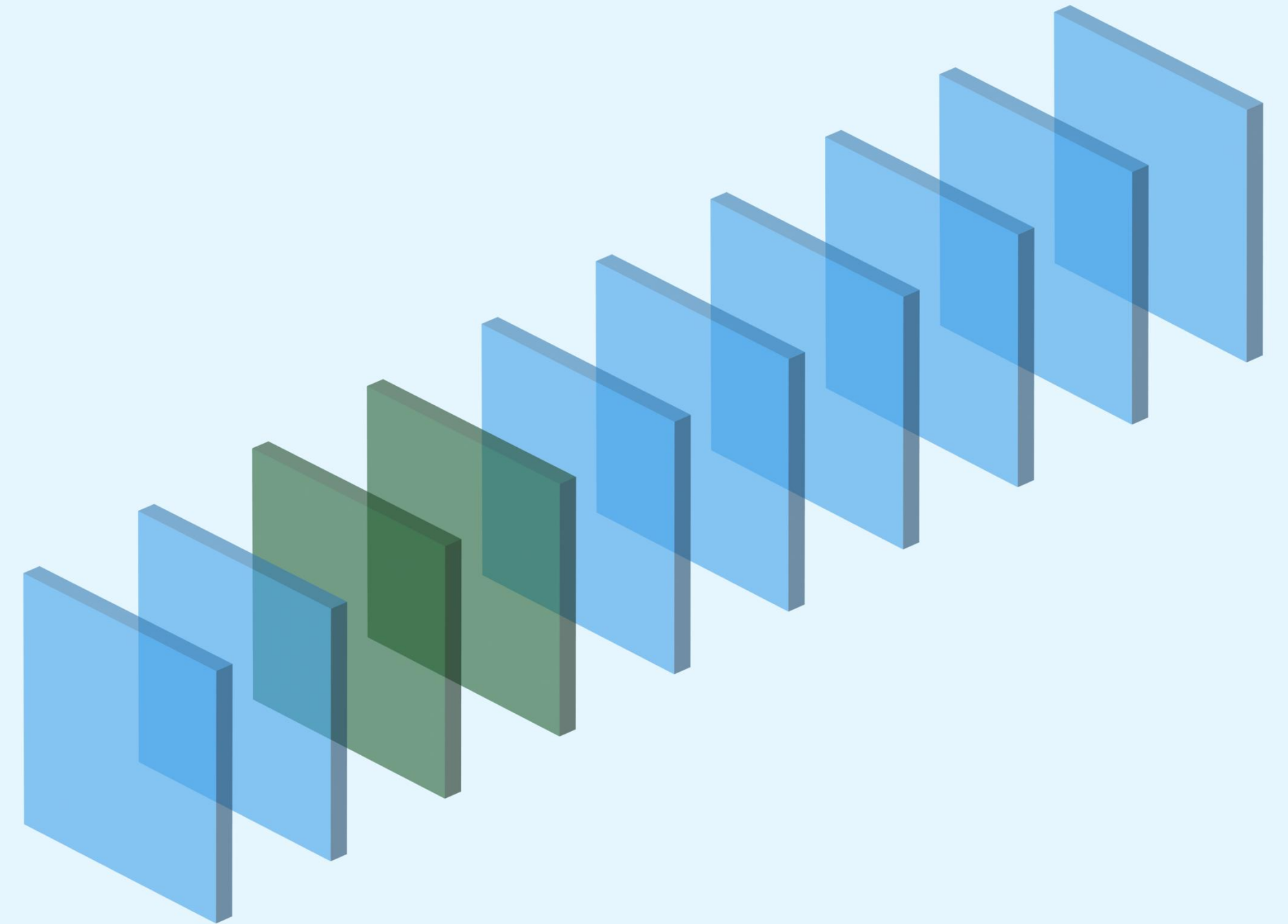


# *Geospatial Foundation Models: Overview, Application and Benchmarking*

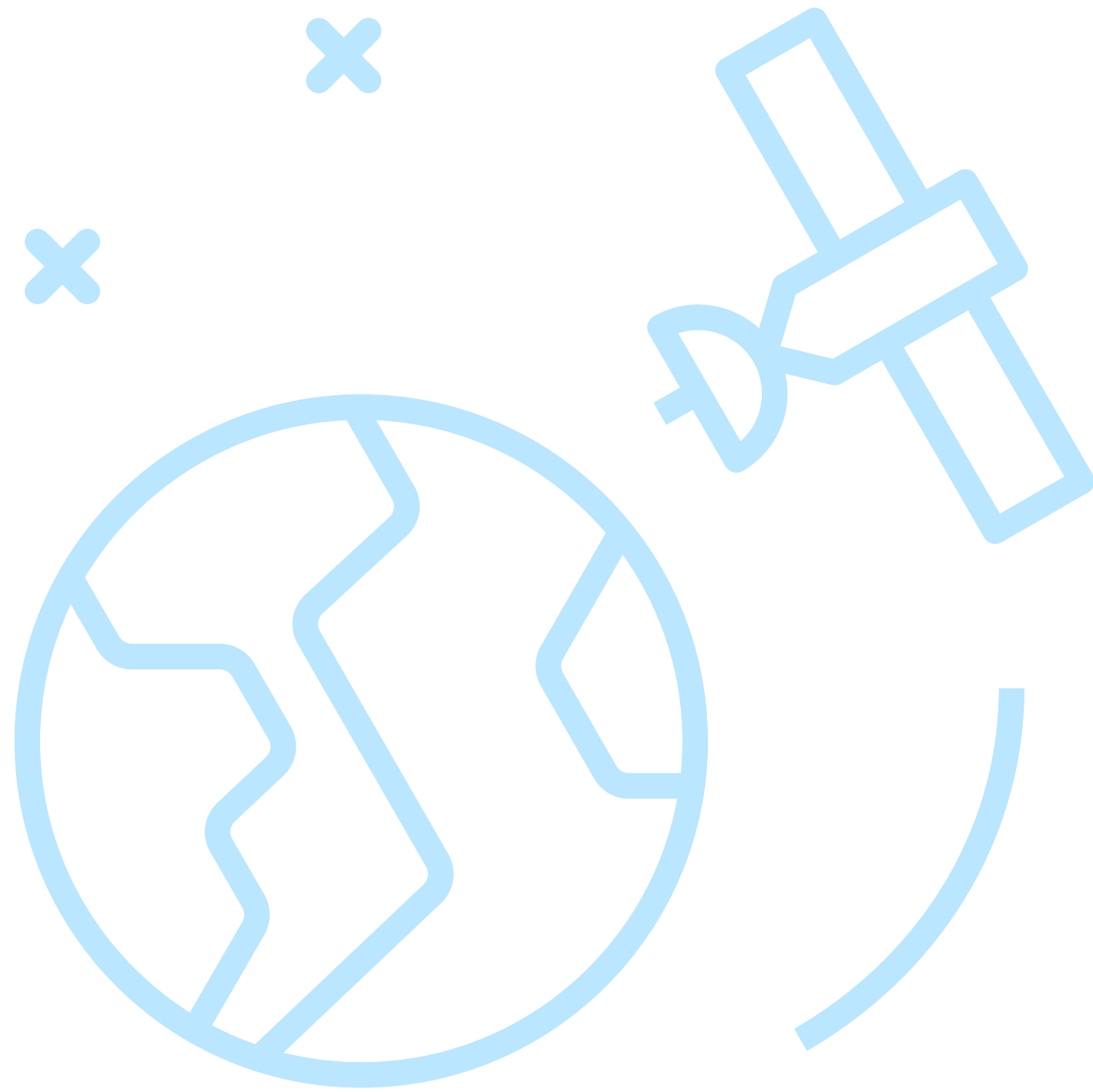
Bianca Zadrozny  
Senior Research Manager

João Lucas de Sousa Almeida  
Research Software Engineer

Daniela Szwarcman  
Staff Research Scientist



# Outline



Introduction

Foundation models in computer vision

Geospatial foundation models (GeoFMs)

Multimodal GeoFMs

Fine-tuning GeoFMs in practice with TerraTorch

Benchmarking GeoFMs

Closing/QA

# Introduction

# Timeline

Data size



## Simple networks

Task-specific hand-crafted feature representations

## Deep networks

Learnt feature representations

## Foundation models

Generalizable & adaptable learnt representations



# Foundation models

- *Pre-trained* on large datasets, leveraging self-supervised techniques.
- Learn *generalizable* and *adaptable* data representations which can be effectively used in multiple *downstream tasks*.

# Foundation models

- *Pre-trained* on large datasets, leveraging self-supervised techniques.
- Learn *generalizable* and *adaptable* data representations which can be effectively used in multiple *downstream tasks*.

*“Transfer learning is what makes foundation models possible, but *scale* is what makes them powerful”.<sup>1</sup>*

*Pre-training*: a model is trained on a surrogate task and then adapted to the downstream task of interest via a second training phase called *fine-tuning*.

# Foundation models

- *Pre-trained* on large datasets, leveraging self-supervised techniques.
- Learn *generalizable* and *adaptable* data representations which can be effectively used in multiple *downstream tasks*.

*“Transfer learning is what makes foundation models possible, but *scale* is what makes them powerful”.*<sup>1</sup>

*Pre-training*: a model is trained on a surrogate task and then adapted to the downstream task of interest via a second training phase called *fine-tuning*.

Based on *deep neural networks* and *self-supervised* learning at *scale*.

Self-supervised learning → scalable

540

IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, VOL. 28, NO. 4, JULY 1990

# Neural Network Approaches Versus Statistical Methods in Classification of Multisource Remote Sensing Data

JON A. BENEDIKTSSON, STUDENT MEMBER, IEEE, PHILIP H. SWAIN, SENIOR MEMBER, IEEE, AND OKAN K. ERSOY, MEMBER, IEEE

6232

IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, VOL. 54, NO. 10, OCTOBER 2016

# Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks

Yushi Chen, *Member, IEEE*, Hanlu Jiang, Chunyang Li, Xiuping Jia, *Senior Member, IEEE*, and Pedram Ghamisi, *Member, IEEE*



# Remote sensing + ML

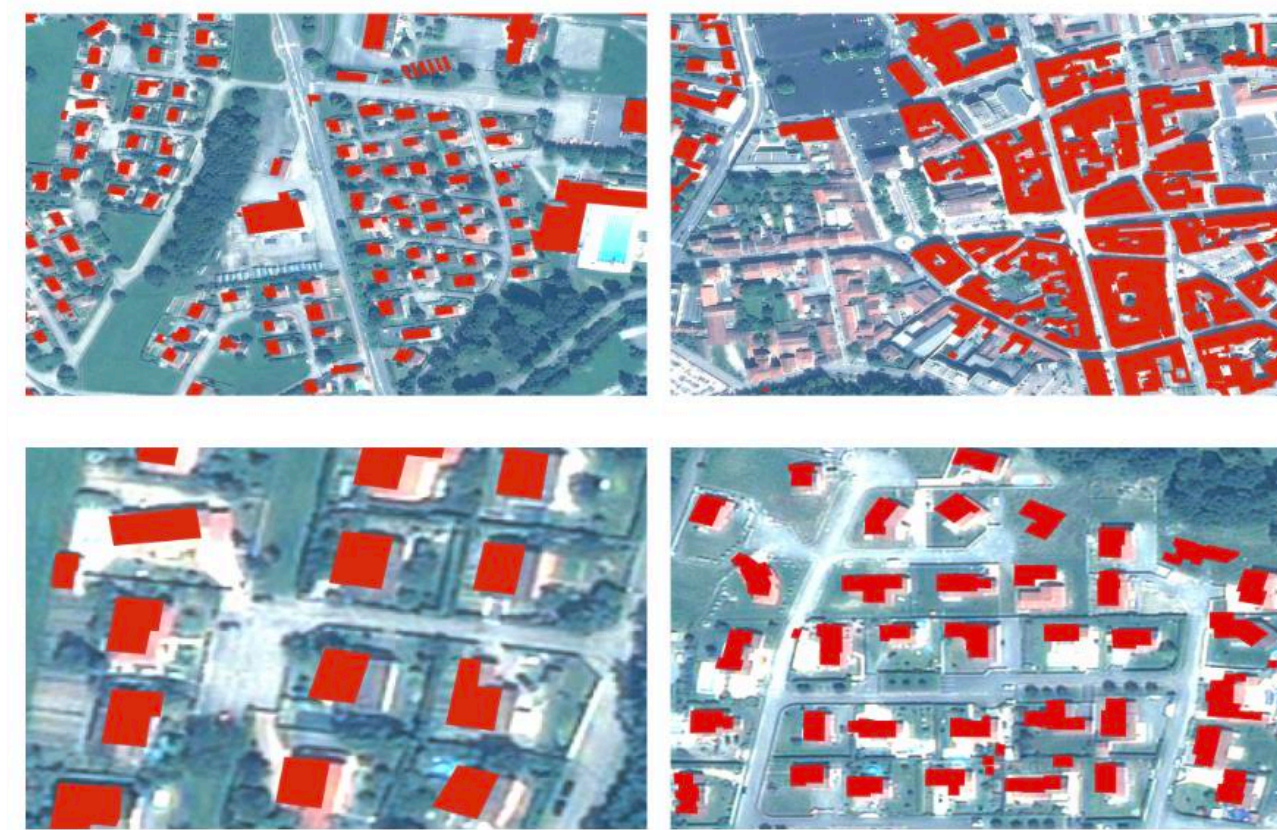
## Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification<sup>1</sup>

- Fully *convolutional network* for dense predictions
- 2-step training approach:
  1. Train on raw OSM labeled data (large)
  2. Train on manually labeled data (small)

Fine-tuning

*“The overall success of CNNs lies mostly in the fact that the networks are forced by construction to learn *hierarchical* contextual *translation-invariant* features, which are particularly useful for image categorization”.*<sup>1</sup>

Data for step 1



Data for step 2



Pictures extracted from (1)



# Remote sensing + ML

Jack-bo1220 / Awesome-Remote-Sensing-Foundation-Models

CodeIssues 7Pull requests 2ActionsProjectsSecurityInsights

Awesome-Remote-Sensing-Foundation-ModelsPublic

Watch 43Fork 87Star 899

main1 BranchTags

Go to fileAdd fileCode

Jack-bo1220Update README.md6ac1b80 · 2 weeks ago145 Commits

README.mdUpdate README.md2 weeks ago

README

Maintained? yesawesomeWatchers 43Stars 899Forks 87

# Awesome Remote Sensing Foundation Models

🌟 A collection of papers, datasets, benchmarks, code, and pre-trained weights for Remote Sensing Foundation Models (RSFMs).

## Latest Updates

🔥🔥🔥 Last Updated on 2024.10.15 🔥🔥🔥

- 2024.10.15: Update PANGAEA and TEOChat.
- 2024.10.04: Update SAR-JEPA.
- 2024.10.02: Update PIS.
- 2024.9.29: Update EarthMarker.

About

No description, website, or top provided.

ReadmeActivity899 stars43 watching87 forksReport repository

Releases

No releases published

Packages

No packages published

Contributors 3

Jack-bo1220

danielz02Chenhui Zhang

baichuanzhoubaichuanzhou

README

Remote Sensing <u>Vision</u> Foundation Models				
Abbreviation	Title	Publication	Paper	Code & Weights
GeoKR	Geographical Knowledge-Driven Representation Learning for Remote Sensing Images	TGRS2021	<a href="#">GeoKR</a>	<a href="#">link</a>
-	Self-Supervised Learning of Remote Sensing Scene Representations Using Contrastive Multiview Coding	CVPRW2021	<a href="#">Paper</a>	<a href="#">link</a>
GASSL	Geography-Aware Self-Supervised Learning	ICCV2021	<a href="#">GASSL</a>	<a href="#">link</a>
SeCo	Seasonal Contrast: Unsupervised Pre-Training From Uncurated Remote Sensing Data	ICCV2021	<a href="#">SeCo</a>	<a href="#">link</a>
DINO-MM	Self-supervised Vision Transformers for Joint SAR-optical Representation Learning	IGARSS2022	<a href="#">DINO-MM</a>	<a href="#">link</a>
SatMAE	SatMAE: Pre-training Transformers for Temporal and Multi-Spectral Satellite Imagery	NeurIPS2022	<a href="#">SatMAE</a>	<a href="#">link</a>
RS-BYOL	Self-Supervised Learning for Invariant Representations From Multi-Spectral and SAR Images	JSTARS2022	<a href="#">RS-BYOL</a>	null
GeCo	Geographical Supervision Correction for Remote Sensing Representation Learning	TGRS2022	<a href="#">GeCo</a>	null
RingMo	RingMo: A remote sensing foundation model with masked image modeling	TGRS2022	<a href="#">RingMo</a>	<a href="#">Code</a>

IBM Research | © 2025 IBM Corporation

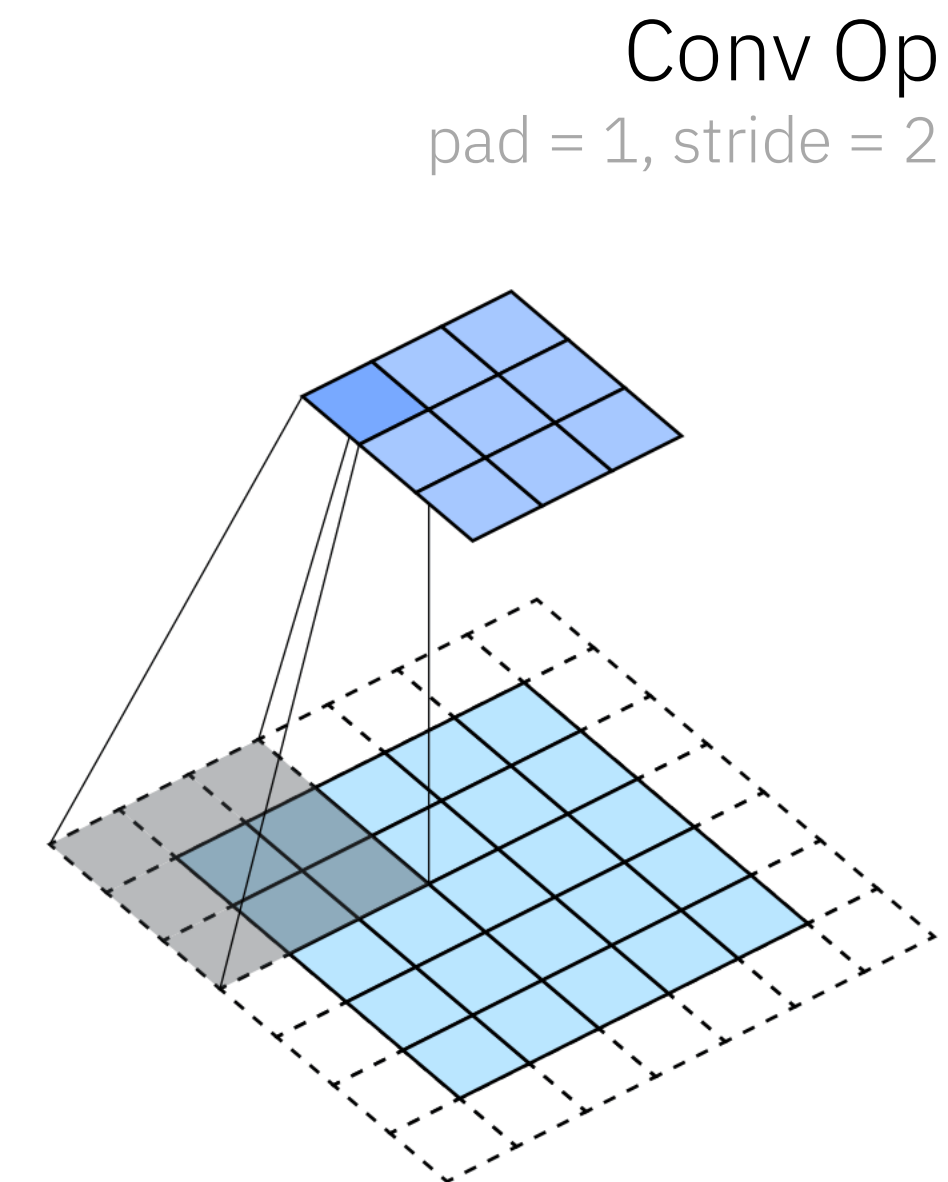
10

# Foundation models in computer vision

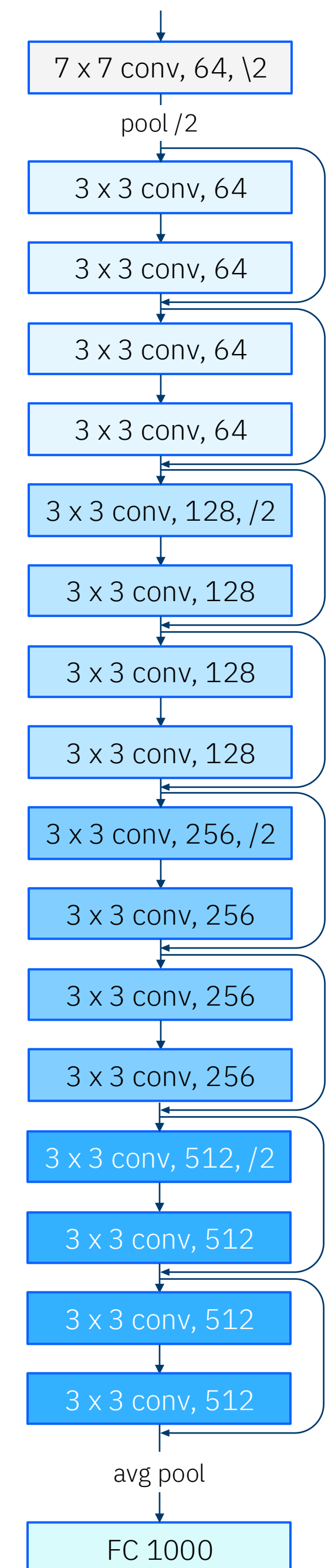
# From convolutional networks to vision transformers

## Convolutional neural networks (CNNs)

- inductive bias → *locality*, 2D neighborhood structure, and translation equivariance
- good at *dense* predictions
- efficient memory utilization (parameter sharing)
- good with relatively *small datasets*



ResNet18  
~11M params

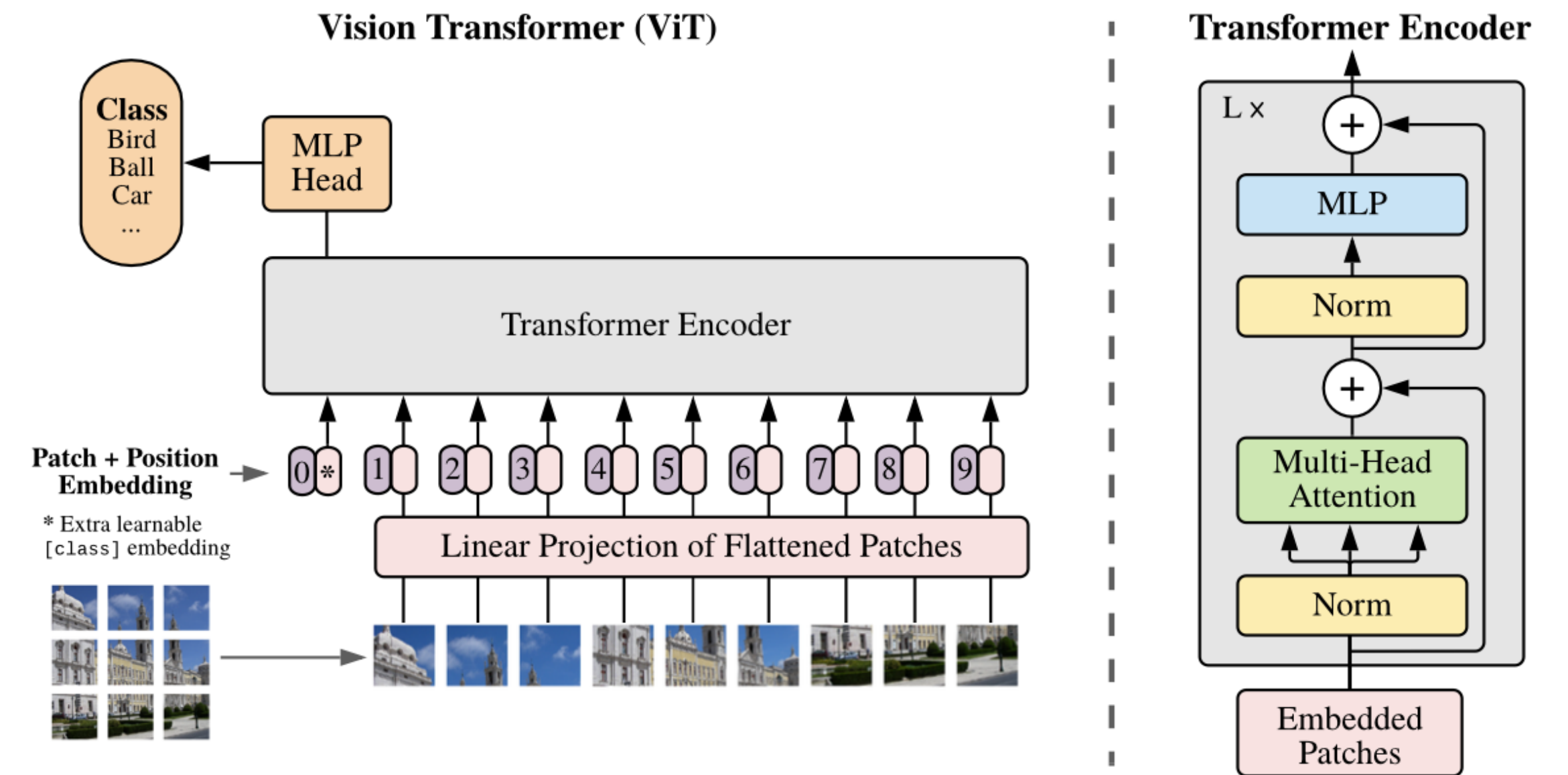




# From convolutional networks to vision transformers

## Vision Transformers

- inductive bias  $\rightarrow$  practically none
- good at capturing *global context*, and *long-range dependencies*
- good scalability with *large/huge datasets*



Vision transformer (ViT) architecture. Extracted from (1)

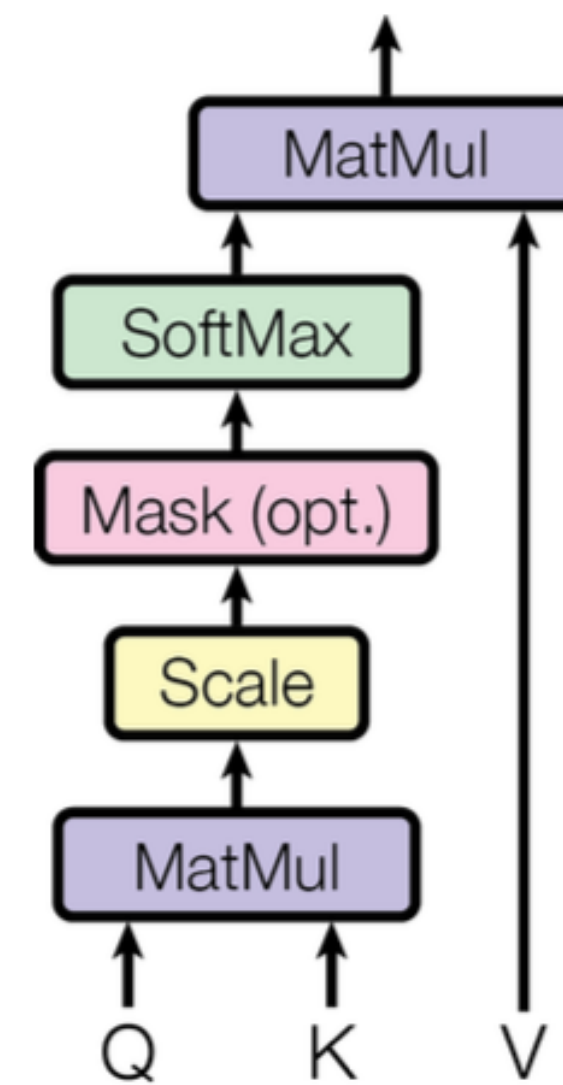
# Self-attention

The key component of vision transformers is the *self-attention* mechanism. It models the interactions between all parts of a sequence.

