacher changes slowly → prevents drift/collapse.

Distributional Supervision:
Soft targets encode which alternatives are closer/farther, not just “be similar.”

Match across Crops:
Teacher (global) vs. student (other crop) must agree → invariance to position/scale/appearance.

Multi-Crop Boost:
2 global (~224^2) + 8–10 local (~96^2) views per image → many consistency constraints.

Augmentations Change Appearance:
Rand-crop, flip, color jitter, grayscale, blur → backgrounds change, object signal persists.

Contrastive Vision Models

DINOv1

> Beyond the fact that adapting self-supervised methods to [the vision transformer] architecture works particularly well, we make the following observations: first, self-supervised ViT features contain explicit information about the semantic segmentation of an image, which does not emerge as clearly with supervised ViTs, nor with ConvNets.

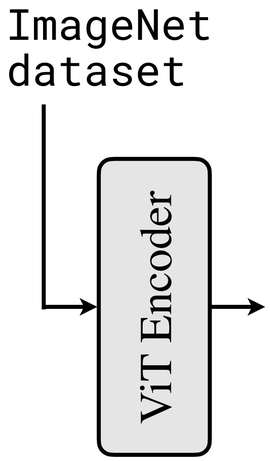
Caron et al., (2021)

Contrastive Vision Models

DINOv1

epoch: 0

Caron et al., (2021)



Semantic Regularities:

DINO discovers consistent object patterns and shared visual traits across images, yielding meaningful representations without labels.

Interpretable Clustering:

In its embedding space, ImageNet classes form coherent, human-like groupings and similar categories cluster together.