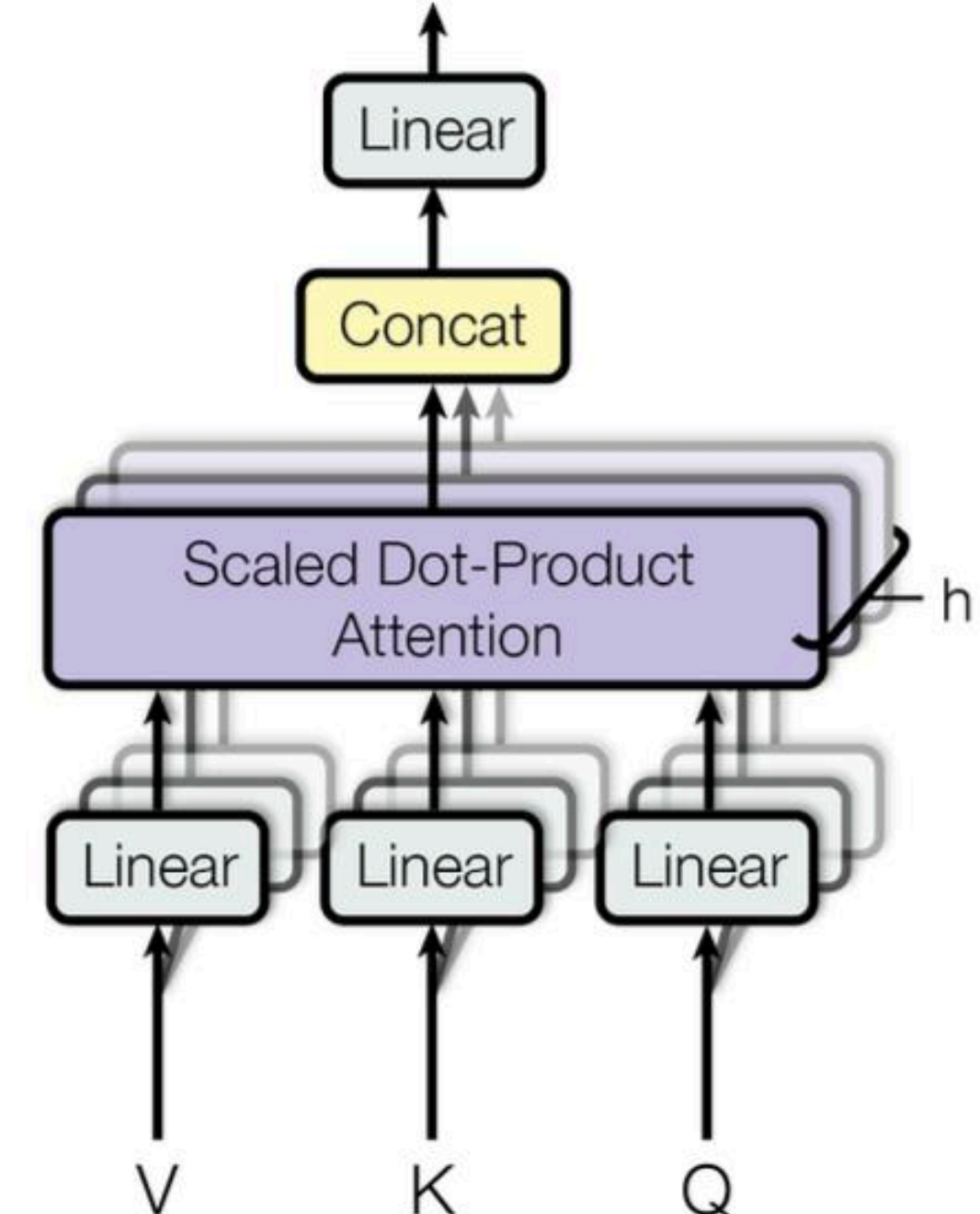
Multi-head attention → multiple attention mechanisms applied in parallel.

*“Multi-head attention allows the model to *jointly attend* to information from *different representation subspaces* at different positions. With a single attention head, averaging inhibits this”.<sup>1</sup>*

Scaled dot product attention



Multi-head attention



Self-attention mechanism. Adapted from (1)

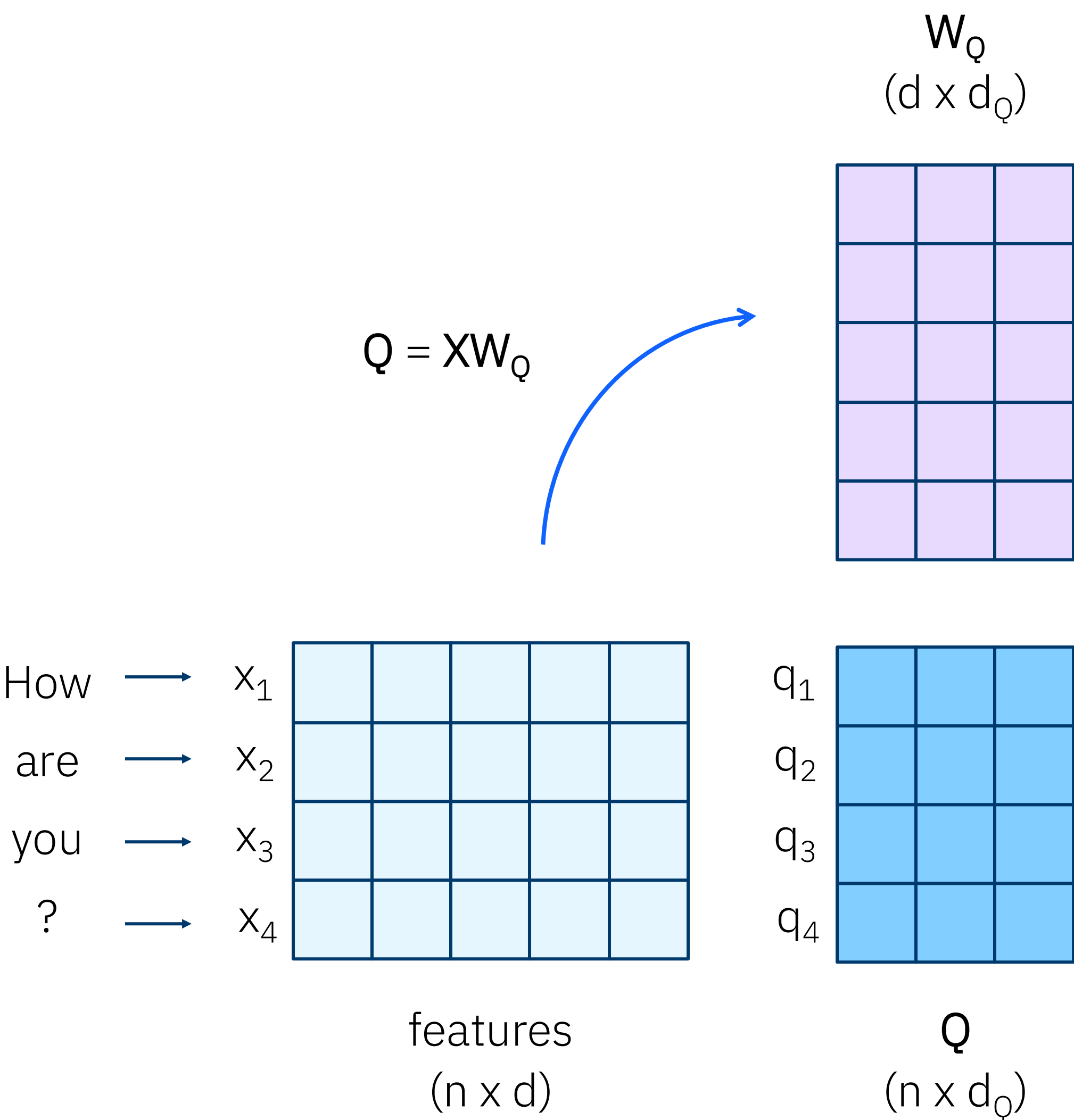
# Self-attention

Query, Key, and Value  $\rightarrow Q, K, V$

– Terminology borrowed from retrieval systems

– Steps:

- 1. project input into *learnable linear layers* to get  $Q, K, V$



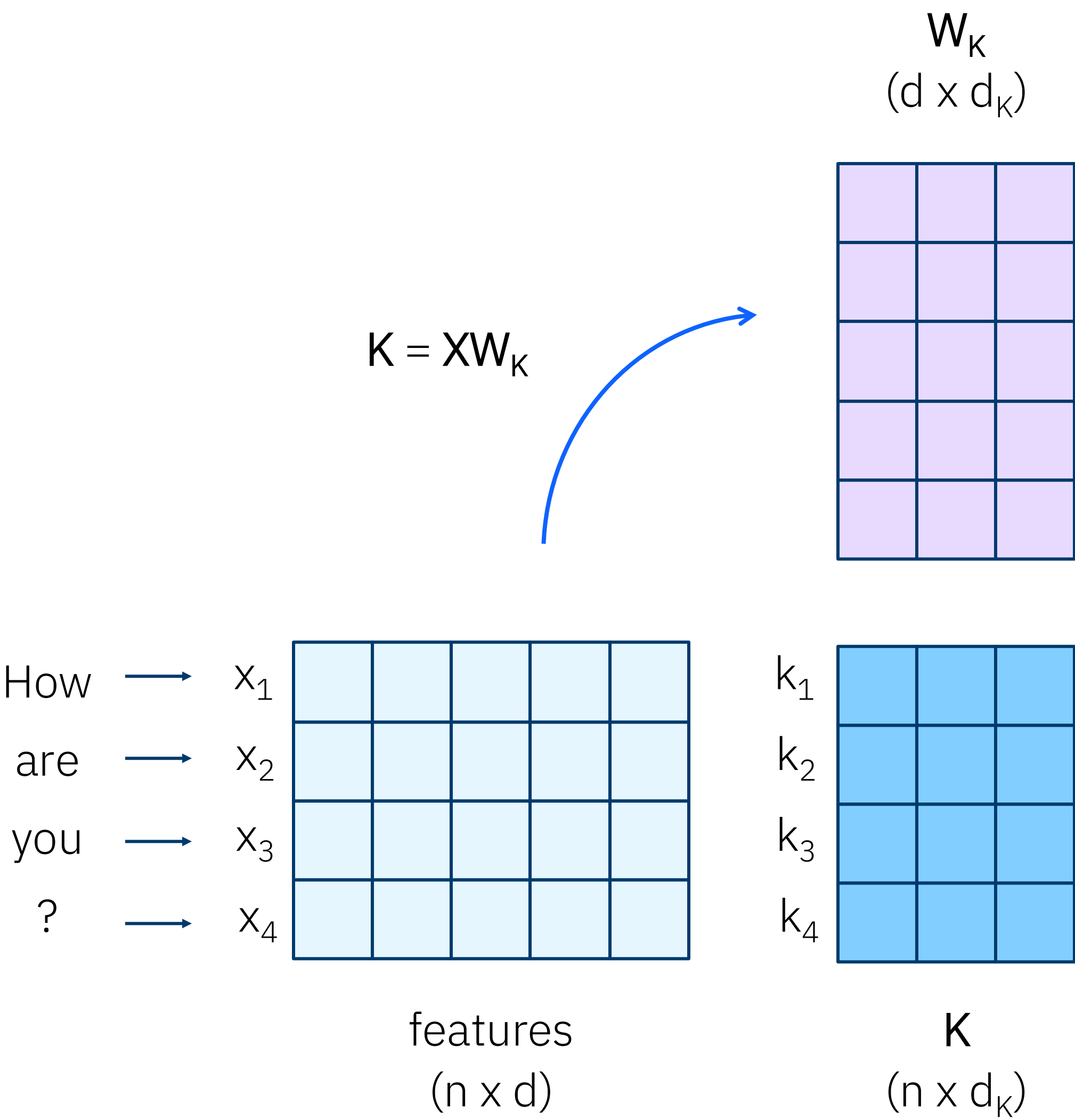
# Self-attention

Query, Key, and Value  $\rightarrow Q, K, V$

– Terminology borrowed from retrieval systems

– Steps:

- 1. project input into *learnable linear layers* to get  $Q, K, V$



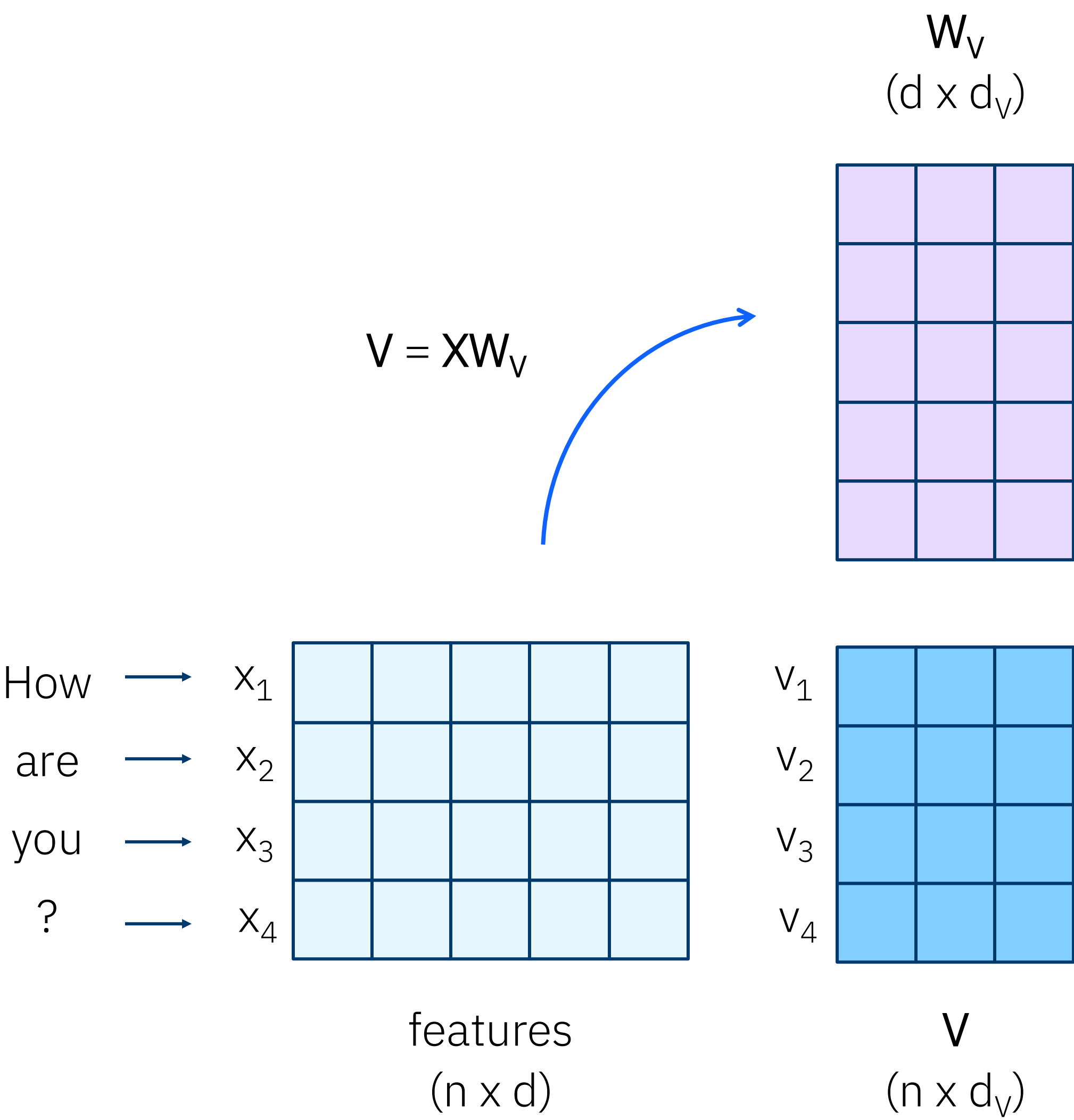
# Self-attention

Query, Key, and Value  $\rightarrow Q, K, V$

– Terminology borrowed from retrieval systems

– Steps:

- 1. project input into *learnable linear layers* to get  $Q, K, V$



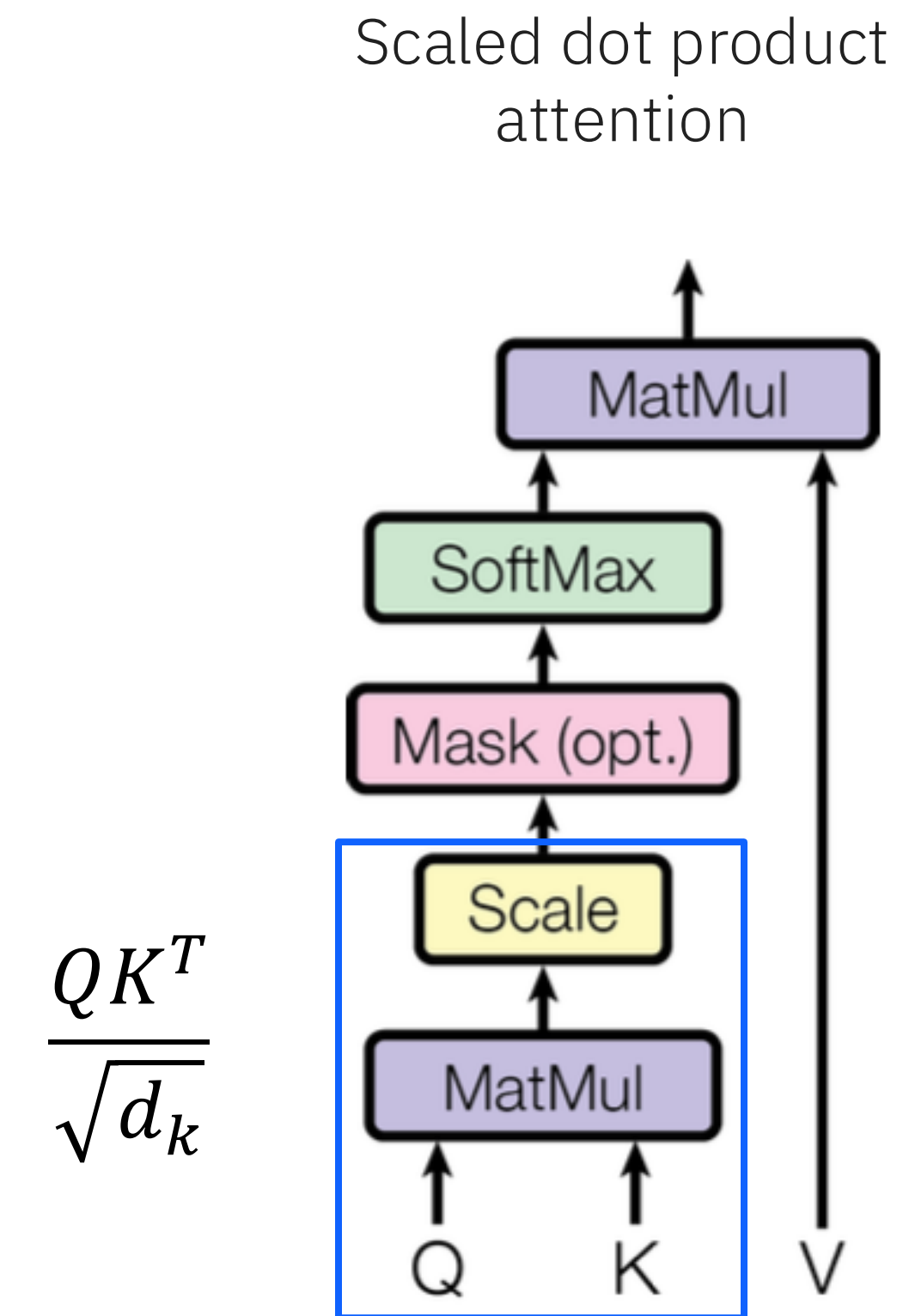
# Self-attention

Query, Key, and Value  $\rightarrow Q, K, V$

– Terminology borrowed from retrieval systems

– Steps:

1. project input into *learnable linear layers* to get  $Q, K, V$
2. compute *similarity* between  $Q$  and  $K$  and scale



Self-attention mechanism. Adapted from (1)

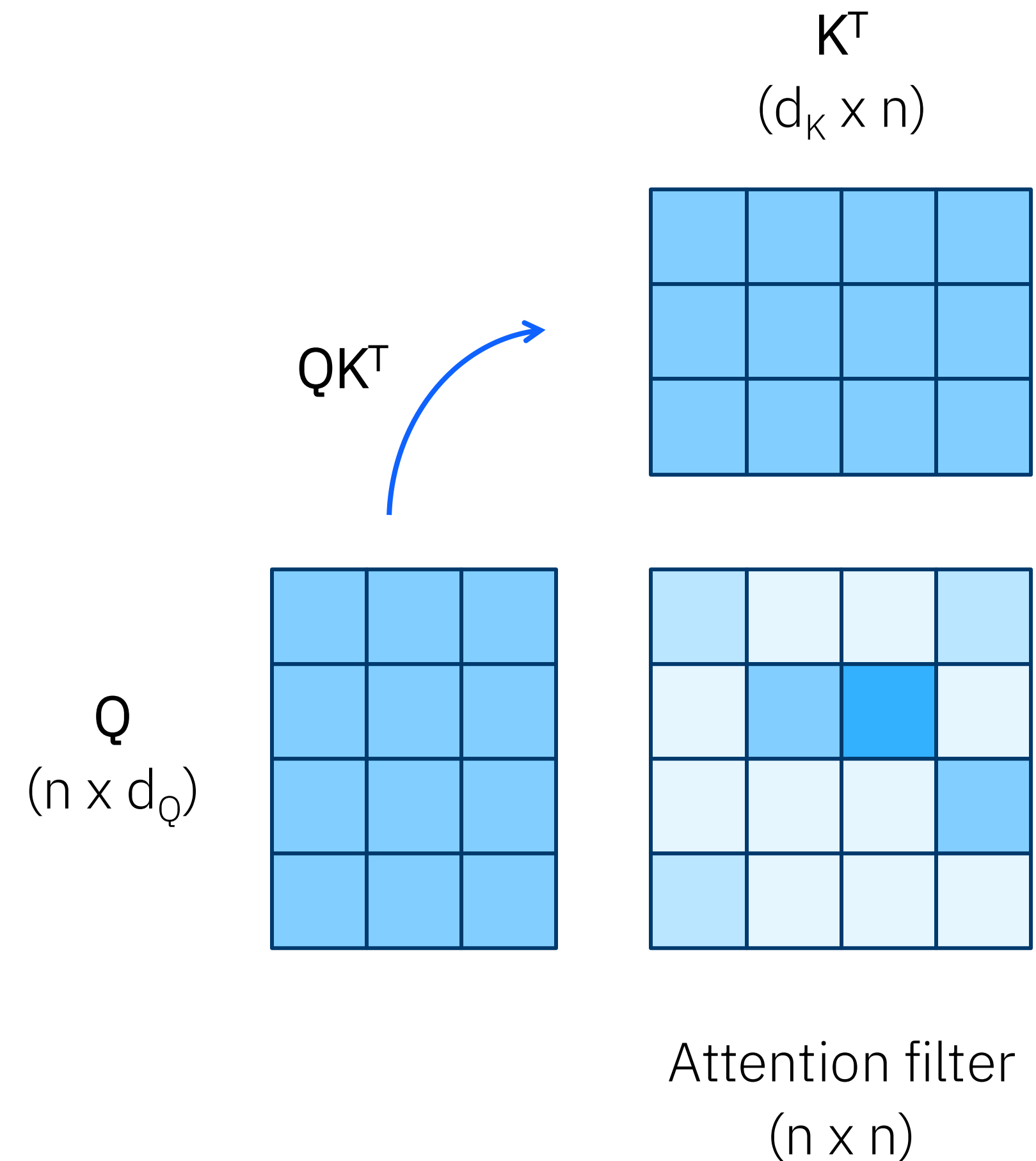
# Self-attention

Query, Key, and Value  $\rightarrow Q, K, V$

– Terminology borrowed from retrieval systems

– Steps:

1. project input into *learnable linear layers* to get  $Q, K, V$
2. compute *similarity* between  $Q$  and  $K$  and scale





# Self-attention

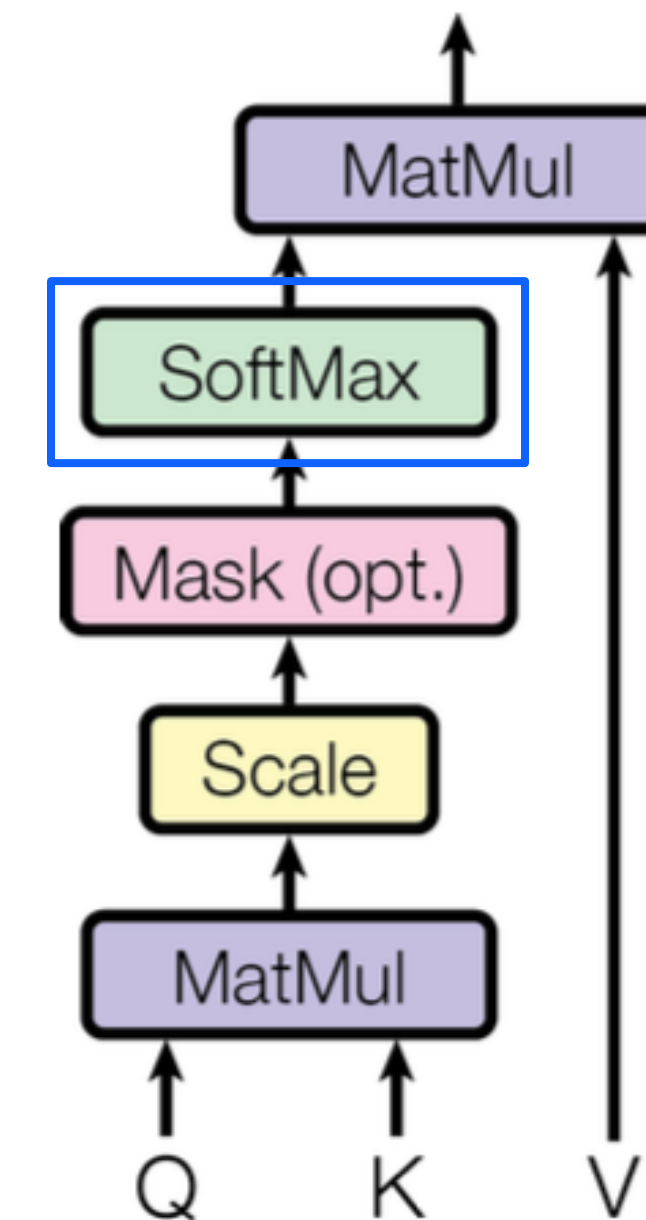
Query, Key, and Value  $\rightarrow Q, K, V$