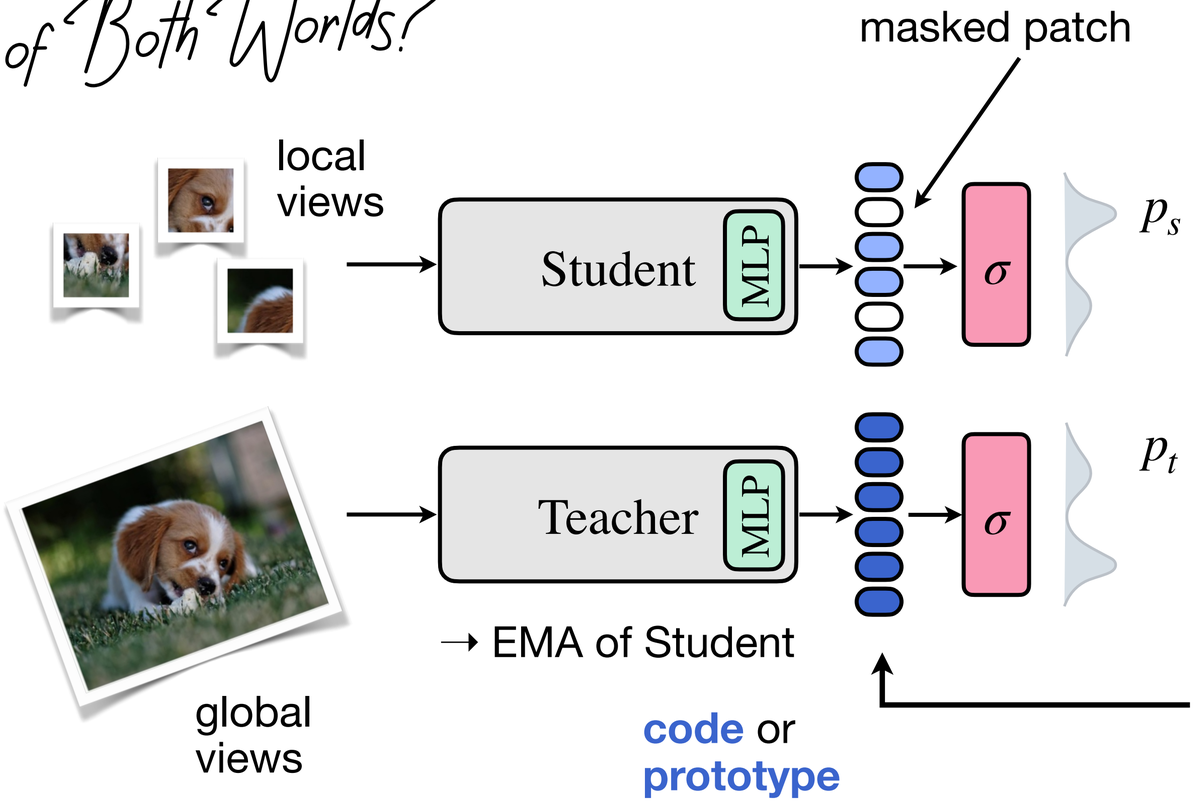
Lecture 10: Emergent Behaviors

Hybrid Vision Models: Combining Distillation with Masking

iBOT

DINO captures global, image-level semantics but underrepresents fine detail,
whereas **MAE** recovers detail yet often optimizes pixel appearance rather than semantic content.

Best of Both Worlds?



3. Masked Patch Prediction

For each masked patch, find the corresponding patch in the teacher's view (via crop alignment).

2. Global Distribution Matching

Train the student to match the teacher's token distribution via cross-entropy.

DINO

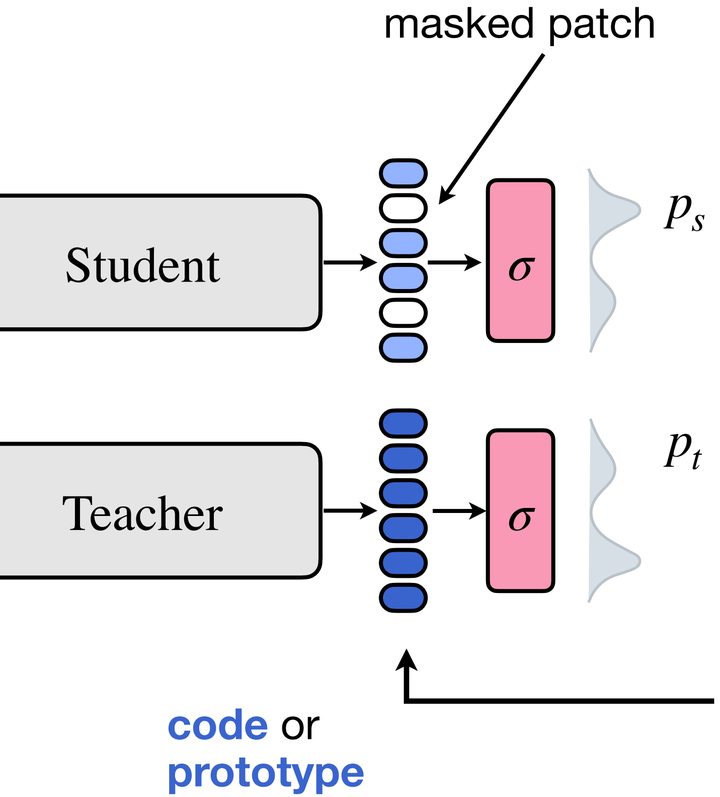
1. Online Tokenizer

→ a distribution over a learnable codebook.

Hybrid Vision Models: Combining Distillation with Masking

Zhou et al., (2022)

iBOT



3. Masked Patch Prediction

For each masked patch, find the corresponding patch in the teacher's view (via crop alignment).

→ Forces *informative* patch embeddings!

2. Global Distribution Matching

Train the student to match the teacher's token distribution via cross-entropy.

1. Online Tokenizer

→ a distribution over a learnable codebook.

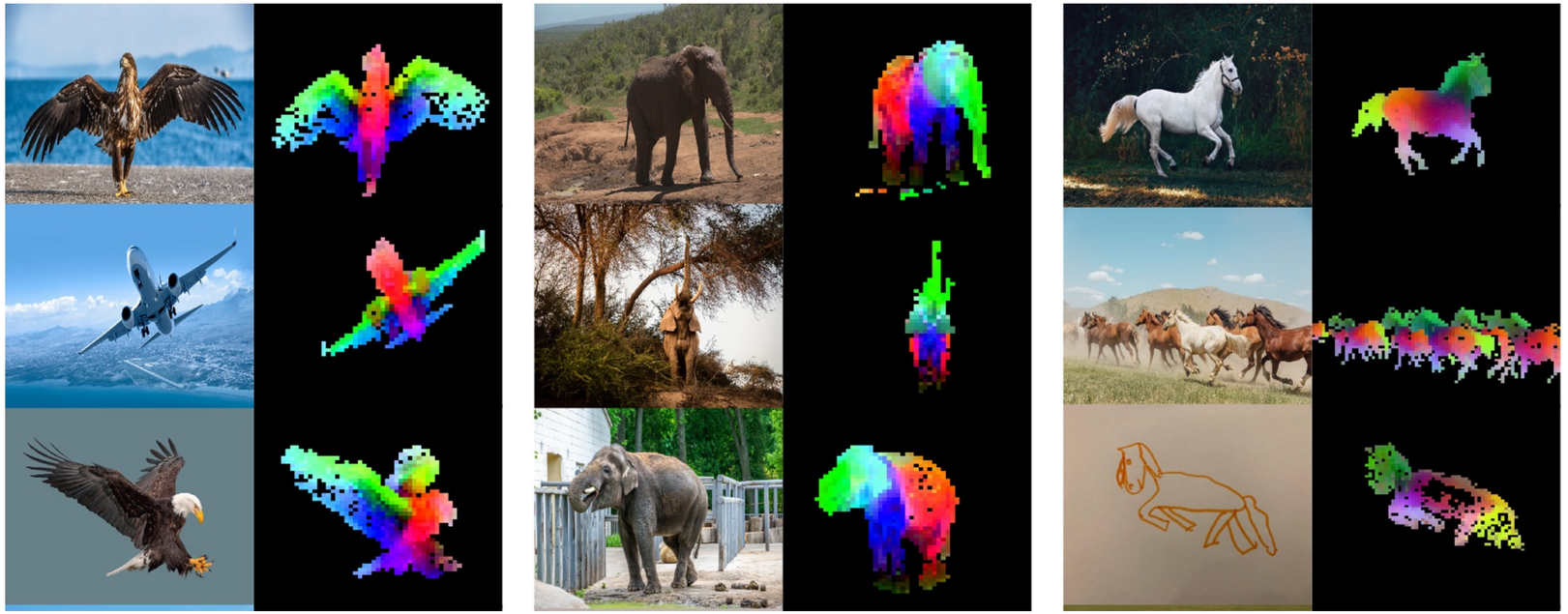
→ **Semantic per-patch targets!**

Turns each teacher patch feature into prototype distribution (learned codebook).

Hybrid Vision Models: Industrial Scaling

DINOv2

> We revisit existing approaches and combine different techniques to scale our pretraining in terms of data and model size. Most of the technical contributions aim at accelerating and stabilizing the training at scale. In terms of data, we propose an automatic pipeline to build a dedicated, diverse, and curated image dataset instead of uncurated data, as typically done in the self-supervised literature.



Oquab et al., (2024)