– Terminology borrowed from retrieval systems

– Steps:

1. project input into *learnable linear layers* to get  $Q, K, V$
2. compute *similarity* between  $Q$  and  $K$  and scale
3. softmax to restrict the values between 0 and 1

Scaled dot product  
attention



Self-attention mechanism. Adapted from (1)



# Self-attention

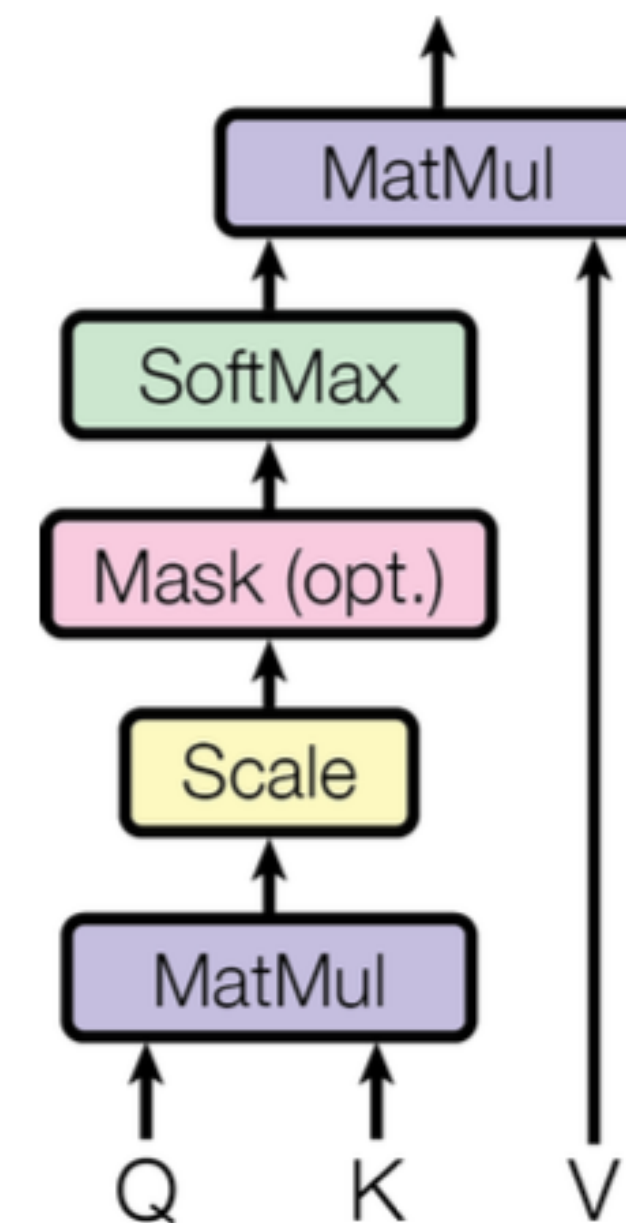
Query, Key, and Value  $\rightarrow Q, K, V$

– Terminology borrowed from retrieval systems

– Steps:

1. project input into *learnable linear layers* to get  $Q, K, V$
2. compute *similarity* between  $Q$  and  $K$  and scale
3. softmax to restrict the values between 0 and 1
4. multiply by  $V$

Scaled dot product  
attention



Self-attention mechanism. Adapted from (1)

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

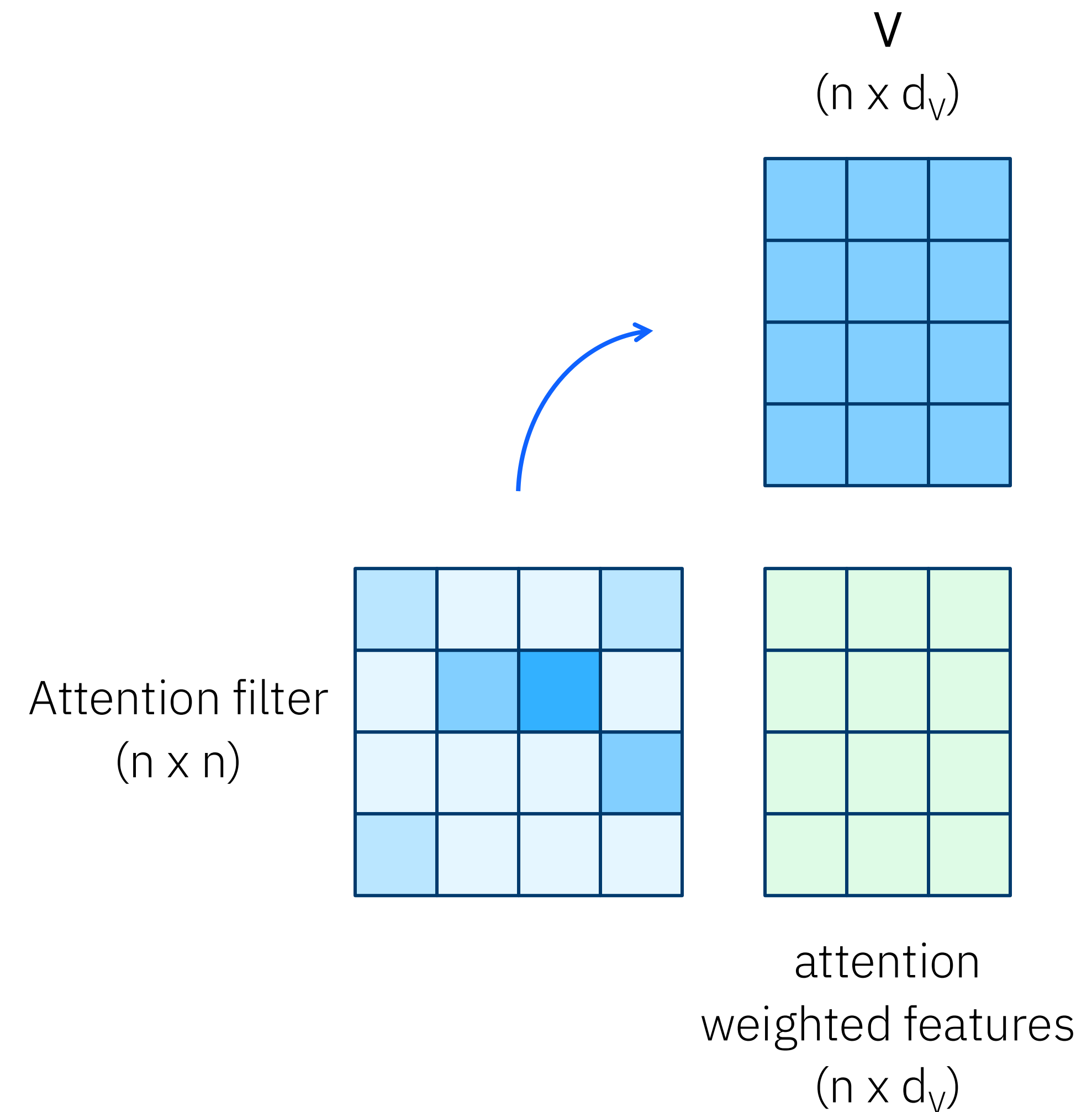
# Self-attention

Query, Key, and Value  $\rightarrow Q, K, V$

– Terminology borrowed from retrieval systems

– Steps:

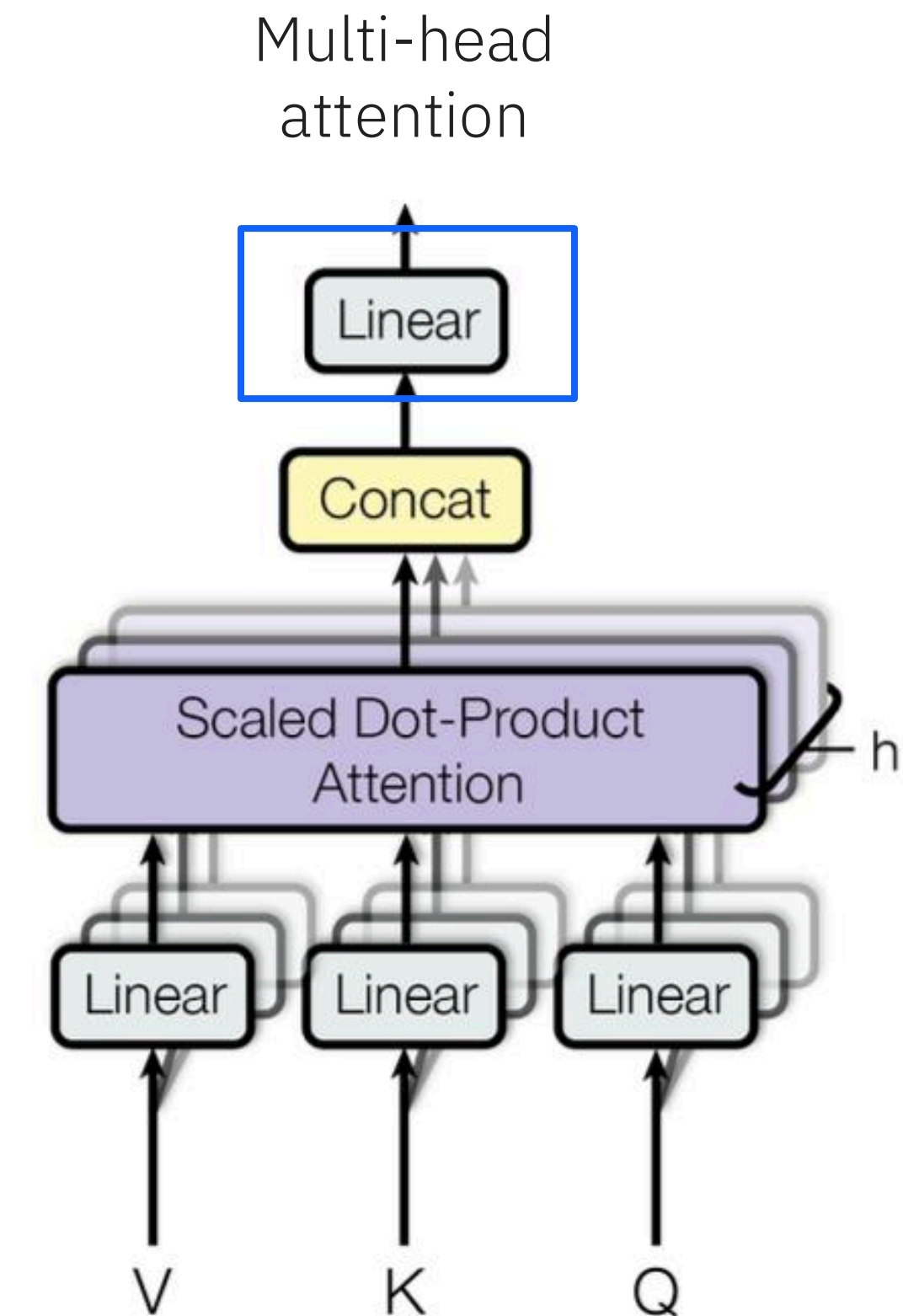
1. project input into *learnable linear layers* to get  $Q, K, V$
2. compute *similarity* between  $Q$  and  $K$  and scale
3. softmax to restrict the values between 0 and 1
4. multiply by  $V$



# Self-attention

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$
$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

- $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_Q}$
- $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_K}$
- $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_V}$
- $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$

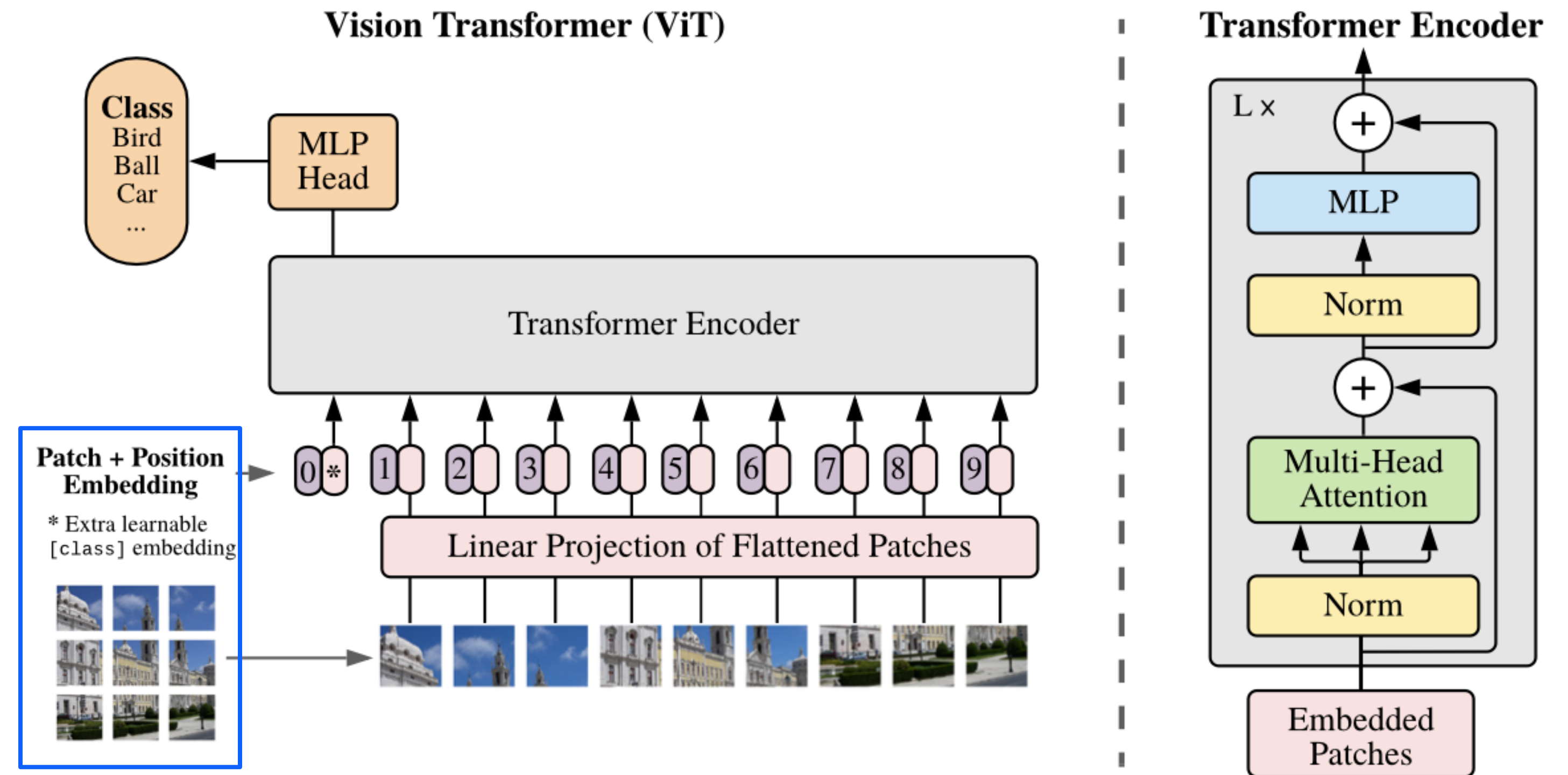


Multi-head attention mechanism. Adapted from (1)

# Vision transformer (ViT)

## Patch embeddings

- Divide image into smaller *patches* and *project* into an embedding space.
- patch size =  $16 \times 16$
- Image  $\rightarrow$  *tokens*
- Position embeddings: information about the *position* of tokens in the input image.



Vision transformer (ViT) architecture. Extracted from (1)

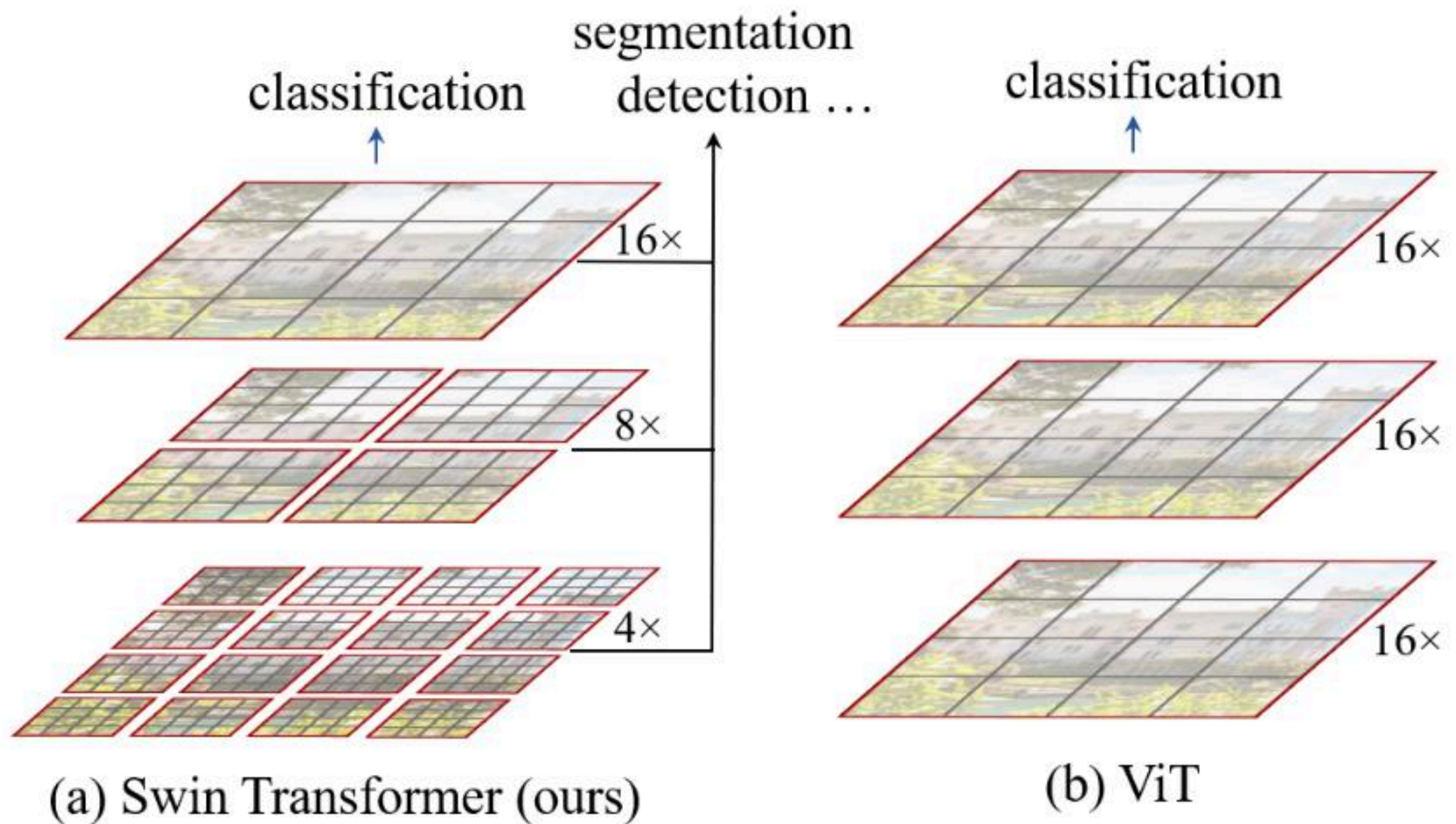


# Swin transformer

## –CNNs x ViT

- Images have different *scales*, unlike text → ViT tokens have fixed size.
- Pixels in images have much *higher resolution* than words in text → dense tasks require accuracy at the *pixel level*.

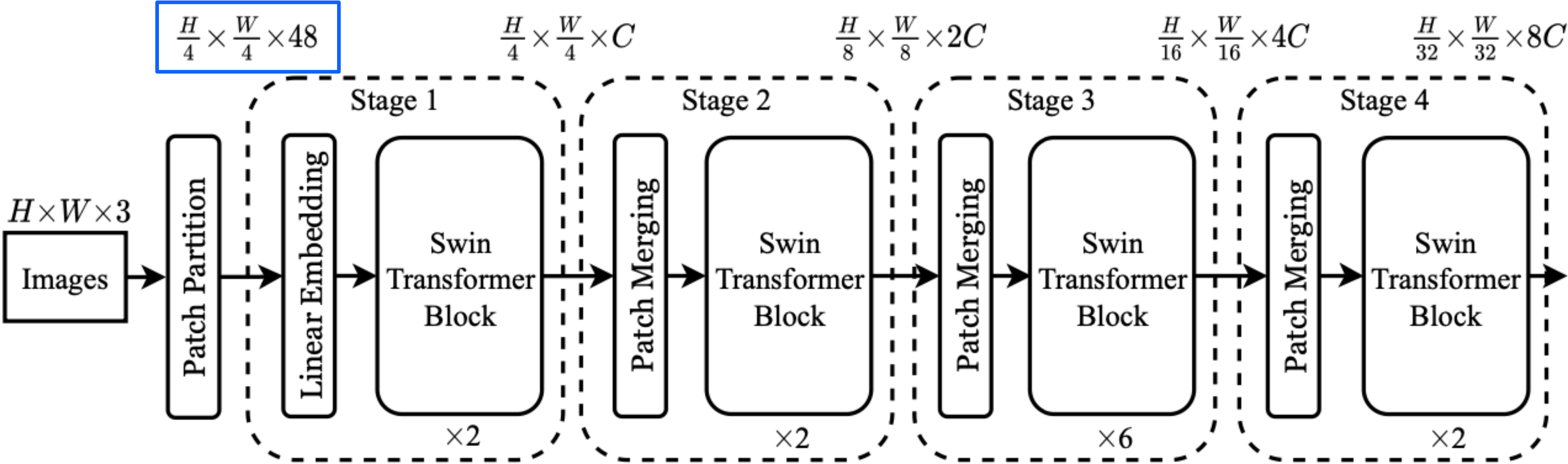
## – *Hierarchical* transformer architecture → dense predictions



Extracted from (1)

# Swin transformer

–Smaller patch size (4 x 4)



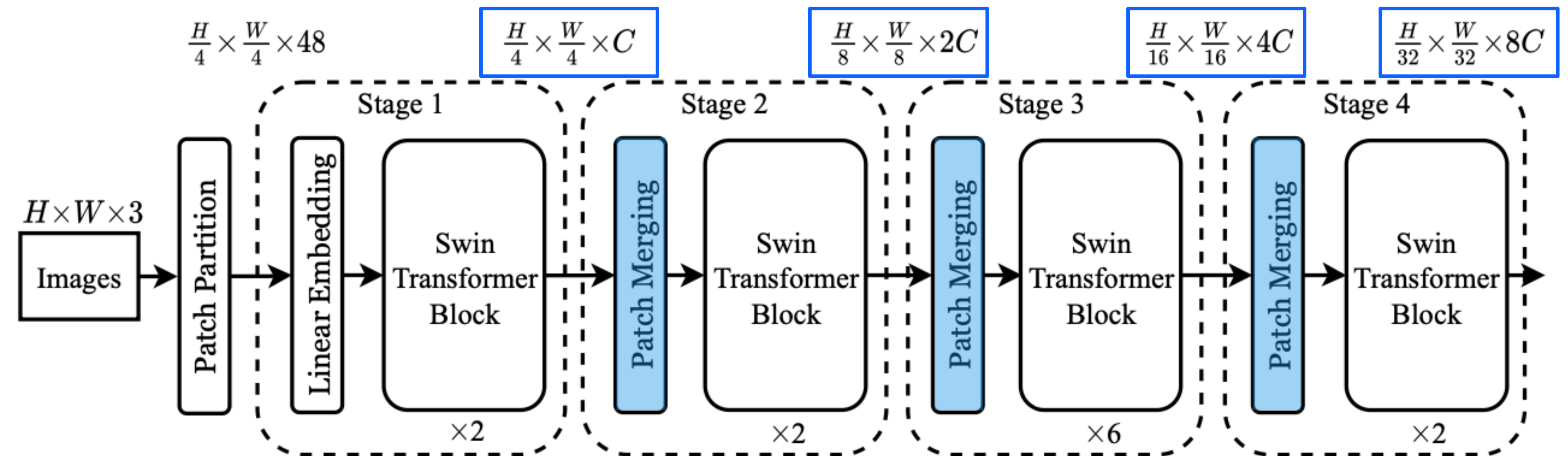
Extracted from (1)

# Swin transformer

–Smaller patch size (4 x 4)

- Start from small patches and *gradually merge* nearby patches in deeper layers.

–With the *hierarchical feature maps*, the Swin Transformer can leverage techniques for *dense prediction* such as FPN or U-Net.