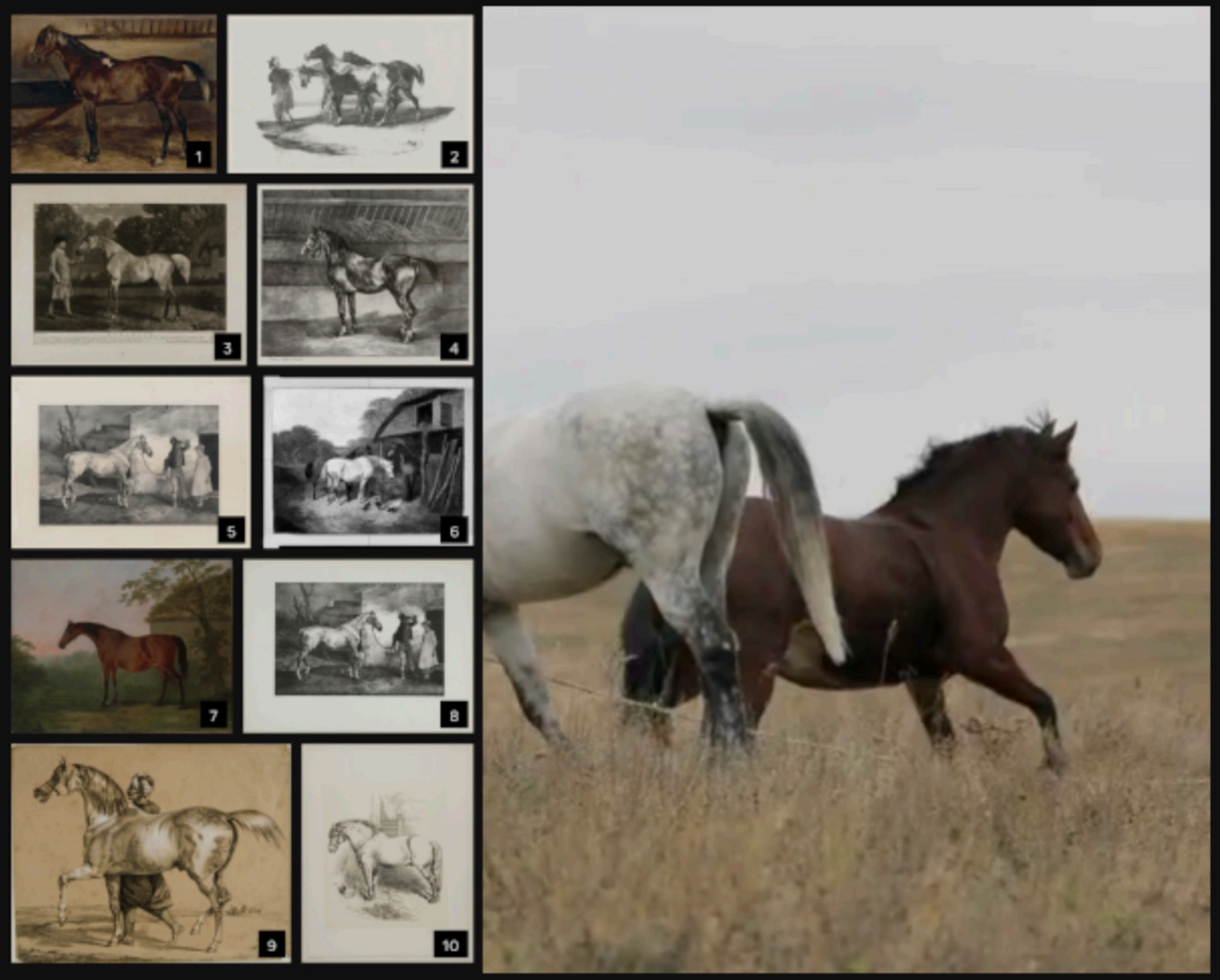
Large pretraining dataset of 142 million images.

Depth Estimation

State-of-the-art results and strong generalization on estimating depth from a single image.

Semantic Segmentation

Competitive results without any fine-tuning on clustering an images into object classes.



Instance Retrieval

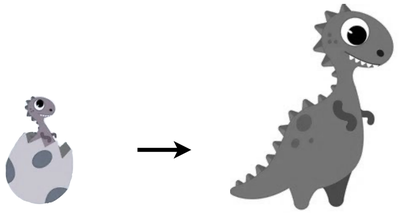
Directly use frozen features to find art pieces similar to a given image from a large art collection.

Lecture 10: Emergent Behaviors

Hybrid Vision Models: Industrial Scaling

DINOv3

Siméoni et al., (2025)



Contributions based on 1. **data scaling**

Too Much Data!

> We build our large-scale pre-training dataset by leveraging a large data pool of web images collected from public posts on **Instagram** […] and we obtain an initial data pool of approximately 17 billions of images.