Extracted from (1)

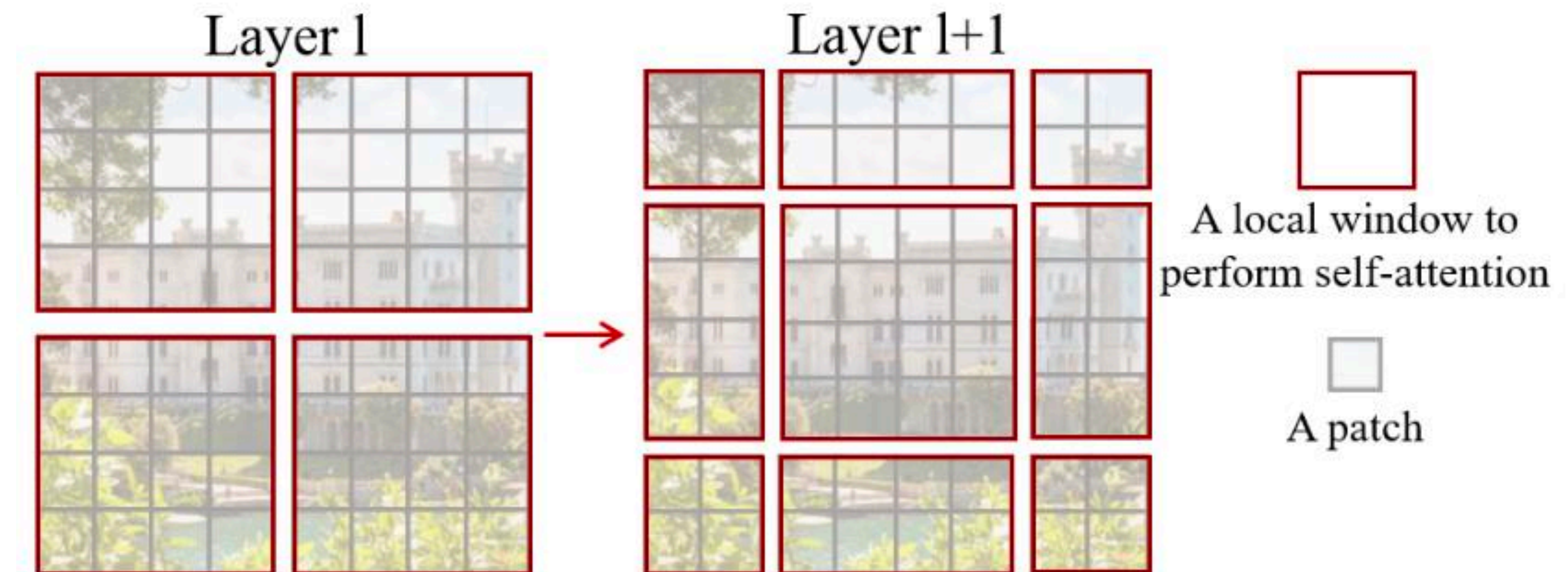


# Swin transformer

Global attention x *local window* attention

–Linear computational complexity

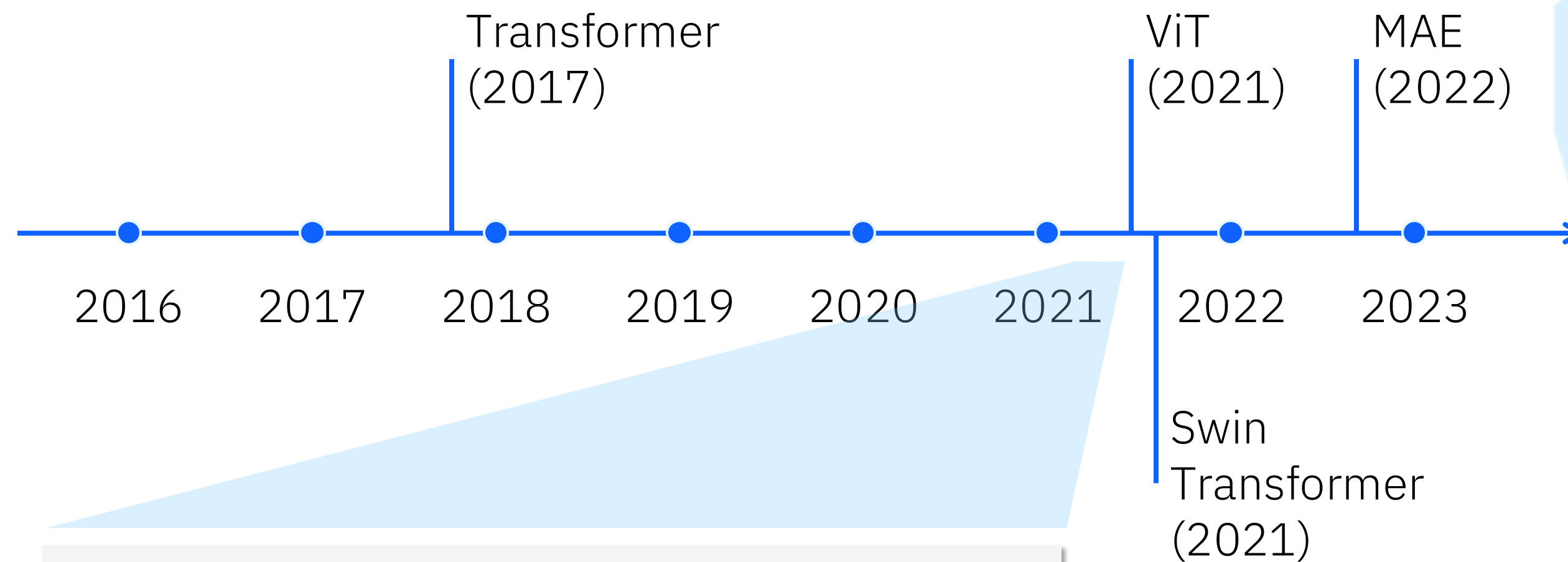
- Compute self-attention locally within non-overlapping *windows*.
- The number of patches in each window is *fixed*: complexity is linear to image size.



Extracted from (1)



# Masked AutoEncoder: ViT + self-supervised learning



## Supervised training

Swin → direct:

– classification, segmentation, object detection

ViT → with *pre-training* phase:

– ImageNet → 1k classes, 1.3M images

– ImageNet-21k → 21k classes, 14M images

– JFT → 18k classes, 303M images

## Masked Autoencoders Are Scalable Vision Learners

Kaiming He<sup>\*,†</sup> Xinlei Chen<sup>\*</sup> Saining Xie Yanghao Li Piotr Dollár Ross Girshick

<sup>\*</sup>equal technical contribution <sup>†</sup>project lead

Facebook AI Research (FAIR)

*“With MAE pre-training, we can train data-hungry models like ViT-Large/-Huge on ImageNet-1K with improved generalization performance. With a vanilla ViT-Huge model, we achieve 87.8% accuracy when fine-tuned on ImageNet-1K. This outperforms all previous results that use only ImageNet-1K data”.<sup>1</sup>*

# Self-supervised learning

Training ML models for classification, regression, segmentation, etc., requires comparing the model's output for a given input to a *ground truth*.

Supervised learning → requires *labeled* data

- Manual annotations
- High costs, time-consuming

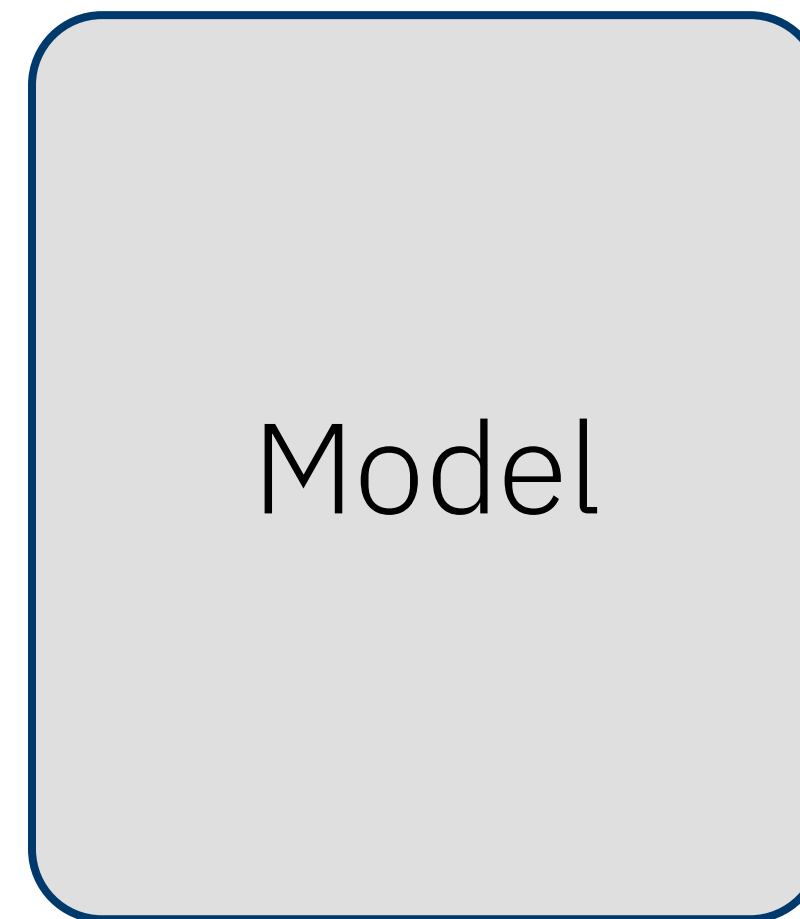
Self-supervised learning (SSL) → implicit labels

- Labels can be *inferred* from *unlabeled* data
- Models are pre-trained on a *pretext task* – learn *representations* – and fine-tuned on downstream tasks.
- Fine-tune on specific *labeled* dataset

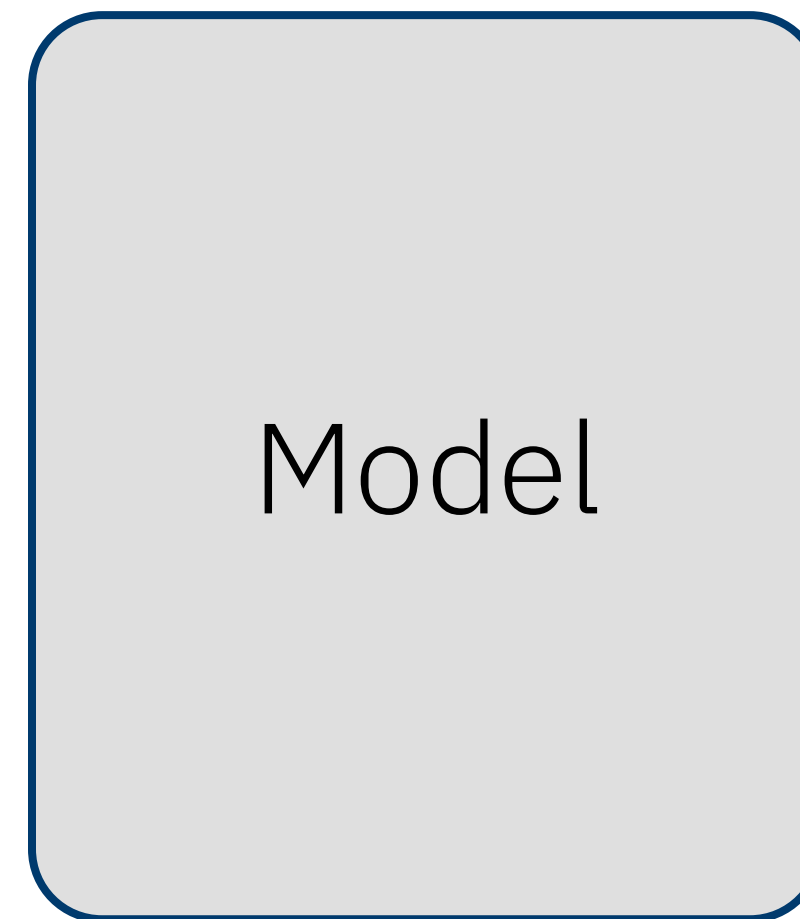
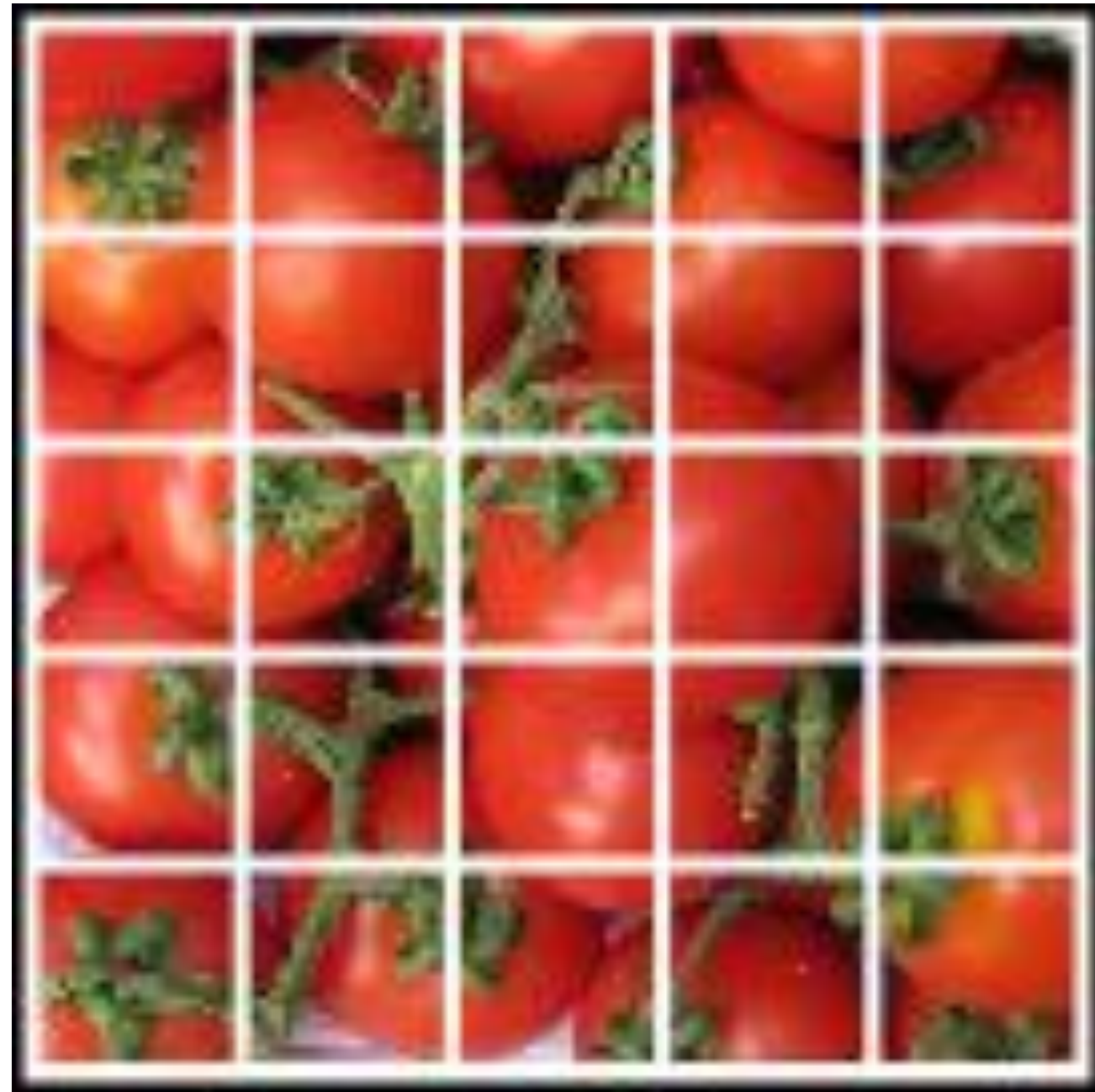
Techniques – pretext tasks

- Masked Image Modeling (MIM)
  - Predict or *reconstruct* masked parts of the image
  - Generative
- Contrastive learning
  - Given pairs of data samples, *distinguish* between similar (positive) and dissimilar (negative) pairs.
  - Positive pairs usually obtained through *data augmentation* techniques.
  - Discriminative

# Masked image modeling

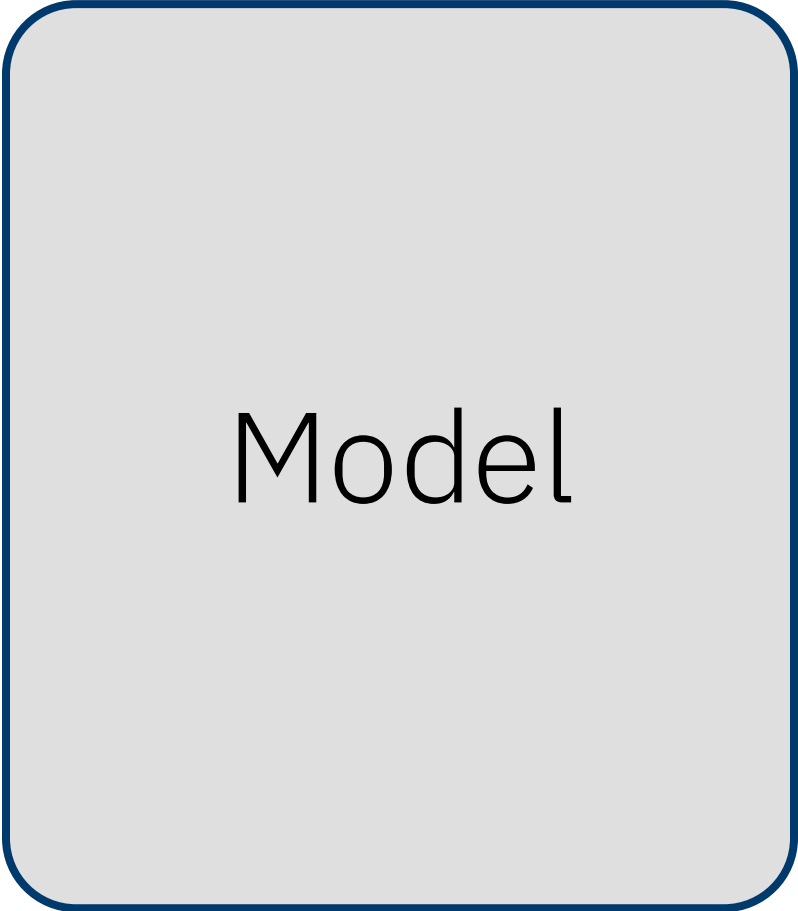


# Masked image modeling





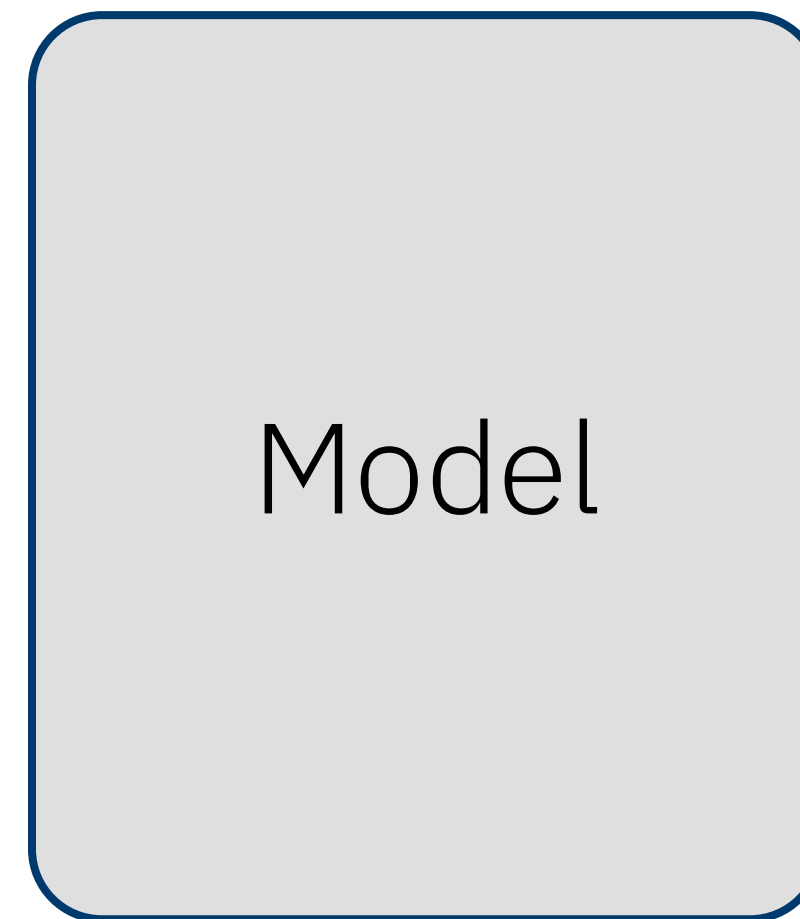
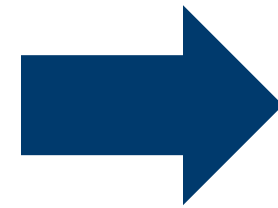
# Masked image modeling



# Masked image modeling



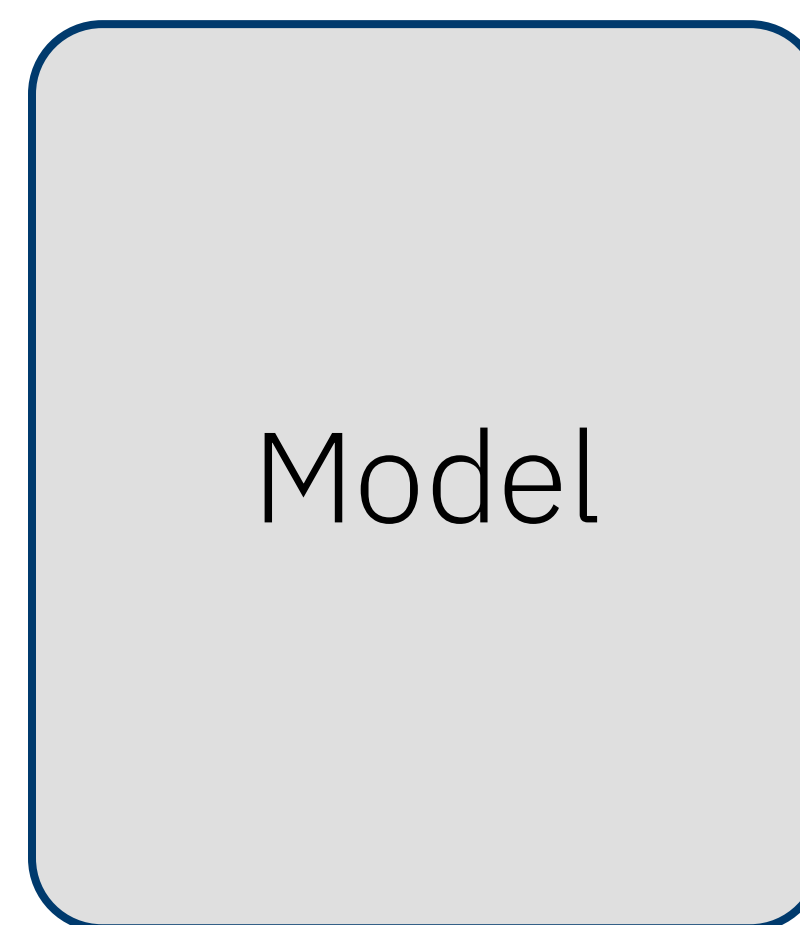
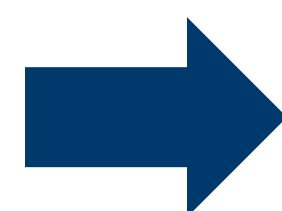
masked input