Automatic Curation Method
based on `hierarchical k-means` and `balanced sampling`

→ to guarantee a balanced coverage of all visual concepts.

Retrieval-Based Curation system

→ Create a dataset that covers visual concepts relevant for downstream tasks.

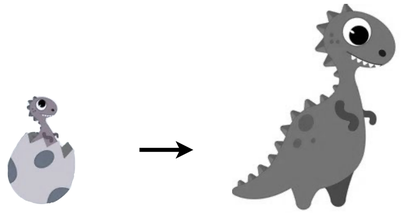
Publicly available computer vision datasets

→ to optimize model performance.

Hybrid Vision Models: Industrial Scaling

DINOv3

Siméoni et al., (2025)

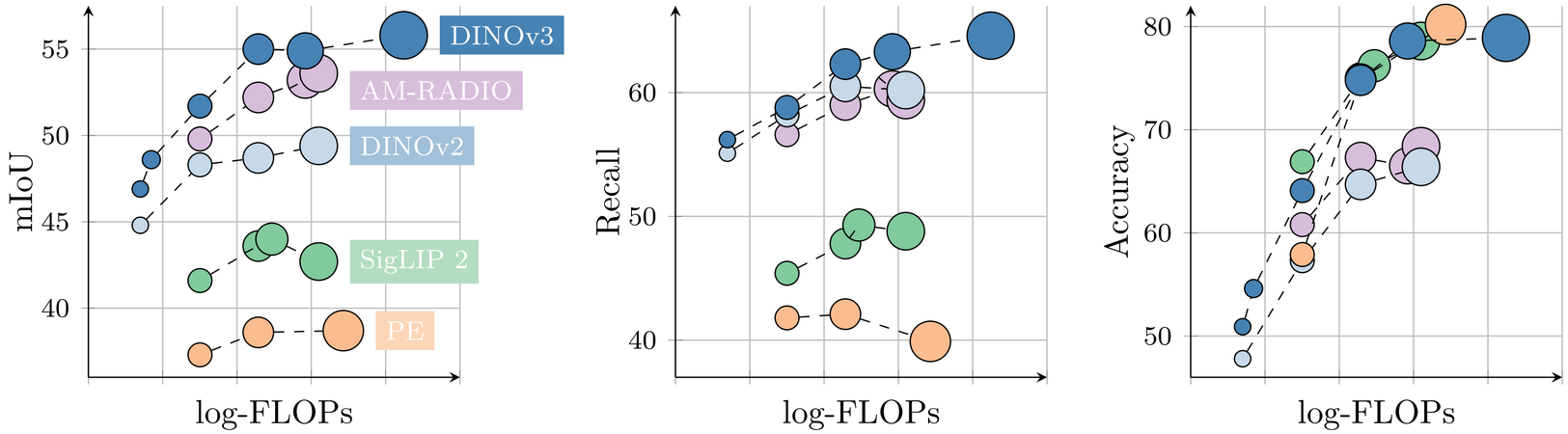


Contributions based on 1. **data scaling**

> We build our large-scale pre-training dataset by leveraging a large data pool of web images collected from public posts on **Instagram** […] and we obtain an initial data pool of approximately 17 billions of images.

2. **model scaling**

> We increase our main model size to 7B parameters by defining a custom variant of the ViT architecture. We include modern position embeddings (axial RoPE).