# Masked image modeling



masked input



reconstructed image

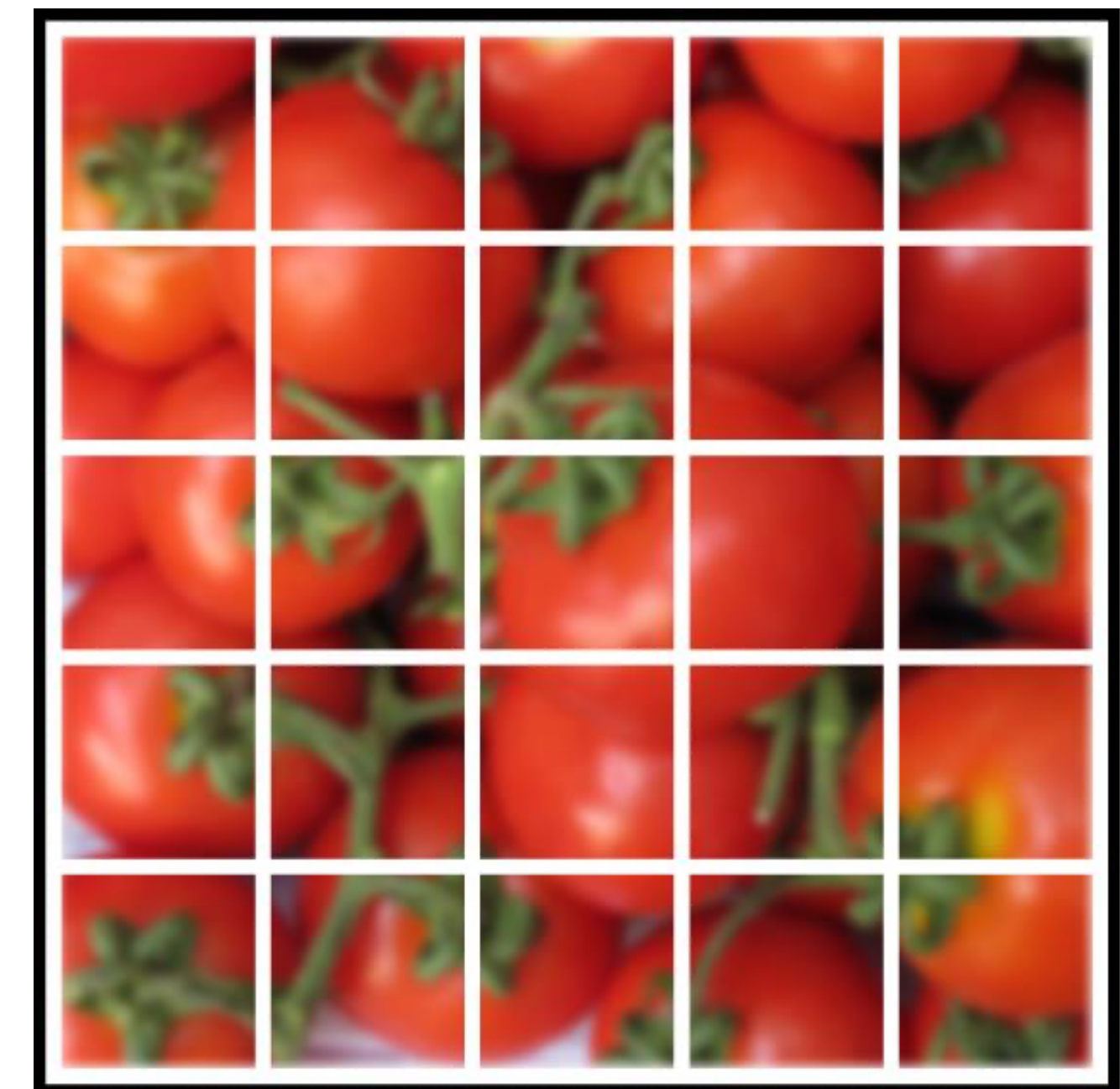


# Masked image modeling



target = original image

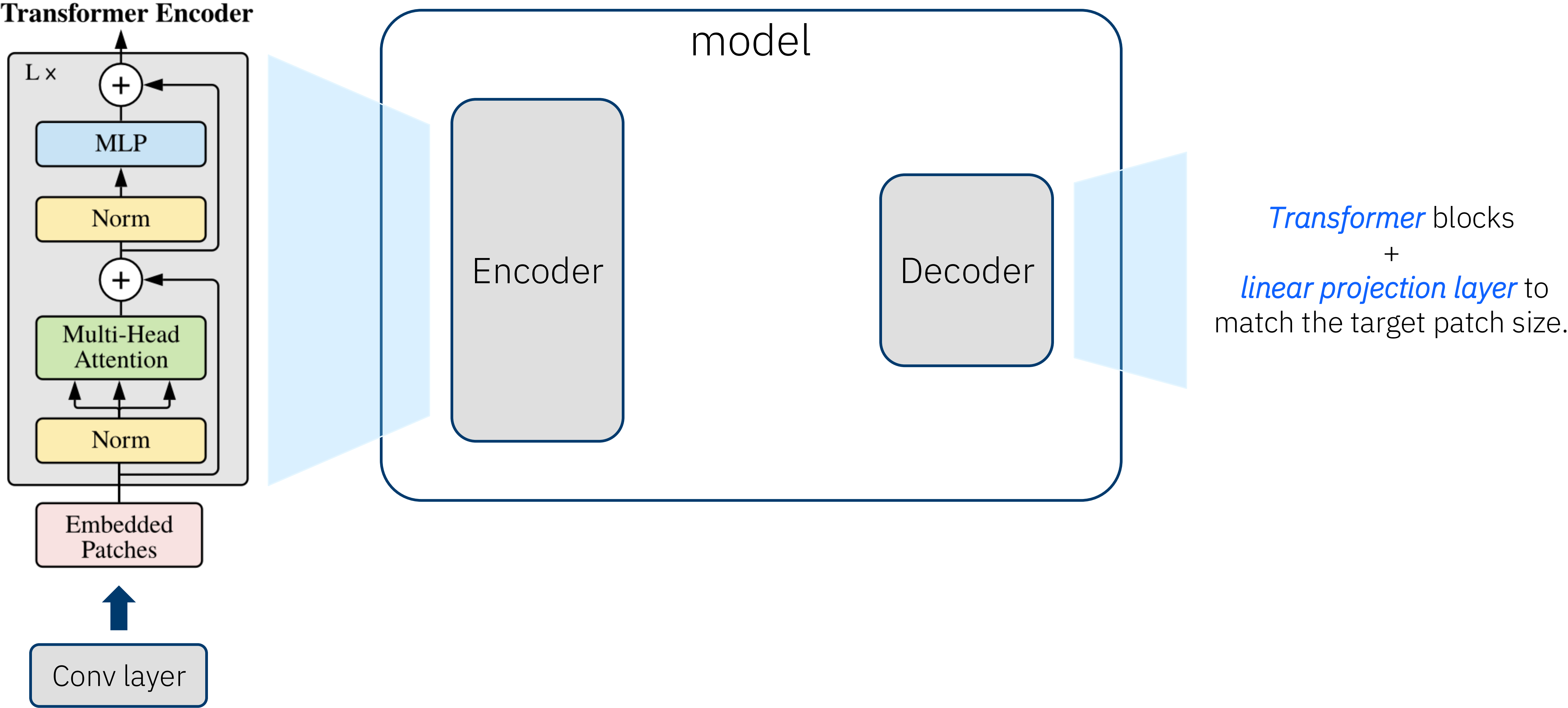
MSE loss



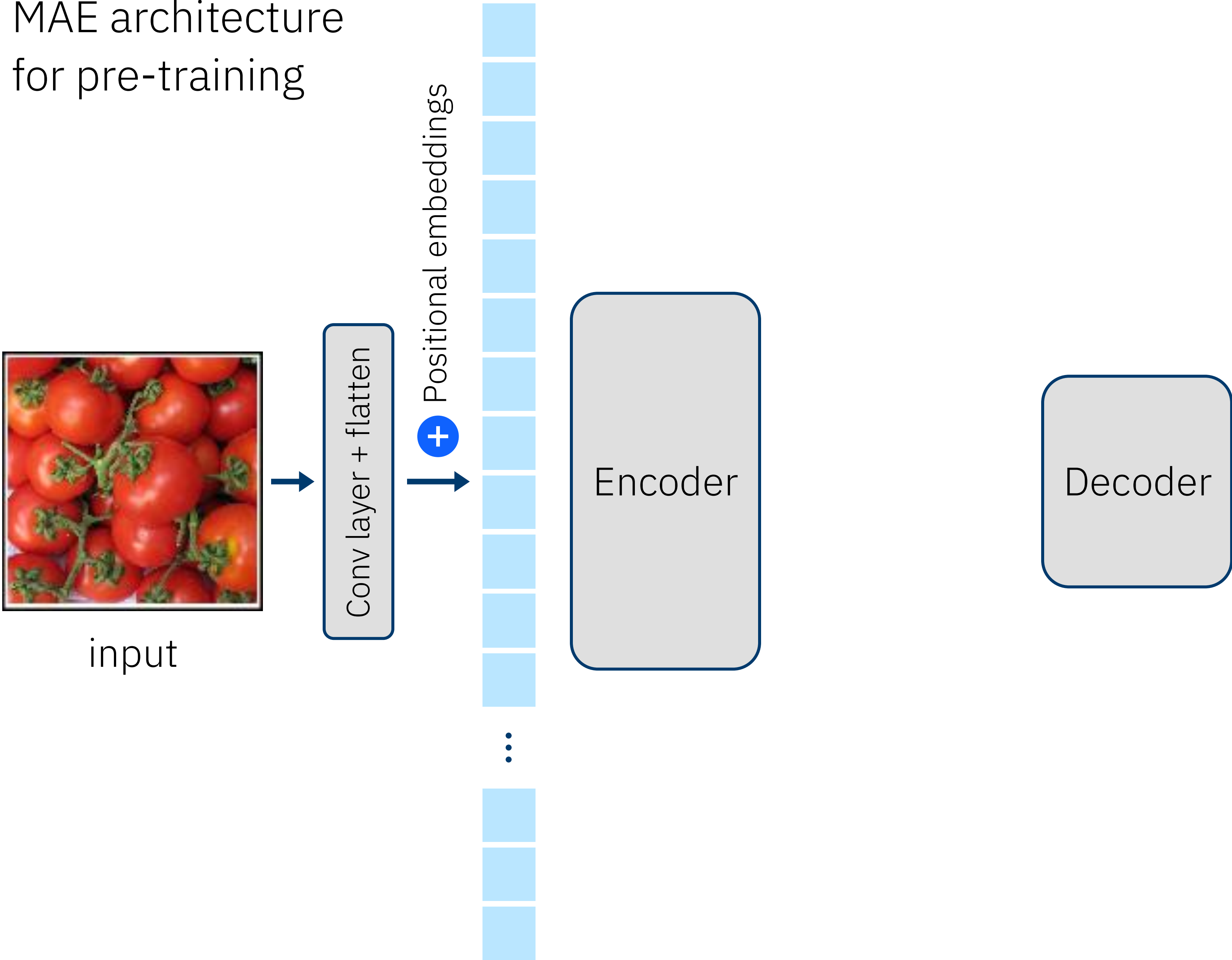
reconstructed image



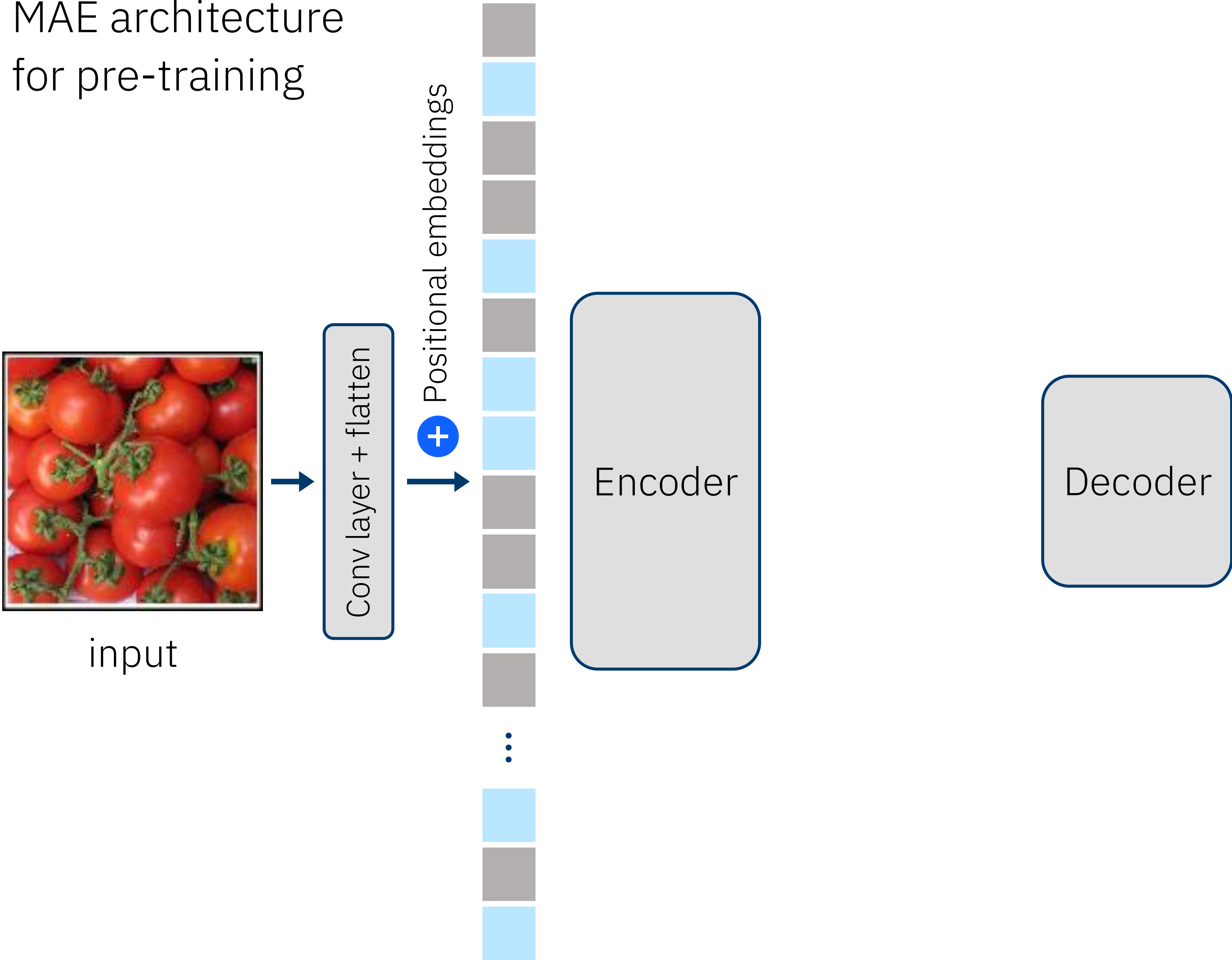
MAE architecture  
for pre-training



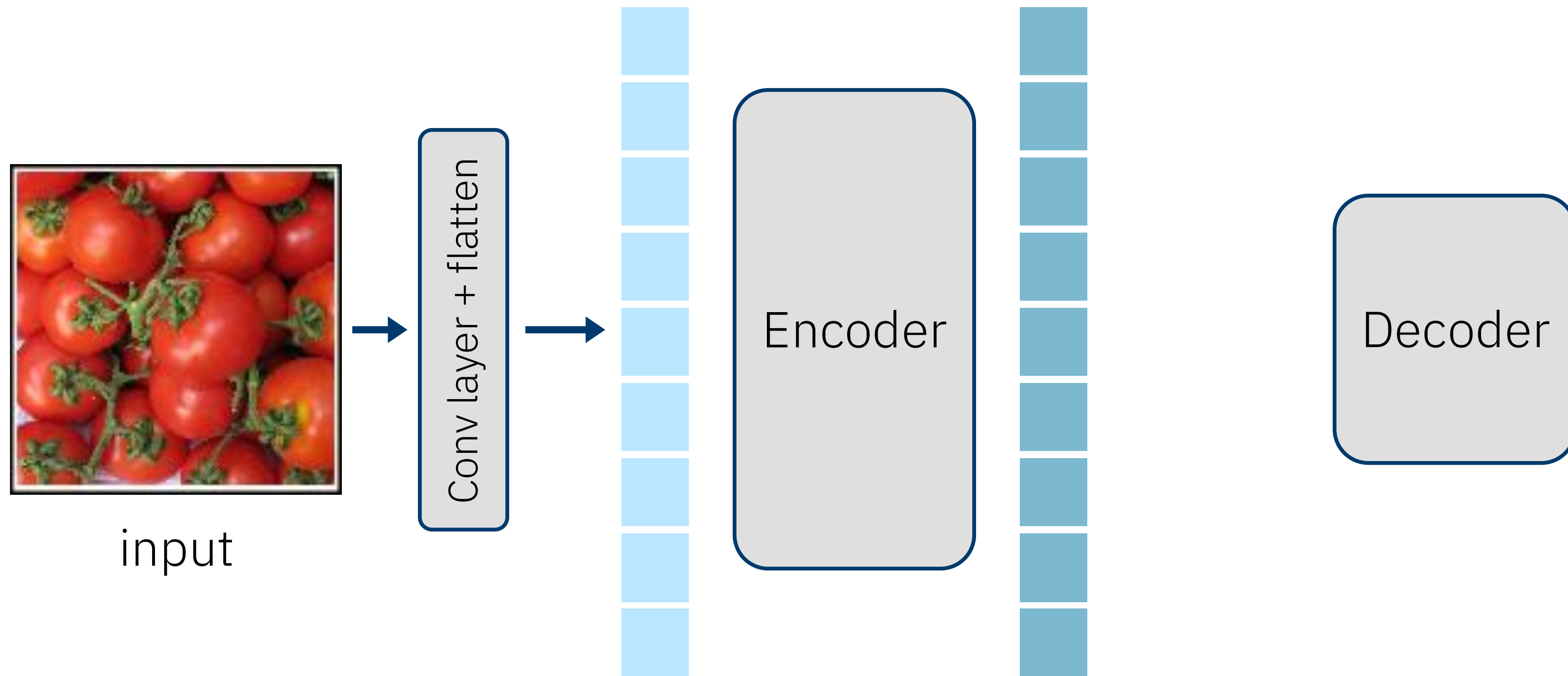
MAE architecture  
for pre-training



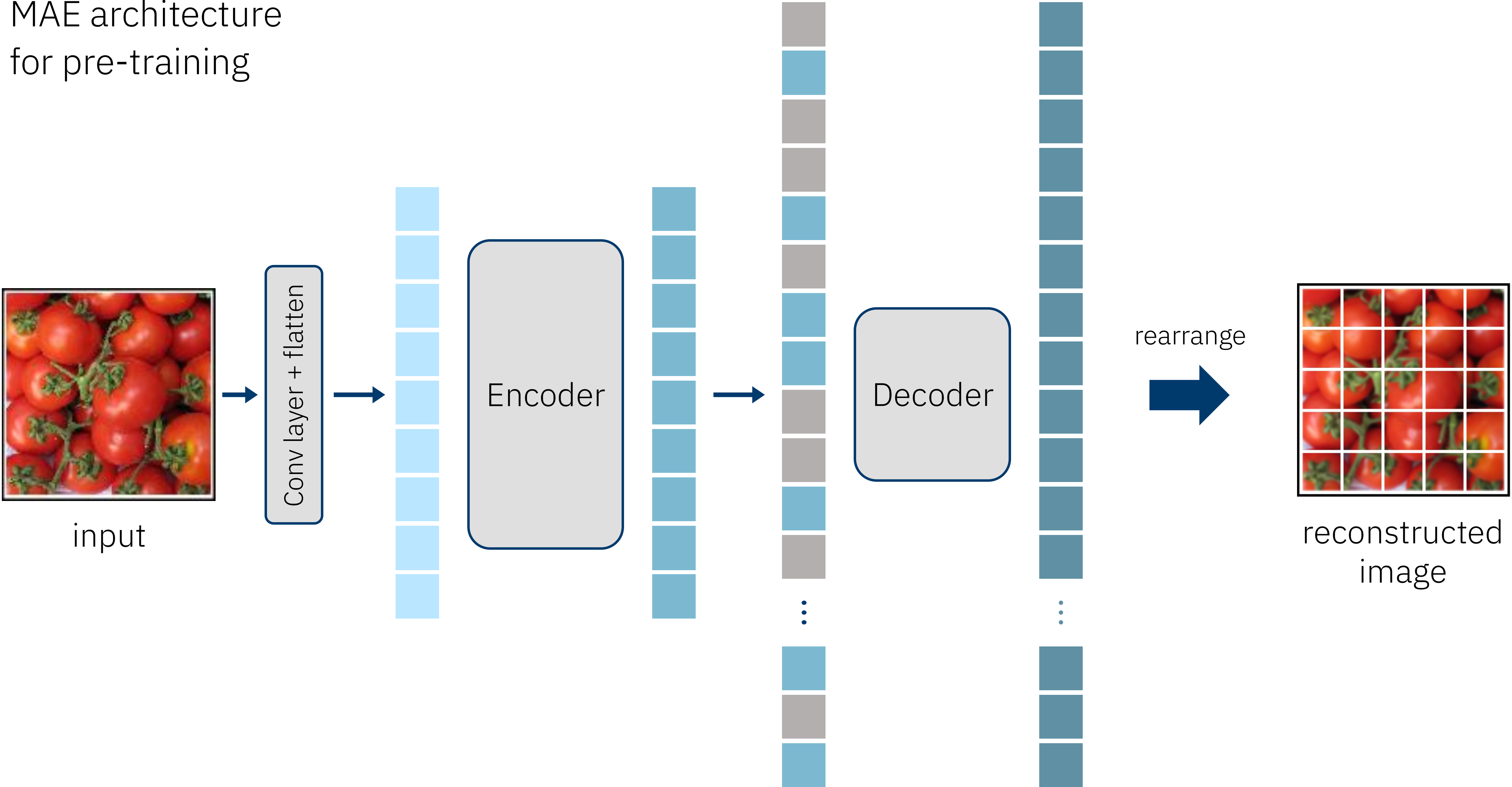
MAE architecture  
for pre-training



# MAE architecture for pre-training



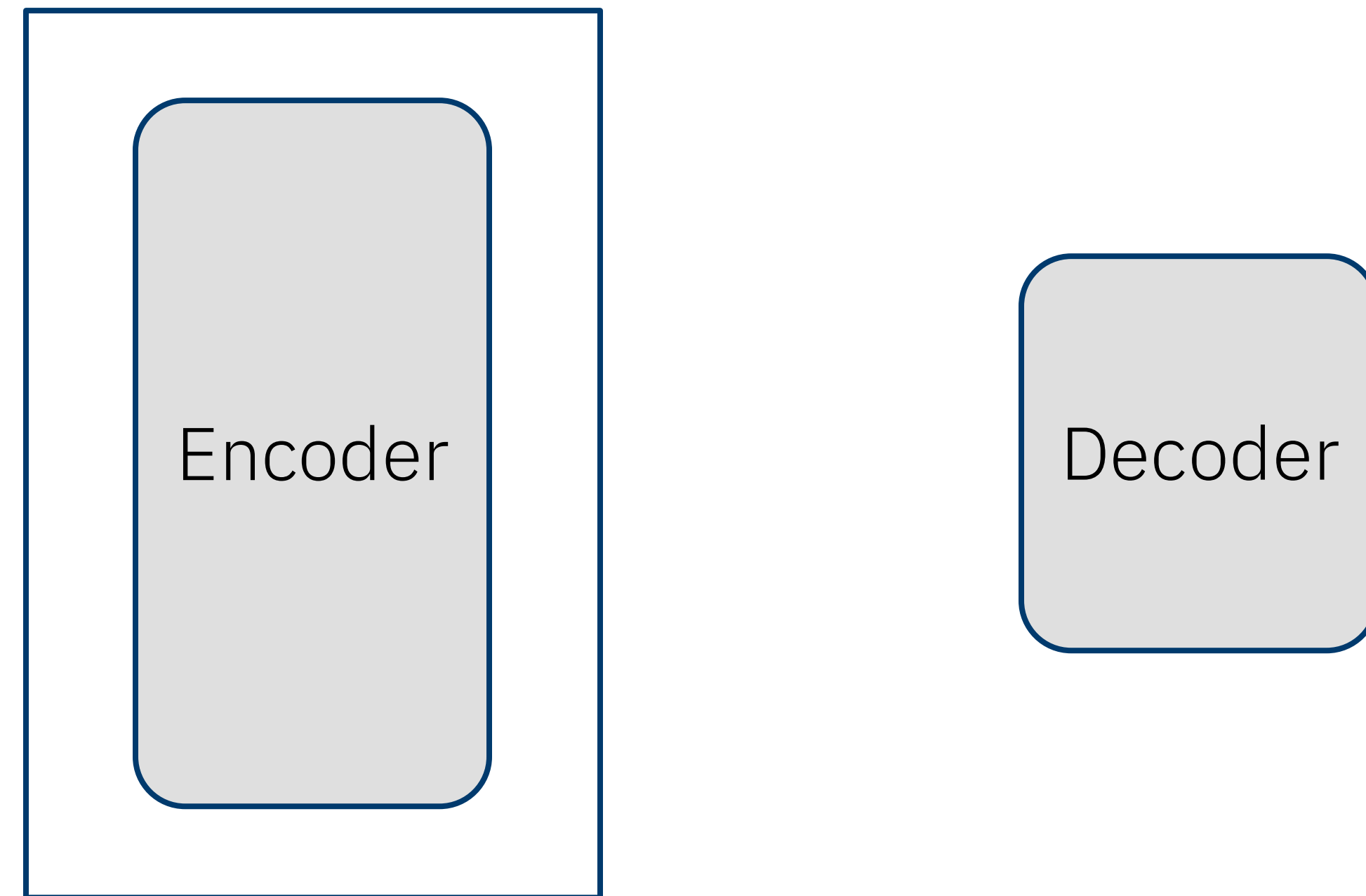
MAE architecture  
for pre-training





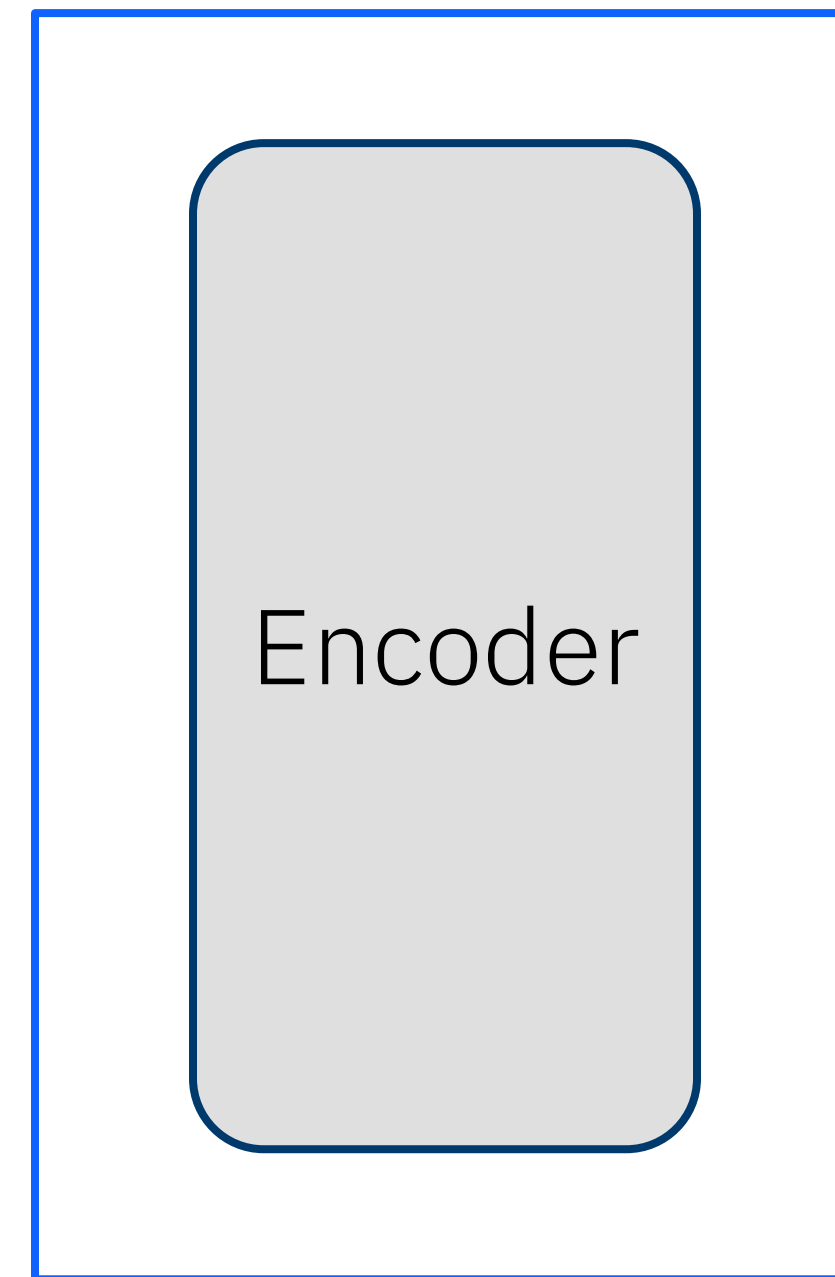
# MAE architecture for fine-tuning

Pre-trained model



# MAE architecture for fine-tuning

Pre-trained model



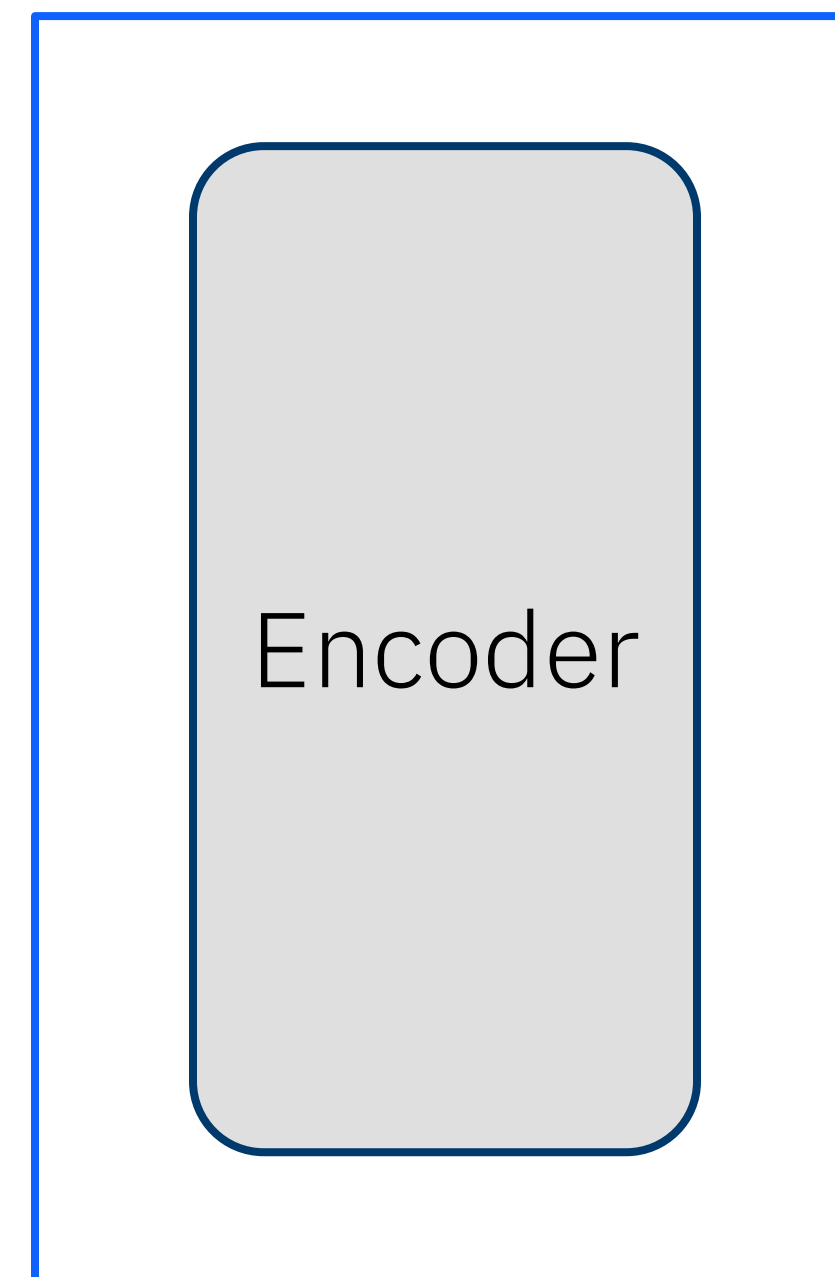
classification  
layers

⋮

segmentation  
layers

# MAE architecture for fine-tuning

## Pre-trained model



### Classification layers:

- 1 fully connected (FC) layer
- multiple FC layers
- avg pooling + FC layer...

### Segmentation layers:

- U-Net
- UperNet
- FCN...

# Pre-training x fine-tuning

## Pre-training

- Large/huge *unlabeled* dataset
- Goal: Learn *representations* – characteristics of the data.
- Pre-text task defines pre-training architecture.
- Long training sessions
- Requires *significant GPU resources*
- Requires *distributed training techniques* – efficient data-loading, GPU communication, ...

## Fine-tuning

- Smaller *labeled* dataset (task-specific)
- Goal: learn *specific task*.
- Downstream task defines architecture requirements.
- Shorter training sessions
- Can usually be done with *a single GPU*.
- Parameter *efficient fine-tuning methods* can improve training efficiency on time and resources.



# Fine-tuning

## Frozen encoder

encoder weights (pre-trained)  
are frozen → only decoder is  
trained

## Full fine-tuning

All model parameters are  
trained → encoder + decoder



+GPU resources  
+Memory requirements  
+Accuracy  
...

# Fine-tuning

## Frozen encoder

encoder weights (pre-trained)  
are frozen → only decoder is  
trained

## PEFT

Techniques that fine-tune a *small number* of (extra) encoder parameters → significantly *reduces memory usage* while keeping *comparable performance* to full fine-tuned model.

## Full fine-tuning

All model parameters are  
trained → encoder + decoder

+GPU resources  
+Memory requirements  
+Accuracy  
...

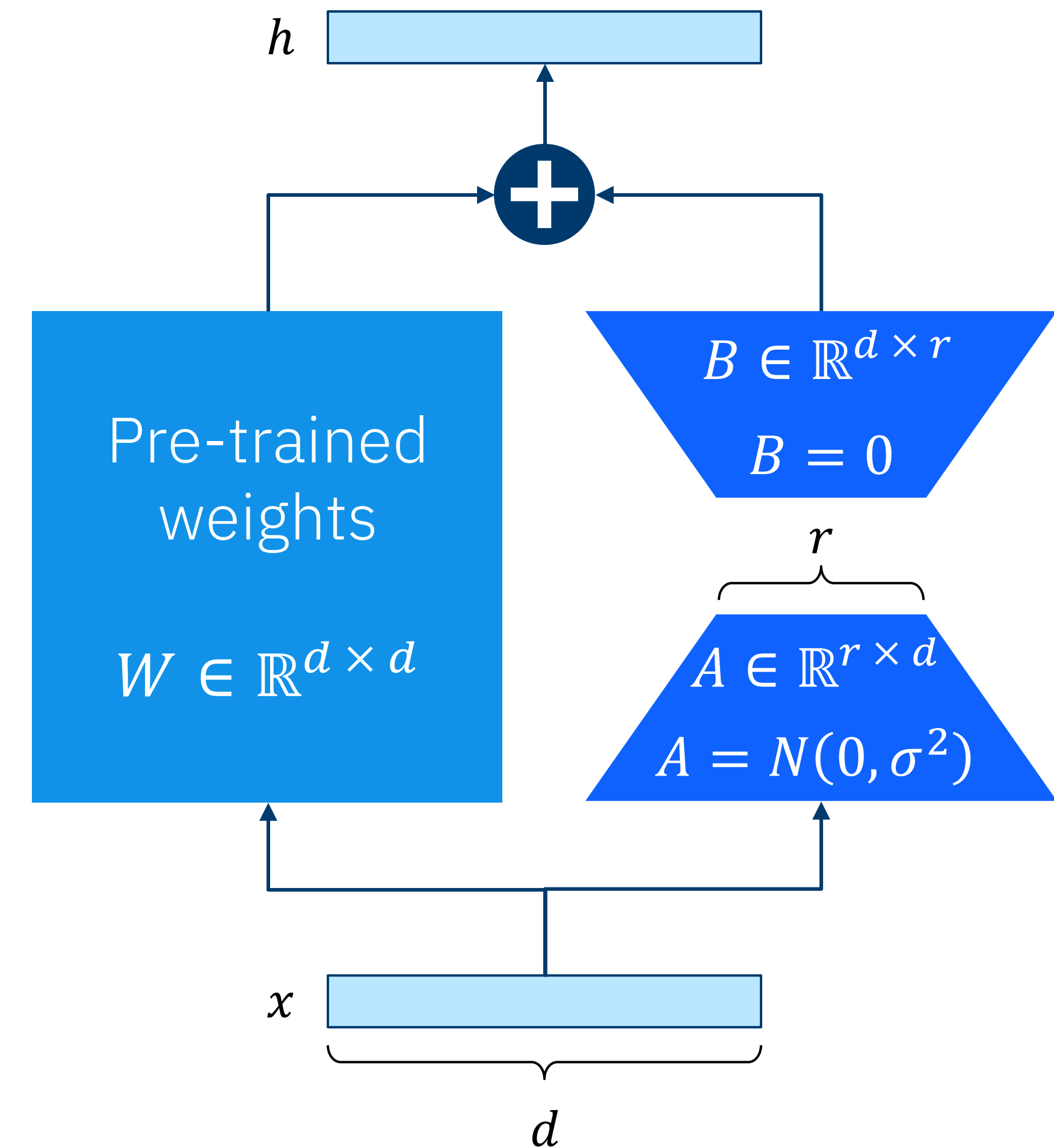
# Low Rank Adaptation - LoRA

LoRA represents the weight updates  $\Delta W$  with two smaller matrices, called *update matrices*, through low-rank decomposition.

These new matrices are trained to adapt to the new downstream task, while the original weight matrix remains *frozen* and doesn't receive any further updates.

With the original weights frozen, one can have multiple lightweight, portable LoRA models for multiple downstream tasks built on top of them.

This approach allows to keep the overall number of trainable parameters low, while performance of LoRA models is comparable to the performance of fully fine-tuned models.



Adapted from (1)

# Geospatial Foundation Models (GeoFMs)





IBM Newsroom


News ▾Media resources ▾Inside IBM ▾Blog ▾

# IBM and NASA Open Source Largest Geospatial AI Foundation Model on Hugging Face

Effort aims to widen access to NASA earth science data for geospatial intelligence and accelerate climate-related discoveries

Aug 3, 2023

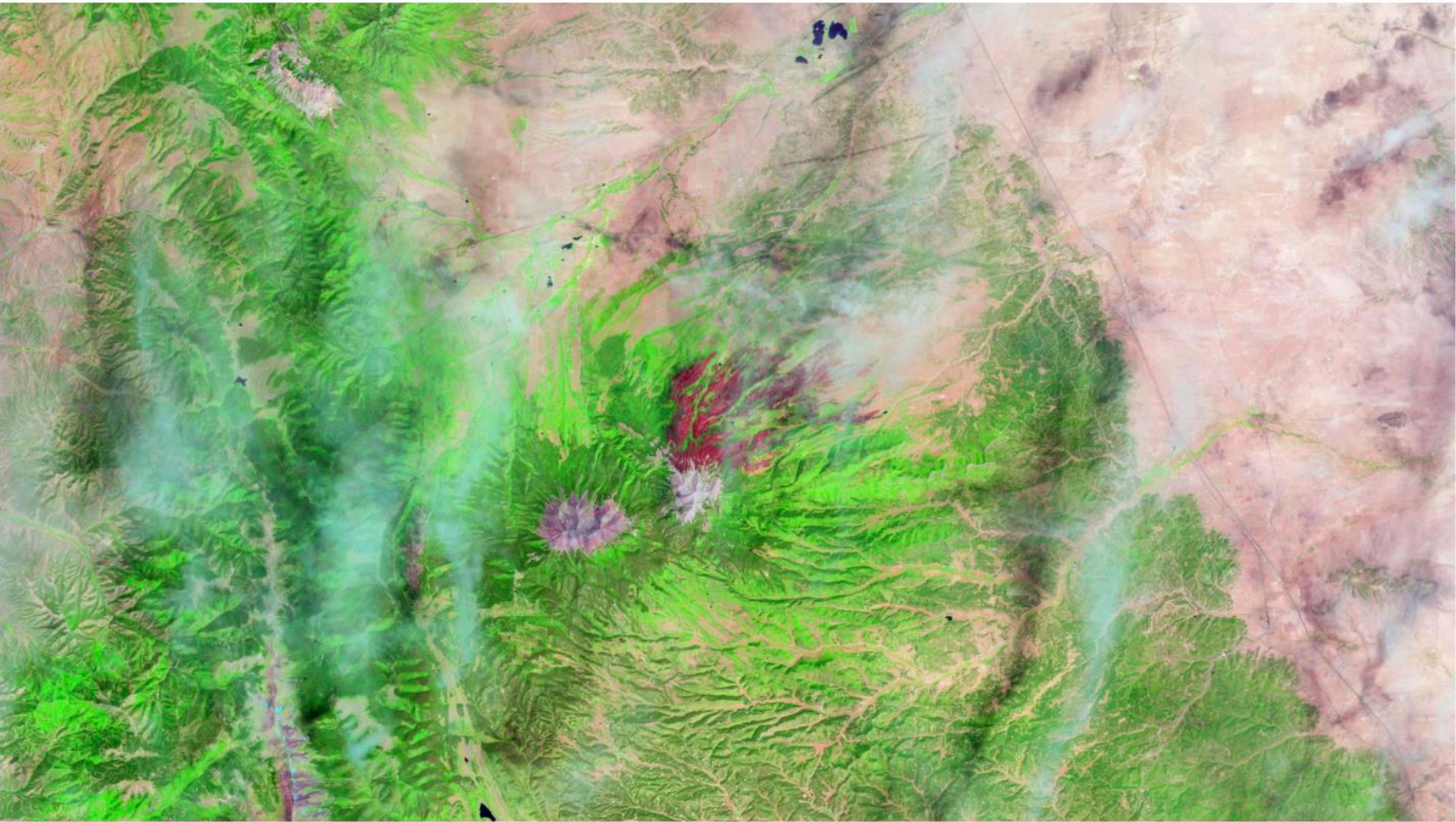




News & Events ▾

4 MIN READ

## Expanded AI Model with Global Data Enhances Earth Science Applications



On June 22, 2013, the Operational Land Imager (OLI) on Landsat 8 captured this false-color image of the East Peak fire burning in southern Colorado near Trinidad. Burned areas appear dark red, while actively burning areas look orange. Dark green areas are forests; light green areas are grasslands. Data from Landsat 8 were used to train the Prithvi artificial intelligence model, which can help detect burn scars.

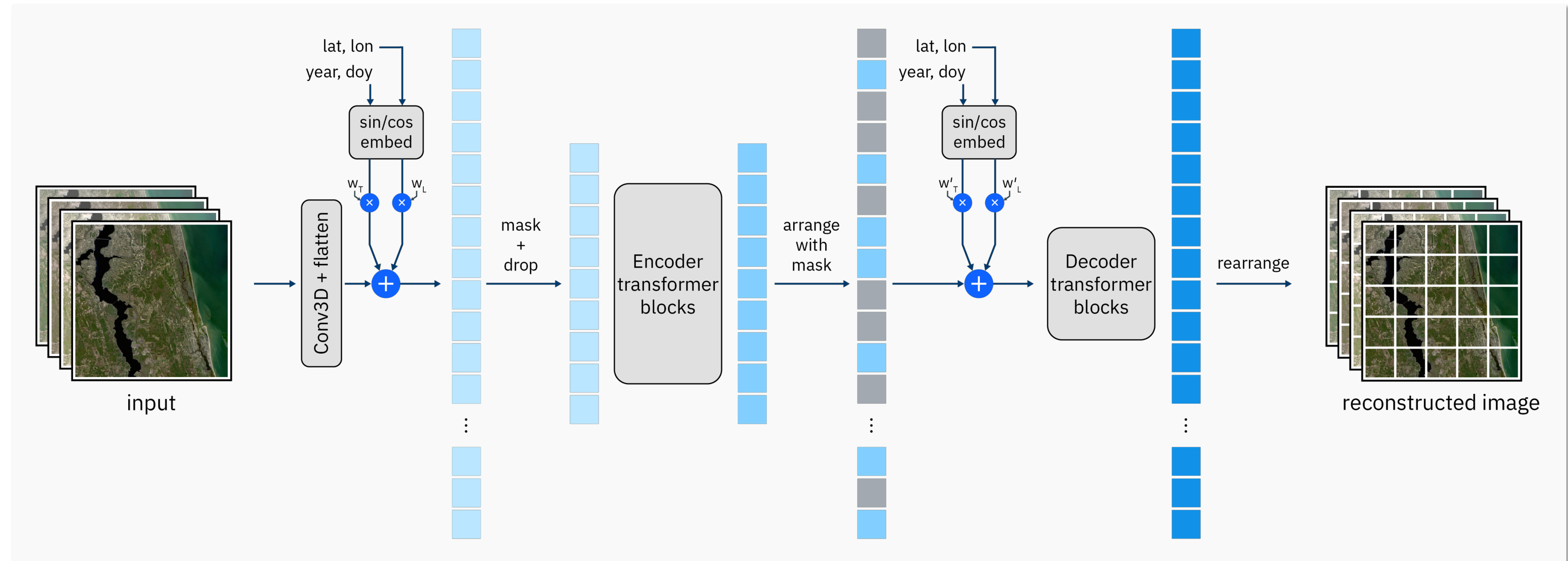
NASA Earth Observatory



# Prithvi-EO-2.0

## Key features:

- Trained on *4.2M global samples* from NASA's HLS
- 300M and 600M parameter versions
- Temporal and location information added to the models
- Extensive benchmarking with GEO-Bench
- SME feedback on model and dataset design





# Data: Harmonized Landsat Sentinel-2

The Harmonized Landsat Sentinel-2 ([HLS](#)) provides consistent global observations of the land.

- Data from NASA/USGS's [Landsat 8](#) and [9](#) and the ESA's [Sentinel-2A](#) and [Sentinel-2B](#) satellites
- Data available in [tiles](#): each has [3660 x 3660 pixels](#) (~110 x 110 km).
- 30m resolution
- We used bands present in both Landsat and Sentinel: [Blue](#), [Green](#), [Red](#), [NIR](#), [SWIR 1](#), and [SWIR 2](#).

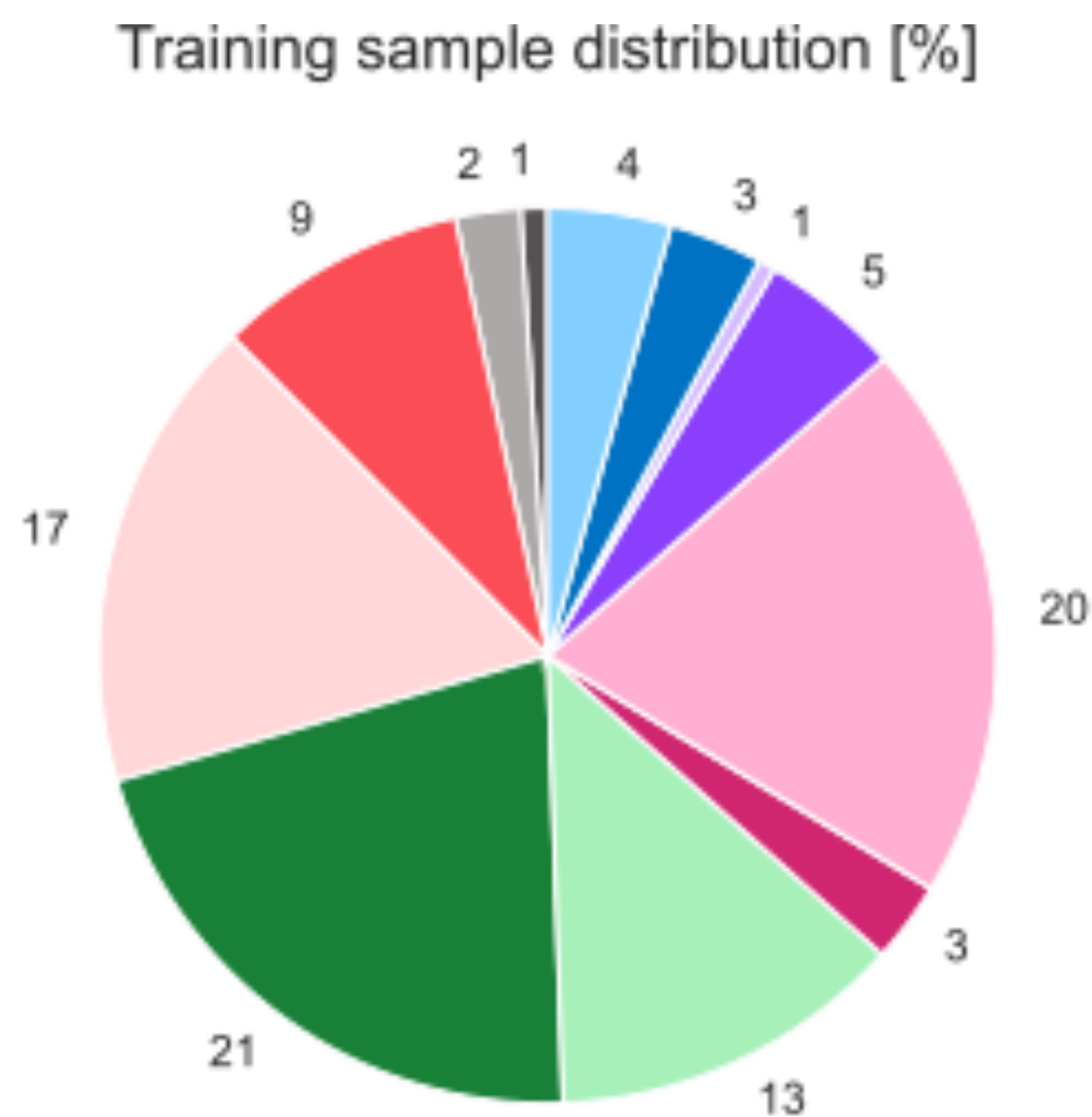
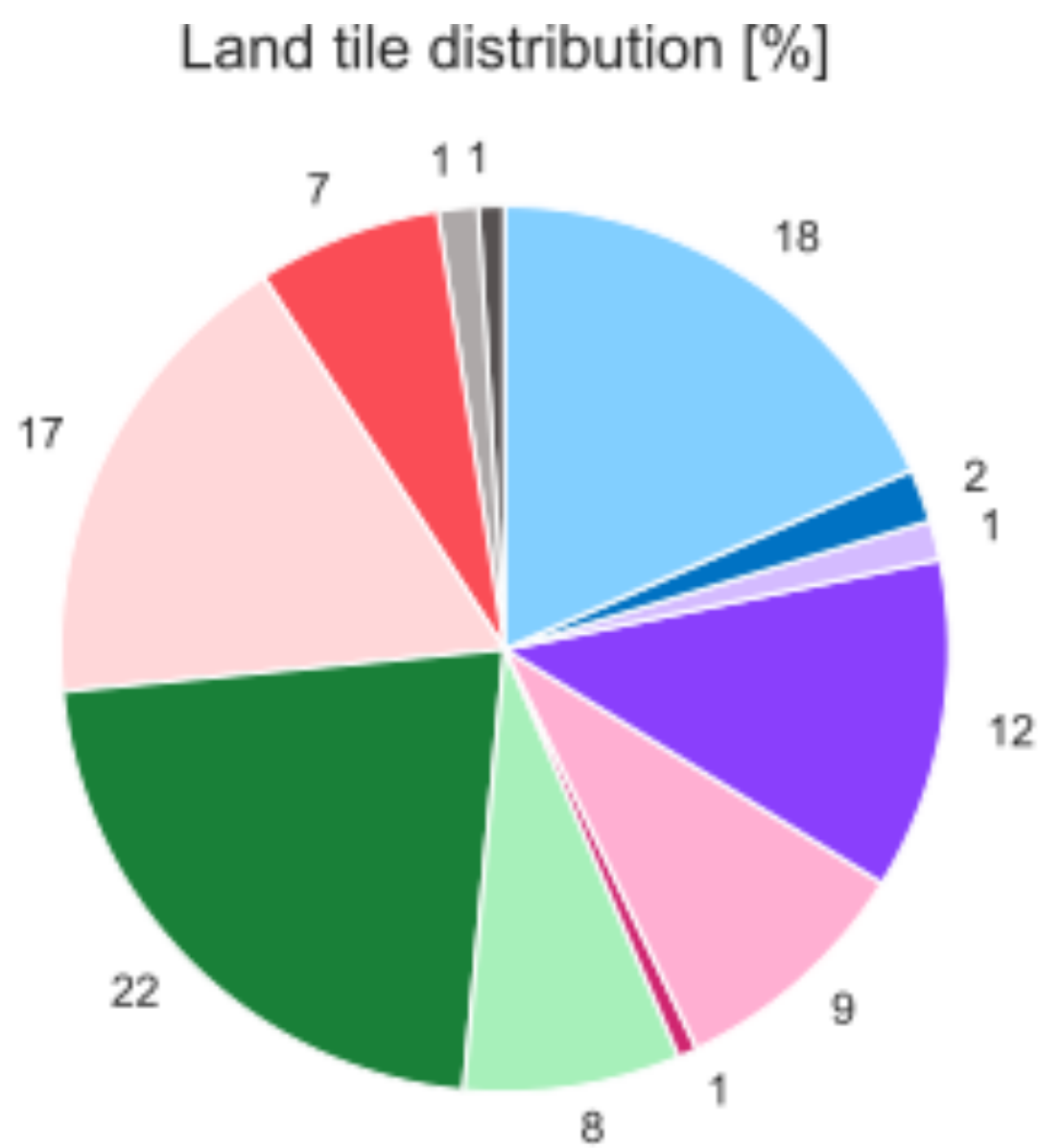




# Data: Sampling

Goal: obtain samples representing *diverse* land use and ecosystems while *minimizing cloud* and *missing value* issues.

- 1. Compute proportion of land use/land cover (LULC) classes and ecoregions for each HLS tile.
- 2. Sample tiles based on the LULC classes – urban areas are upsampled.

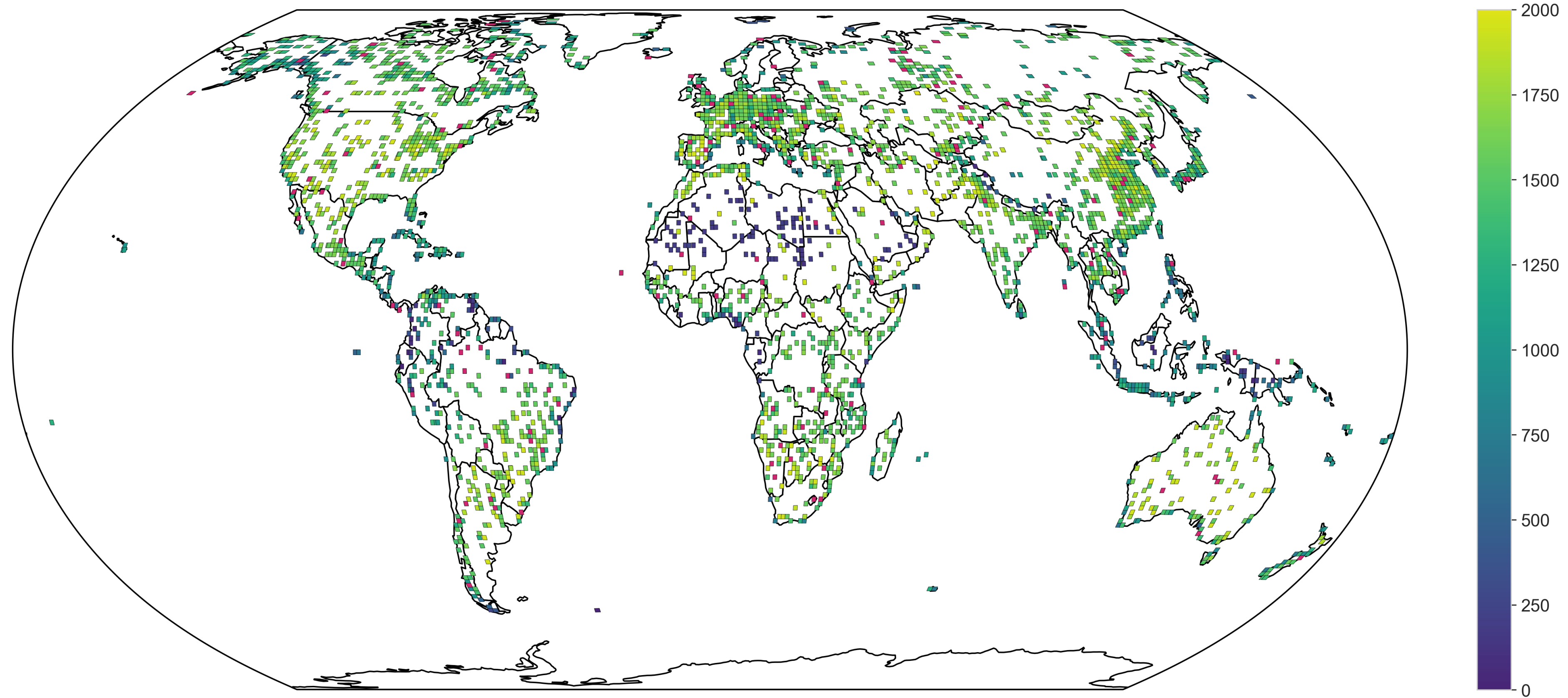




# Data: Sampling

Goal: obtain samples representing *diverse* land use and ecosystems while *minimizing cloud* and *missing value* issues.