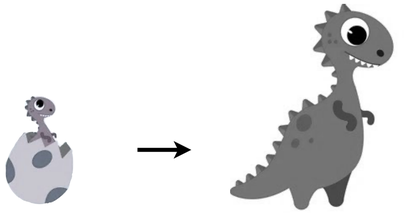


Semantic segmentation (ADE20k) | 3D keypoint matching (NAVI) | OOD classif. (ObjectNet)

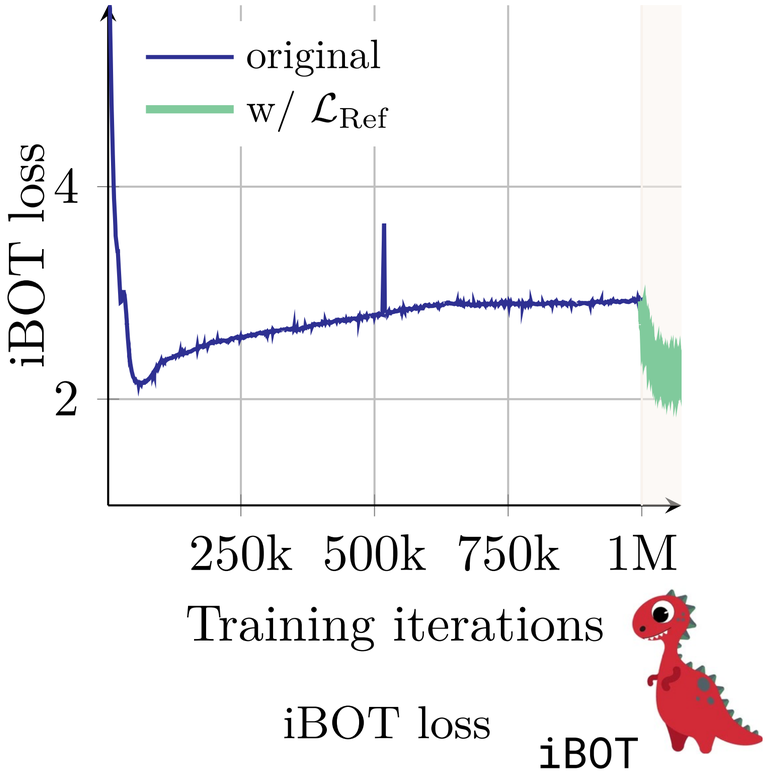
Hybrid Vision Models: Industrial Scaling

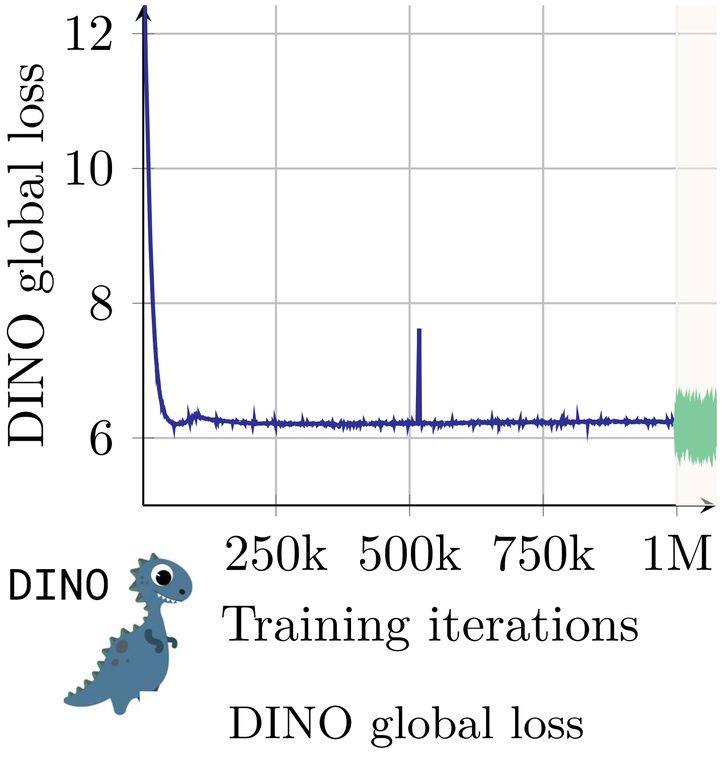
DINOv3

Siméoni et al., (2025)



Trade-off between learning global vs. local content.





`iBOT` and `DINO` trade off global vs. local signal:

- `DINO` loss (global distillation)
  → stronger image-level semantics.
- `iBOT` loss (masked patch tokens)
  → stronger dense/local features.