1. Compute proportion of land use/land cover (LULC) classes and ecoregions for each HLS tile.
2. Sample tiles based on the LULC classes – urban areas are upsampled.
3. Select a 5% train-validation split: **3156** train and **168** validation tiles.



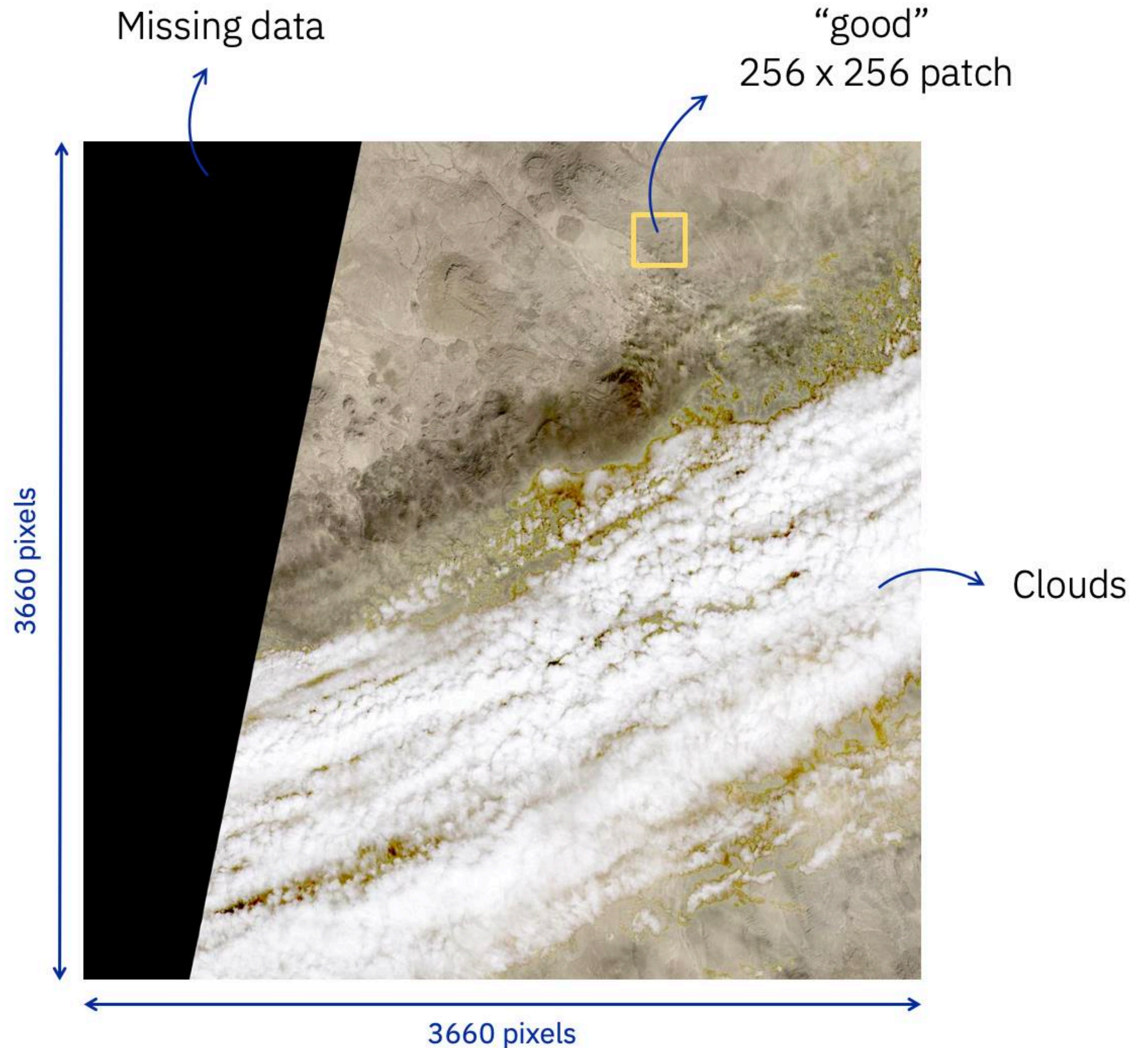
Global HLS dataset distribution visualized on a tile-level. The number of training samples are color-coded in blue to green while validation tiles are visualized in magenta.



## Data: creating samples

Once we have a representative set of tiles, the next step is creating the training samples:

1. Extract *256 x 256 good-quality patches* from selected tiles: sequences of *4* images over time.
  - consecutive images are separated by at least 1 month up to 6 months
2. We use the provided HLS mask to remove patches with more than *20% of cloudy pixels* or more than *1% of missing values*.



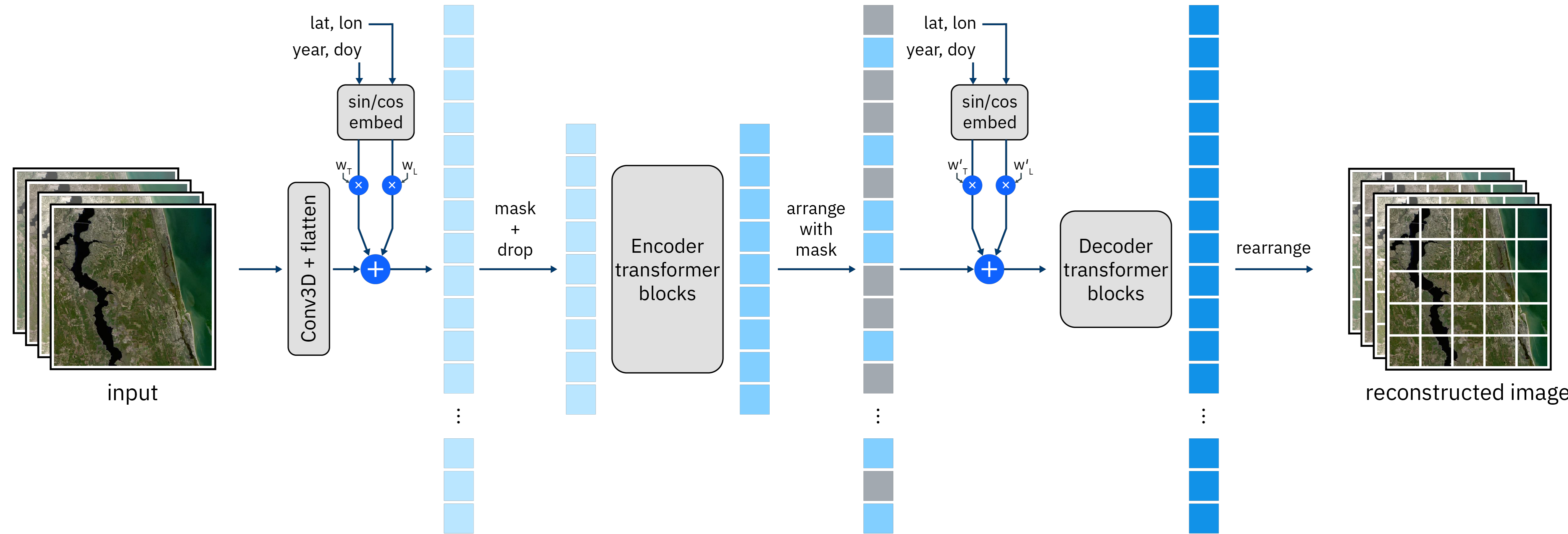


# Model architecture

Our model is based on the Masked AutoEncoder (*MAE*) approach.

The model consumes a *sequence* of 4 images with 6 bands (RGB + NIR + SWIR1/2)

*Location* and *temporal* embeddings are added to inform the model about geo-location and time of acquisition of the input samples.

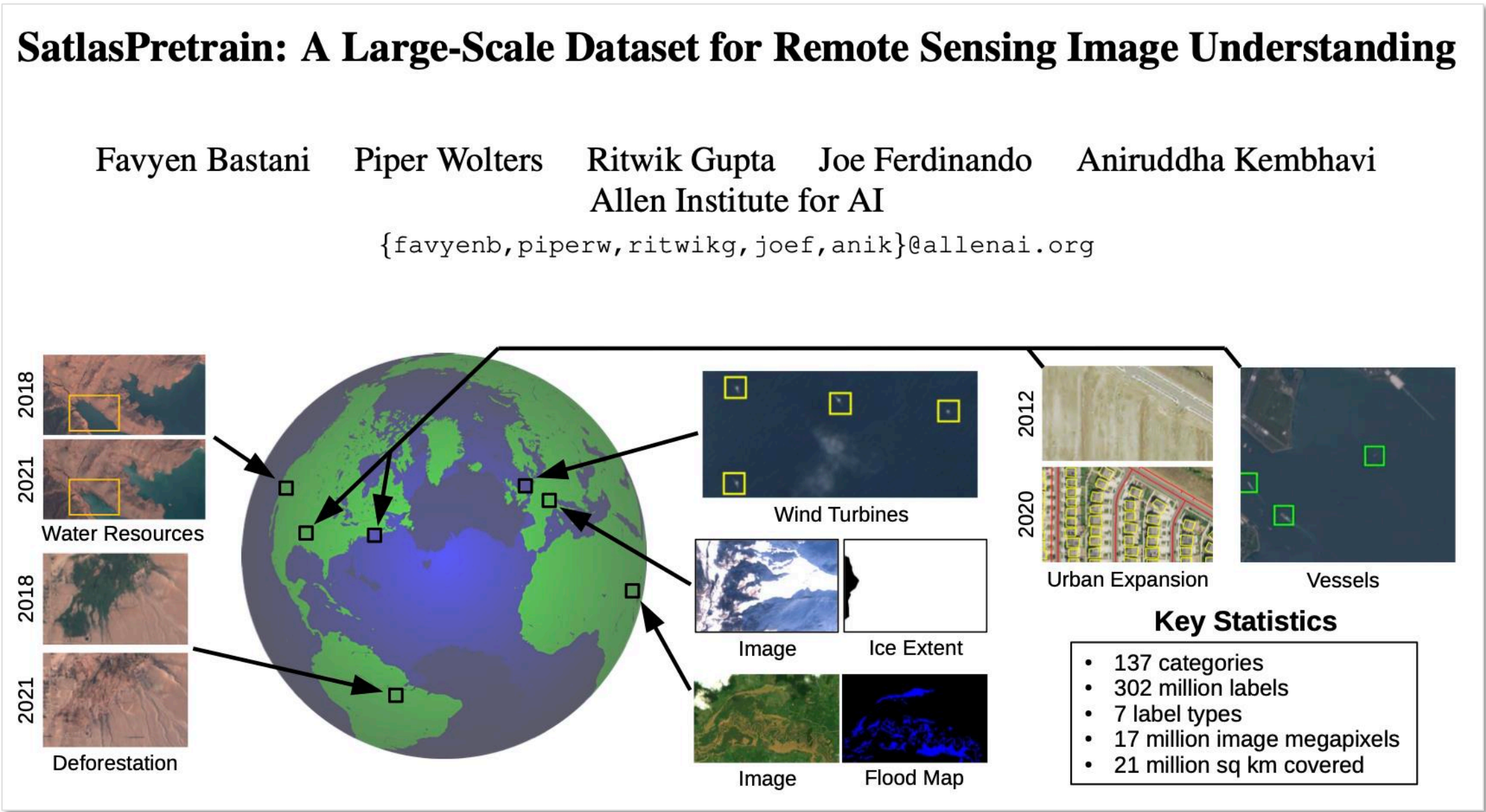


# Satlas

*SatlasPretrain* is a remote sensing dataset that combines images from Sentinel-2 and NAIP with 302M labels under 137 categories and 7 label types:

- 1. Semantic segmentation - e.g., land cover
- 2. Regression - e.g., water depth, percent tree cover
- 3. Points (object detection) - e.g., wind turbines, oil wells
- 4. Polygons (instance segmentation) - e.g., buildings, dams
- 5. Polylines - e.g., roads, rivers, railways
- 6. Properties of objects - e.g., the rotor diameter of a wind turbine.
- 7. Classification - e.g., whether an image exhibits negligible, low, or high wildfire smoke density.

Label sources: new annotation by domain experts, new annotation by Amazon Mechanical Turk workers, and processing existing datasets—OpenStreetMap, NOAA lidar scans, WorldCover, Microsoft Buildings, and C2S.



Extracted from (1)

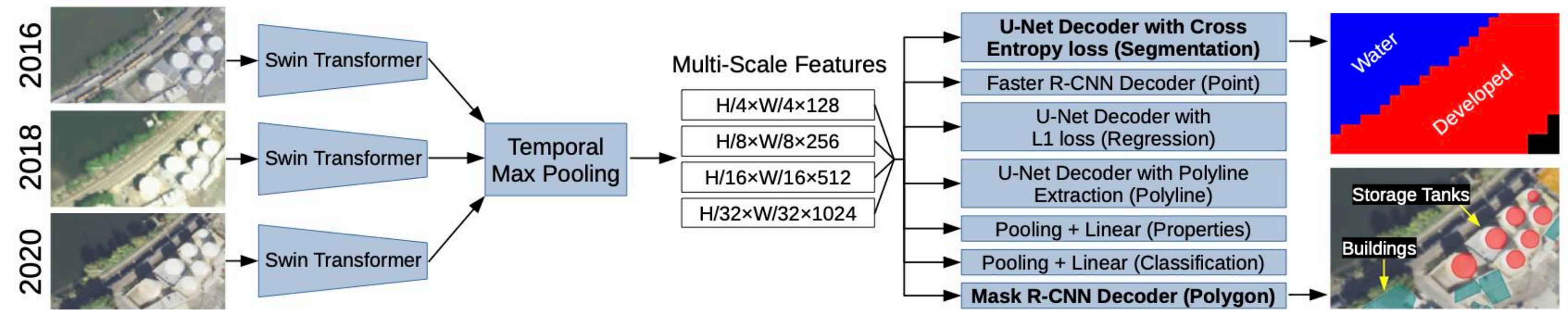


# Satlas

## Model architecture - *SatlasNet*

- Backbone = Swin transformer
- Max temporal pooling to summarize embeddings from different timesteps
- Multiple heads → one for each task

- Supervised pre-training – requires large set of labels
- Train on multiple tasks *at the same time*.
- Hierarchical backbone



Extracted from (1)



# DOFA - Dynamic One-For-All

*“DOFA is a foundation model architecture that builds on the principles of MIM (...) processing input images with any number of channels.”<sup>1</sup>*

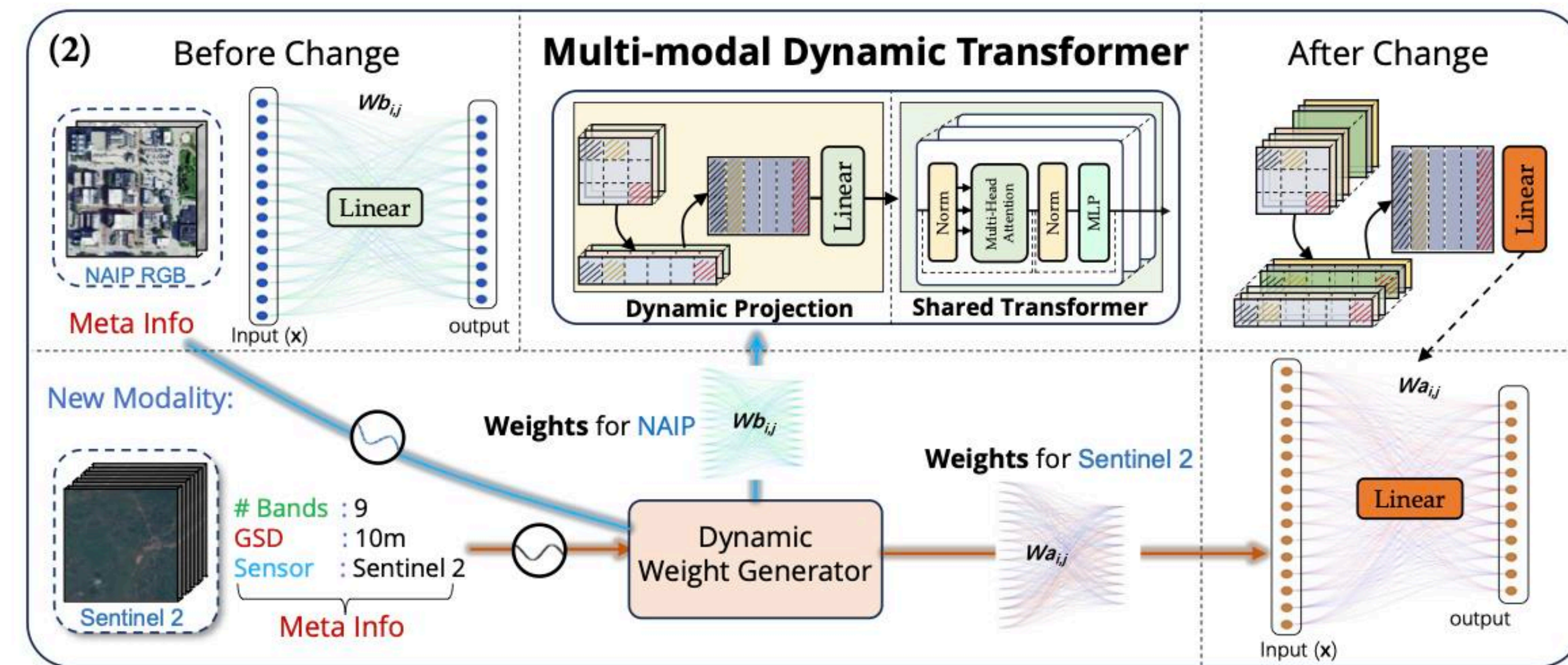
Key concepts:

**Wavelength-conditioned dynamic patch embedding:**  
project the data into the same feature *dimensionality*.

**Multimodal shared Transformer networks:**  
with a shared backbone, the model is compelled to identify and learn common features across different modalities.

**Masked image modeling with any number of spectral bands:**  
DOFA follows the *MAE* approach, but DOFA has the capacity to process input images with various channels.

**Distillation-based multimodal continual pre-training:**  
To reduce computational costs, DOFA uses a *teacher-student* method, where the teacher is an ImageNet pre-trained model.



Extracted from (1)






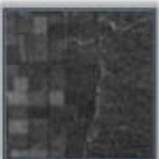

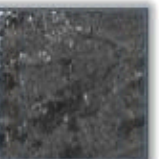
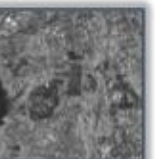

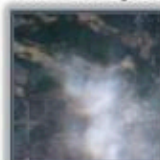


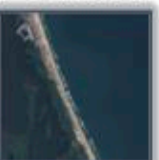











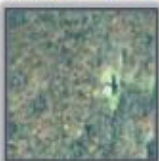

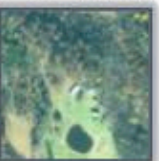


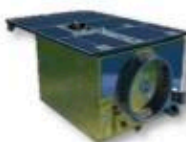
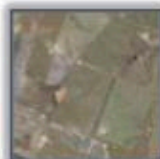

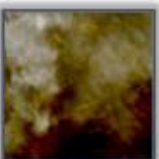
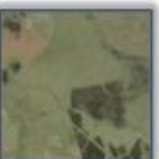
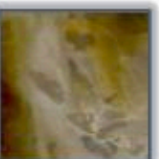
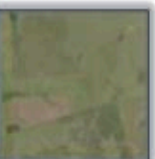
# DOFA - Dynamic One-For-All

The pre-training dataset has **5 distinct sensors**:

- RGB high-res (NAIP)
- multispectral RGB+NIR (Gaofen)
- multispectral Sentinel-2
- SAR (Sentinel 1)
- hyperspectral (EnMAP)

Pre-train *progressively*:

- 100 epochs on a 50k images subset
- 20 epochs on a 410k images subset (selecting 100k samples from each modality and 10k samples from EnMAP data)
- 1 epoch on the entire dataset

Curated Datasets from Different Sensors						
 Sentinel 1	#Samples: 4,642,353	Size: 512x512	Coverage: Global (Dense)	Sensor:	Sentinel 1	
						
	Geotag: Yes		Wavelength: Microwave (2 bands)		Capture time: Yes	
	GSD: 20 m					
 Sentinel 2	#Samples: 977,774	Size: 512x512	Coverage: Global (Dense)	Sensor:	Sentinel 2	
						
	Geotag: Yes		Wavelength: 0.49 ~ 2.15 (9 bands)		Capture time: Yes	
	GSD: 20 m					
 Gaofen	#Samples: 117,450	Size: 512x512	Coverage: China	Sensor:	Gaofen	
						
	Geotag: Yes		Wavelength: RGB,NIR (4 bands)		Capture time: Yes	
	GSD: 4 m					
 Aerial Images	#Samples: 2,332,351	Size: 512x512	Coverage: USA	Sensor:	Aerial Image	
						
	Geotag: Yes		Wavelength: RGB (3 bands)		Capture time: Yes	
	GSD: 1 m					
 EnMAP	#Samples: 11,483	Size: 128x128	Coverage: Global (Sparse)	Sensor:	EnMAP	
						
	Geotag: Yes		Wavelength: 0.46~2.45 (224 bands)		Capture time: Yes	
	GSD: 30 m					

Extracted from (1)

# Multimodal GeoFMs

# The Multimodal Foundation Models

- With the rise of the foundation models and the unprecedented interest about AI for almost any task, it did not take so long to these works converge to create models able of dealing with more than one source of data (now termed as "modalities").
- Following the advent of pioneer large models as [Wu Dao](#), the appeal of the multi-modality became ubiquitous and has spread to others fields, as SciML and Geospatial AI.
- The main question which appears when we start looking at multi-modal models is **what is the recommendable approach to combine the embeddings coming from each modality in a way they are not obfuscated or distorted during the process.**

# The Multimodal Foundation Models

- The concept of "modality" in Geospatial AI is slightly different from others areas. The inputs are not so clearly different as text and video, but they usually represent different aspects of the same object, as spectral bands in various frequencies, RGB channels and depth.
- The models presented in this summary are **Masked Auto-Encoders** (MAE), that means, they are pretrained in a **self-supervised** way to reconstruct data from inputs with missing parts. The percentage of missing parts in these samples can be considerably high in order to enforce the models to learn correlations between the scarce pieces of information and, possibly, better generalize for unseen inputs.
- To allow the images be ingested by the architectures, it is necessary to use a technique called patching, in which the images are subdivided into smaller grids and reshaped to eliminate extra dimensions.
- In the next sections, we present and comment the characteristics of some of these models and try to extract some partial conclusion about the embedding combination problem.



# MP-MAE: Single input, multiple outputs

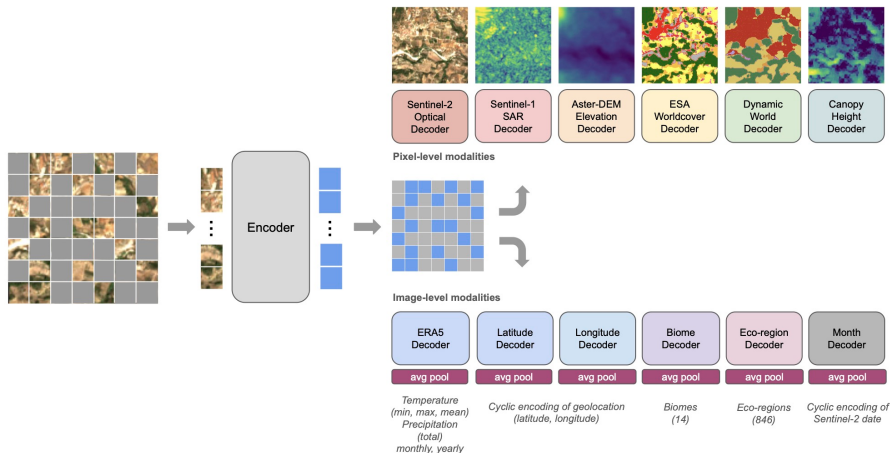


Figure: MP-MAE is ConvNext-V2 architecture trained with the MM-Earth dataset. See the [model page](#)

- **MP-MAE** is focused on learning diverse optical satellite images from Sentinel-2. That means, it receives always a single input, but it is able of reconstructing a set of other modalities.
- MP-MAE is trained on the ConvNeXt architecture, a kind of Convolutional MAE, differently from the most part of the approaches in this area which are dominantly based on transformers.
- It is multi-modal in the sense that the model can deal with different sources of data, but it does not receive more than one modality as input at a time.
- The loss used to train it is a simple weighted mean over the modality-aware losses.
- In this architecture the question about the combination of models simply does not exist, since there is a single encoder and the decoders use the same embedding as input.

# MultiMAE: Multiple inputs, multiple outputs

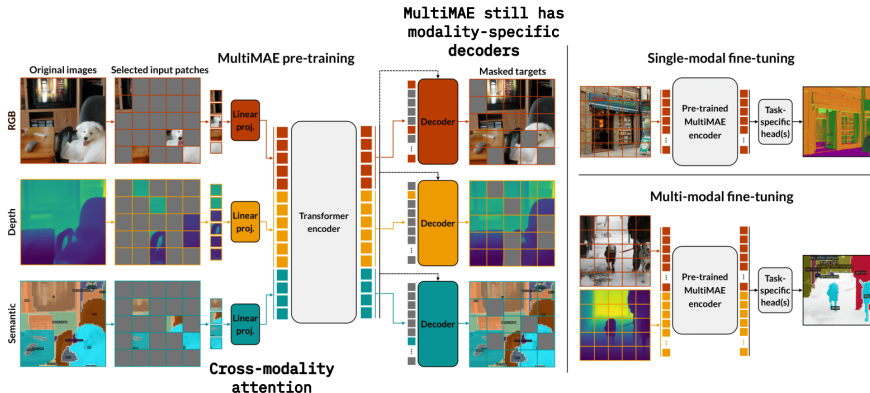


Figure: MultiMAE has crossed attention for the partial embeddings. See the [model page](#)

- **MultiMAE** receives inputs from different modalities and combines them using a simple technique: the unmasked pixels are reshaped, linearly projected and concatenated to create a common embedding.
- In this way the attention layers see the features related to all the modalities at a time and can better find relationships between them.
- After the encoding, the patches related to each modality are remapped onto their original positions and sent to dedicated decoders, which in turn reconstruct the original modalities.
- The decoders are simple shallow networks, basically a combination of multi-head attention and MLP layers.

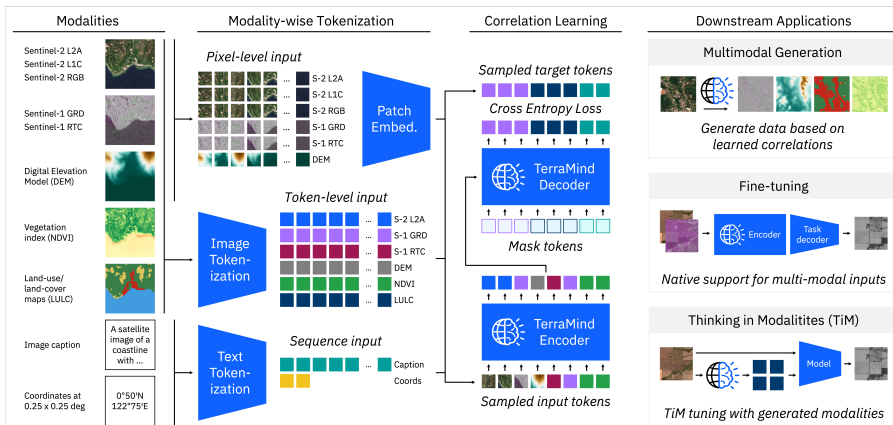


Figure: TerraMind is based on 4M. See the [model page](#)



- TerraMind has two stages for the encoding stage: tokenization and the core encoding.
- In the tokenization, an encoder-decoder architecture is pretrained for each modality in order to create the most appropriated embedding for each one.
- In the core encoding, the partial embeddings are sampled, stacked and passed through the main encoder in order to create the global embedding.
- After it, the final decoder is basically reconstructing the sampled inputs, which helps to guide the encoder towards effectively combining the multiple modalities into some kind of unified representation which can be used for further downstream tasks.

- All these models are available on our platform [TerraTorch](#).
- New models are being added, as CROMA and Galileo.

- The concept of modality can variate depending on the research area, but if we simply consider it as an individual source of data with its own defined variables, the problem of combining modalities can then be interpreted as the problem of conciliating different sources of data in order to enrich the information seen by the model during the training.
- Combining multiple modalities is not a closed problem and there are diverse solutions to achieve reasonable results.
- However the most usual currently available techniques are based on projecting the modalities using linear models or neural networks and aggregating them to produce a global embedding and then perform the decoding.

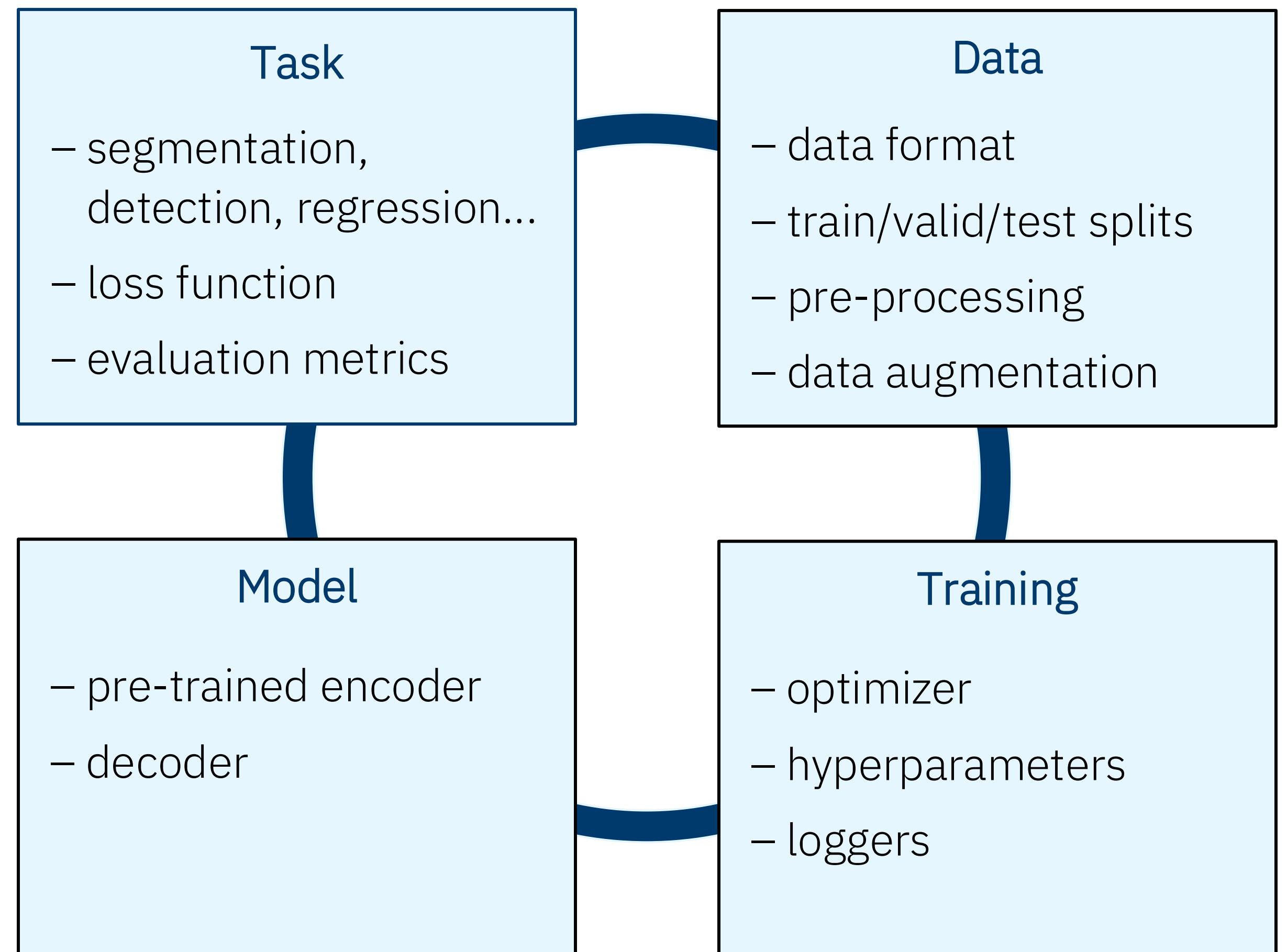
# Fine-tuning GeoFMs with TerraTorch

# Overview

TerraTorch is an open-source library based on Pytorch Lightning and TorchGeo designed to *streamline* the process of fine-tuning *geospatial foundation models* (GFM) for different downstream tasks.

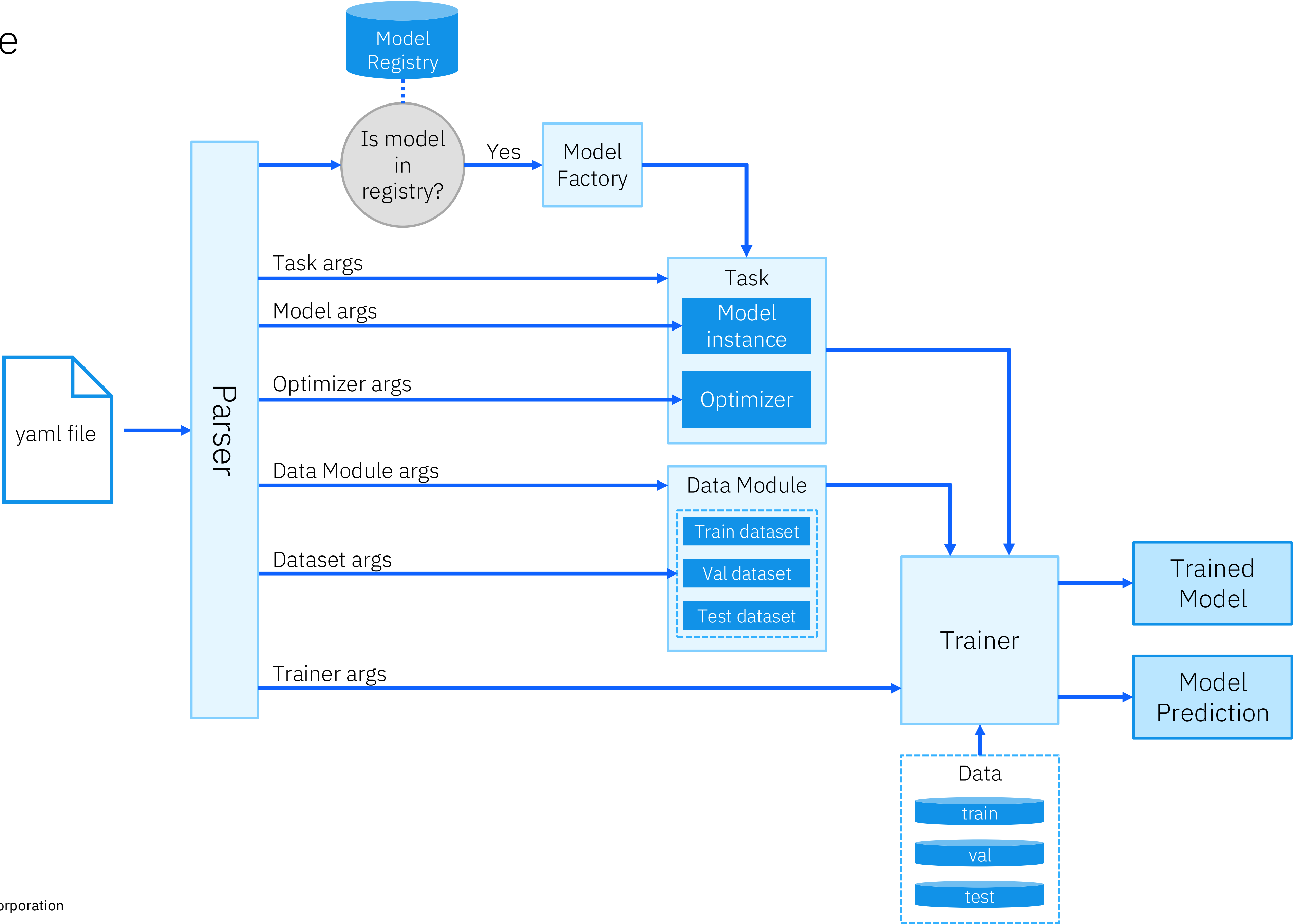
## Features

- TorchGeo functionalities
- Easy access to GFM backbones (Prithvi, timm, SMP)
- Flexible trainers for image segmentation, pixel-wise regression and classification (more in progress)
- Launching fine-tuning tasks through config files





# Architecture



# Practical Examples

<https://drive.google.com/drive/folders/1RfQbOiIdpTZ-81VaSSKw4v0vDgRf-7bE>

[https://github.com/Joao-L-S-Almeida/tutorial\\_neurips](https://github.com/Joao-L-S-Almeida/tutorial_neurips)

# Benchmarking GeoFMs

# GEO-Bench-2

The emergence of **Geospatial Foundation Models (GeoFMs)** holds great promise for advancing **Earth Observation (EO)**, enabling more general and scalable solutions for a variety of tasks.

However, the rapidly evolving nature of this field has meant that **evaluation protocols** have been difficult to standardize.

**GEO-Bench-2** is an effort to address this challenge by providing a **comprehensive and community-focused benchmarking framework** tailored to various EO applications.

Expanding upon its predecessor, this framework is designed to facilitate **consistent, insightful, and fair** comparison of GeoFMs.

<https://github.com/The-AI-Alliance/GEO-Bench-2>

*Simumba, N. et al. GEO-Bench-2: From Performance to Capability, Rethinking Evaluation in Geospatial AI, <https://arxiv.org/pdf/2511.15658>, November 2025.*

# What GEO-Bench-2 Offers

## Diverse and Permissively Licensed Data

Includes a curated selection of **19 datasets** covering core EO tasks, including **classification, segmentation, regression, object detection, and instance segmentation**, ensuring broad usability.

## Targeted Evaluation via “Capabilities”

Datasets are grouped into “**capabilities**” based on shared characteristics (e.g., resolution, band usage, temporality).

This feature supports flexible benchmarking, allowing users to assess a model’s strengths on specific types of EO data.

## Robust Metrics and Efficiency

Utilizes the **normalized interquartile mean (IQM)** for more robust model comparison and incorporate subsampling strategies to help make large-scale evaluation more efficient.

## Ease of Use with TerraTorch

Integration with the TerraTorch open-source toolkit. However, all datasets and datamodules can also be used independently of TerraTorch.



# GEO-Bench-2

## Dataset Selection Criteria

### Challenging and Discriminative

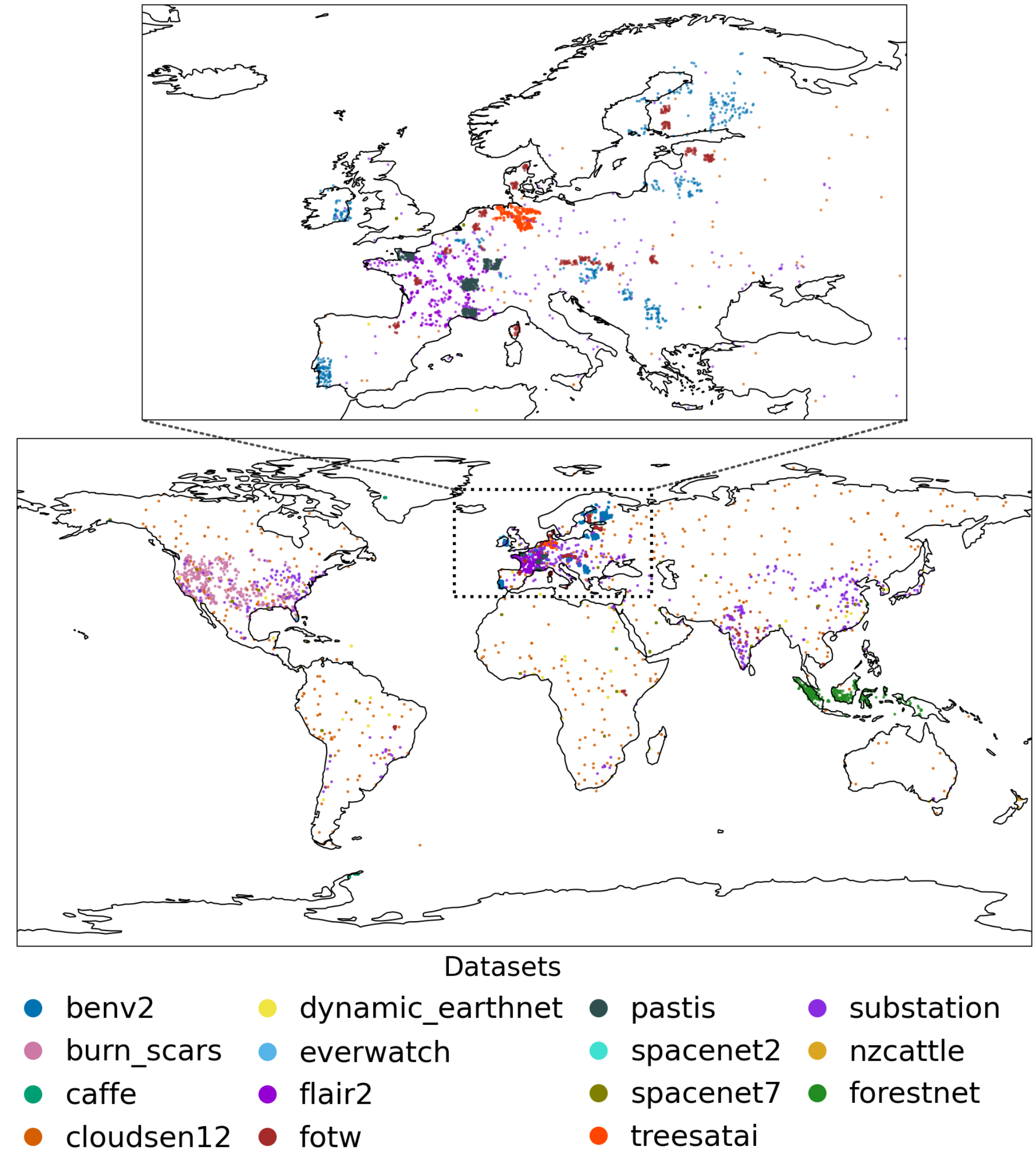
Datasets must distinguish strong GeoFMs from baseline models.

### Open Licenses

Prioritizing permissive licenses to enable academic and industry adoption, avoiding GPL and non-commercial licenses.

### Diversity

Encompasses a wide range of tasks, modalities, and geographic regions, featuring samples from all seven continents. The comparatively higher representation of Europe stems from initiatives such as INSPIRE and Horizon Europe.



# GEO-Bench-2

## Dataset Capability Groups

Dataset	Core	Pixel wise	Classi- fication	Detection	Multi Temporal	< 10m Res	≥10m Res	RGB/ NIR	Multi Spectral
BEN V2	✓		✓				✓		✓
TreeSatAI	✓		✓			✓		✓	
So2Sat			✓				✓		✓
ForestNet			✓				✓		
BioMassters	✓	✓			✓		✓		✓
CaFFe		✓					✓		
CloudSEN12	✓	✓					✓		✓
NASA Burn Scars	✓	✓					✓		✓
Dynamic Earth Net		✓			✓	✓		✓	
FLAIR 2	✓	✓				✓		✓	
FTW	✓	✓					✓	✓	✓
KuroSiwo	✓	✓			✓		✓		
PASTIS	✓	✓			✓		✓		✓
SpaceNet 2		✓				✓			✓
SpaceNet 7	✓	✓				✓		✓	
EverWatch	✓			✓					
NZCattle				✓					
PASTIS (R) panoptic				✓					
Substations	✓			✓					

GEO-Bench-2 evaluates **nine overlapping capabilities** designed to highlight specific challenges in GeoFM development.

These include architectural challenges, such as object detection and segmentation, as well as input-related challenges like sensor fusion and temporal understanding.

**Core** provides a balanced subset across all capabilities with the aim of reducing the compute needed for evaluating on the benchmark while evaluating on the more discriminative datasets.



# Models selected for experiments

Model	Type	# Backbone Params	Learning Technique	Data	Res	N	T	License
Resnet50-ImageNet	ResNet-50	25M	Supervised	ImageNet-22k	NA	14M	1	Apache 2.0
ConvNext-Large-ImageNet [61]	ConvNext	230M	Supervised	ImageNet-22k	NA	14M	1	Apache 2.0
ConvNext-XLarge-ImageNet [61]	ConvNext	390M	Supervised	ImageNet-22k	NA	14M	1	Apache 2.0
DINOv3-ViT-L-SAT [50]	ViT	300M	Distillation	Maxar RGB	0.6 m	493M	1	DINO V3
DINOv3-ConvNext-Large-WEB [50]	ConvNext	230M	Distillation	LVD-1689M	NA	1689M	1	DINO V3
Resnet50-DeCUR [58]	ResNet-50	25M	Contrastive	Sentinel-2	10 m	1M	1	Apache 2.0
DOFA-ViT-300M [63]	ViT	300M	MAE	Sentinel-1 and -2, EnMap, Gaofen, Landsat	1-30 m	8M	1	CC-BY-4.0
Clay-V1 ViT-B [1]	ViT	86M	MAE	Landsat 8 and 9, Sentinel-1 and -2, NAIP, LINZ, MODIS	1-30 m	70M	1	Apache 2.0
Satlas-SwinB-Sentinel2 [10]	Swin	88M	Supervised	Sentinel-2	10 m	NA	1	ODC-BY
Satlas-NAIP [10]	Swin	88M	Supervised	NAIP	1 m	NA	1	ODC-BY
Prithvi-EO-V2-300M-TL [52]	ViT	300M	MAE	HLS	30 m	4.2M	4	Apache 2.0
Prithvi-EO-V2-600M-TL [52]	ViT	600M	MAE	HLS	30 m	4.2M	4	Apache 2.0
TerraMind-V1-Base [34]	ViT	86M	Correlation	Sentinel-1 and -2, LULC, DEM, NDVI	10 m	9M	1	Apache 2.0
TerraMind-V1-Large [34]	ViT	300M	Correlation	Sentinel-1 and -2, LULC, DEM, NDVI	10 m	9M	1	Apache 2.0

# Adaptation protocol

## Base Model Adaptation

- Classification:**  
Single linear layer with softmax on the encoder output
- Pixel-wise Tasks (Segmentation and Regression):**  
UNet decoder for both semantic segmentation and regression tasks, where we fed equally spaced features from the encoder’s output into the Unet.
- Object Detection and Instance Segmentation:**  
Faster R-CNN and Mask R-CNN, respectively

## Multi-Spectral Bands, SAR and Multi-Modal Datasets

- Multi-Spectral Bands:**  
All bands contained in a dataset are utilized, provided they are compatible with the model. Where an exact match did not exist, bands were matched according to closest wavelength.
- Synthetic Aperture Radar:**  
In case a model could not handle Synthetic Aperture Radar (SAR) natively, VV and VH polarization bands were loaded as the model’s RGB channels in the order VV, VH, and VV.
- Multi-Modal Datasets:**  
Ablation comparing S2-only to S1+S2 for multi-modal datasets. This led to using both modalities only for Terramind on BEN V2 and BioMassters; all other models use S2 only.

## Multi-Temporality

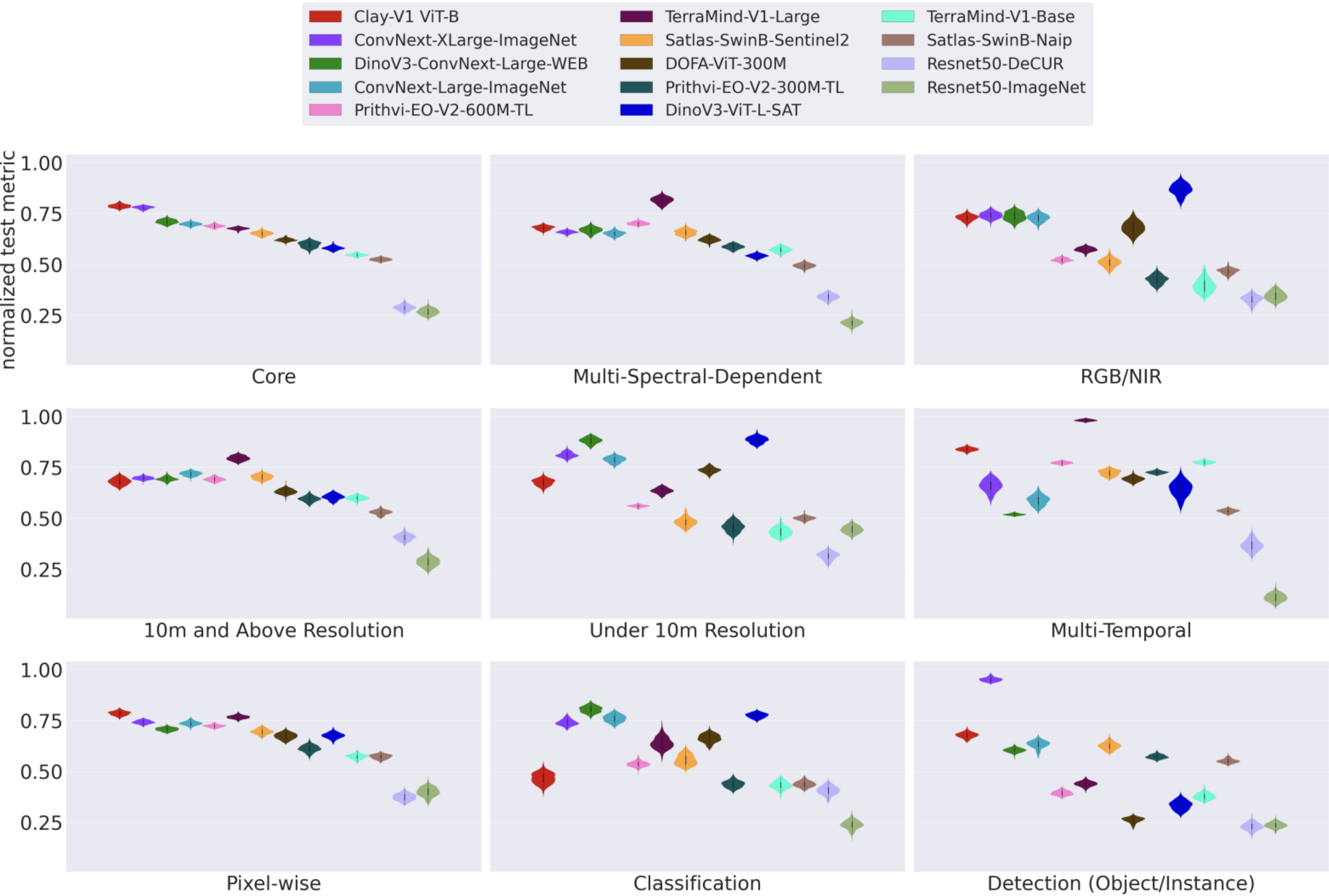
For datasets with multi-temporal inputs, each timestamp was passed through the encoder separately. The resulting encoder outputs were then averaged along the embedding dimensions before being passed to the decoder.

## Evaluation Metrics

- Semantic Segmentation:** Multiclass Jaccard Index
- Single label Classification:** Accuracy
- Multilabel classification:** F1-score
- Pixel Wise Regression:** Root Mean Squared Error
- Instance segmentation and Object detection:** Mean Average Precision



# Results across capabilities



# Impact of Architecture, Size, and Pretraining Dataset

Despite some models performing more consistently across several capabilities (i.e., Clay-V1 ViT-B, or ConvNeXt architectures), no model dominates across all datasets.

Larger models consistently outperform their smaller counterparts, with ResNet-50 models showing notably poor performance regardless of pretraining data.

Clay-V1 ViT-B achieves top Core performance with only 86M parameters. This is likely due to its diverse pretraining on 70M heterogeneous EO samples (1-30m GSD) and smaller patch size of 8, which captures finer details at 4× computational cost compared to standard ViT models with patch size 16.

ConvNeXt architectures pretrained on natural images adapt effectively to EO tasks through full finetuning, independent of model size or pretraining domain.

# Importance of Multi-Spectral Bands

DINOv3-ViT-L-SAT and the ConvNeXt models are top performers in the RGB/NIR or the below 10 m GSD capability, which mostly includes high-resolution RGB data to perform the task.

Despite TerraMindV1-Large, Prithvi-EO-2.0-600-TL, and Clay-V1 ViT-B taking the podium in the Multi-Spectral-Dependent capability, the ConvNeXt models are still close in normalized performance.

In individual datasets like NASA Burn Scars and PASTIS crop classification, there is a more marked difference (e.g. up to 10%) between the models using all multi-spectral bands available and the ones using only RGB data.

# Discussion and Limitations

**GeoFMs are advancing rapidly**, with a strong appeal for domain-specific applications.

- However, the field has yet to experience a paradigm shift comparable to that of generative modeling in text or computer vision.
- While larger models tend to perform better, **geoFMs have not exhibited the scaling laws** observed in LLMs or image generation.

Under full fine-tuning settings, GeoFMs **consistently outperformed simpler models, but not larger models trained on natural images**, particularly for high-resolution RGB-only tasks.

GeoFMs demonstrated clear **advantages** when **multi-spectral** or **multi-temporal** information is critical.

- These results align with DINOv3 findings, where even the 7B RGB-only model could not surpass the 10× smaller Prithvi-EO-2.0 in crop classification.





# Questions?