Too much of one hurts the other.

- Over-weighting `iBOT` can erode global classification quality.
- Over-weighting `DINO` (especially late in long runs) lets dense maps drift/forget.

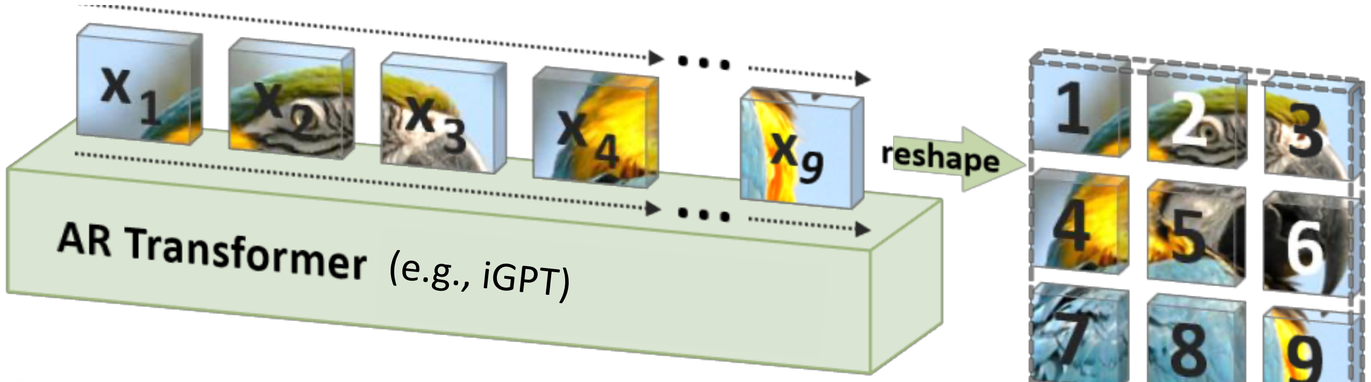
Practical Recipe:

1. Use both losses: apply DINO on global crops and iBOT on masked patches with fixed weights throughout training.
2. Add a feature anchoring term in the later training phase to keep patch representations stable during long runs.

Autoregressive Vision Models

iGPT

Chen et al., (2020)



↳ Standard causal Transformer (autoregressive mask).

Idea:
- Treat an image as a sequence of tokens, similar to words in text.
- Learn to predict the next pixel (or patch) given all previous ones.
- Captures global dependencies via next-token prediction.

Limitations:
- Computationally heavy for high-resolution images (long sequences).
- Struggles to capture 2D spatial structure efficiently.

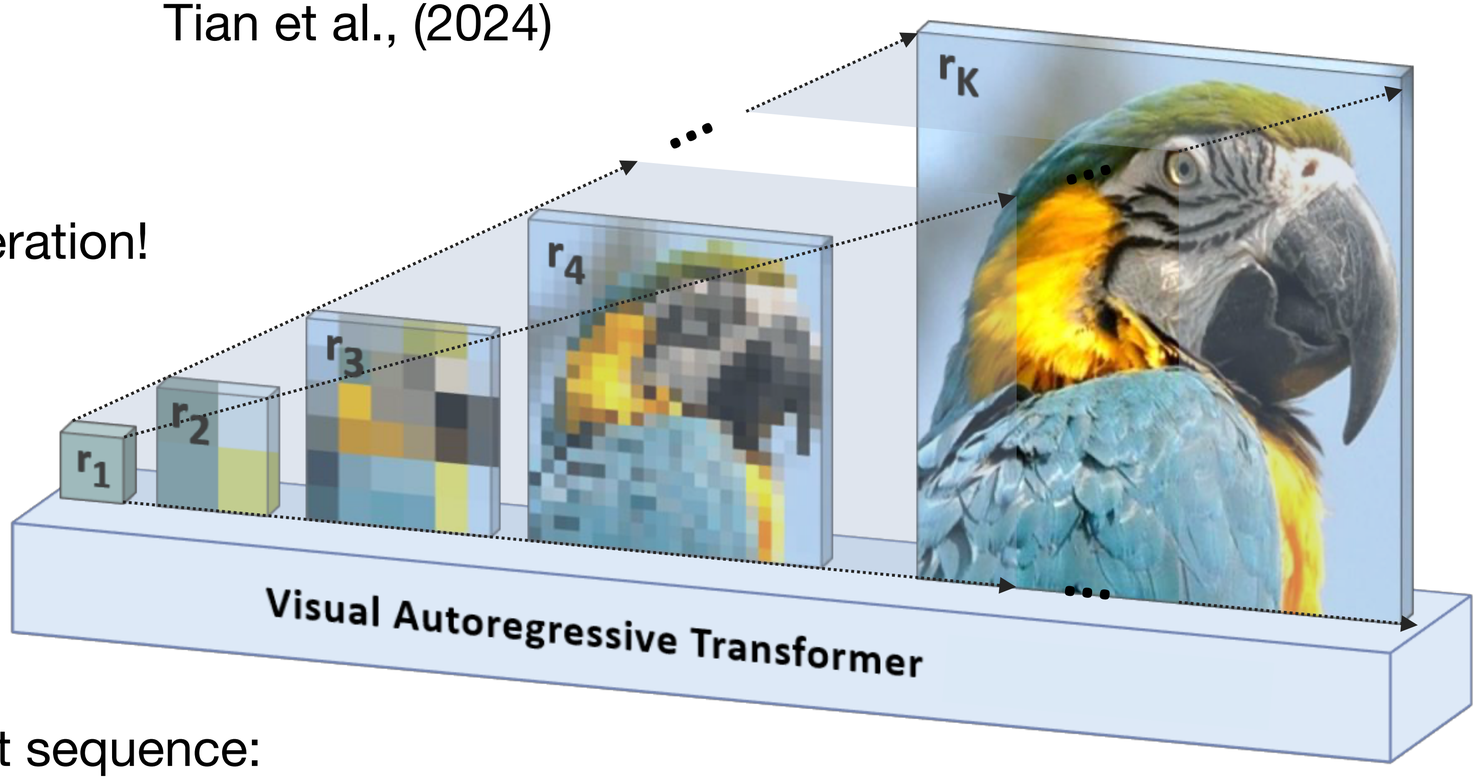
Autoregressive Vision Models

VAR

Tian et al., (2024)

Visual **A**uto**r**egressive Model

Idea: Hierarchical, multi-scale generation!



VAR introduces a hierarchical latent sequence:

- **Coarse-to-fine prediction** across multiple resolutions.
- Each level r_k refines the image conditioned on lower-resolution context.

Models global semantics first, then local details.

Lecture 8: Multimodality

This Week's Papers

Papers are linked in Moodle.

- Radford et al., "Improving Language Understanding by Generative Pre-Training" OpenAI Blog (2018).
- Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (2019).
- Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." *International Conference on Learning Representations (ICLR)* (2021).
- Oquab et al., "DINOv2: Learning Robust Visual Features without Supervision." Transactions on Machine Learning Research (TMLR) (2024).

Week 8's Exercise Sheet

Exercise 6 · Task 1

Differences among BERT, T5, and GPT Models.

Compare the three Transformer families by architecture, positional encoding, and training objective. Explain how encoder-only, encoder-decoder, and decoder-only models differ in attention flow and masking; outline each model's positional encoding and its trade-offs; and summarize their objectives, i.e., BERT's masked LM, T5's span corruption with sentinels, and GPT's autoregressive prediction.

Pen-and-paper exercises for Vision FMs the week after!

Week 8's Code Demonstration

Code Notebook 6 · Task 1

Implementation of BERT, T5, and GPT

This exericise walks through the architecture and training pipelines for BERT, T5 and GPT models. Concretely, it covers encoder, decoder, encoder-decoder architecture implementations, their different training objectives with inputs and targets construction.

→ Jupyter notebook exercise

Coding exercises for Vision FMs the week after!

CS-461

Foundation Models and Generative AI

Have a great week!