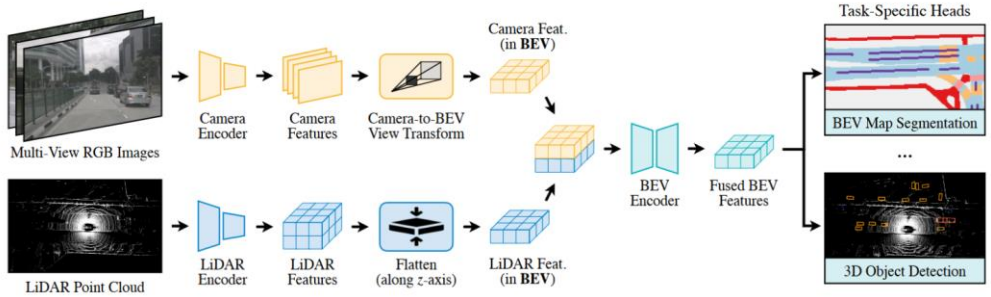
“Attention” and “Transformer” Architectures

James Hays

Some ViT slides from Noah Snavely.

Recap – 3D point processing

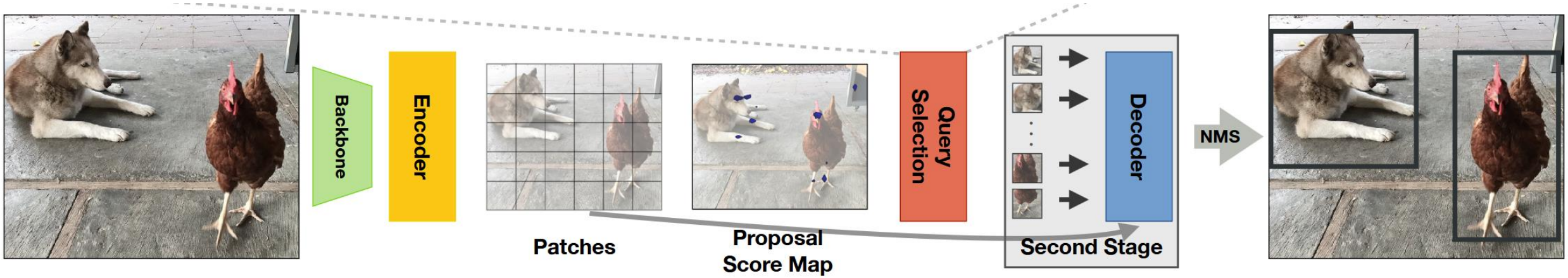
- Popular CNN backbones aren't a direct fit for 3D point processing tasks.
- It's not clear how best to use deep learning on 3D data
  - Use a truly permutation invariant representation (PointNet)
  - Use a voxel representation (VoxelNet)
  - Use a bird's a view representation (PointPillars)
  - Create a range image
- With lidar, multi-modal approaches (adding images, radar) help surprisingly little compared to lidar-only approaches (~3 mAP).



BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation.
Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L. Rus, Song Han

Recap – Object Detection

- Break up complicated tasks into subtasks, e.g. propose objects and then classify objects
- Many recent transformer architectures (e.g. DETR) perform object detection in one stage with no non-maximum suppression



NMS Strikes Back.
Jeffrey Ouyang-Zhang, Jang Hyun Cho, Xingyi Zhou, Philipp Krähenbühl. 2022

Method	Backbone	Extra Data	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	FPS
Deformable-DETR [40]	ResNeXt-101-DCN	-	50.1	69.7	54.6	30.6	52.8	65.6	4.6
EfficientDet-D7x [30]	EfficientNet-D7x-BiFPN	-	55.1	73.4	59.9	-	-	-	*6.5*
ScaledYOLOv4 [32]	CSPDarkNet-P7	-	55.4	73.3	60.7	38.1	59.5	67.4	*16*
CenterNet2 [38]	Res2Net-101-DCN-BiFPN	-	56.4	74.0	61.6	38.7	59.7	68.6	*5*
CopyPaste [10]	EfficientNet-B7	Objects365	56.0	-	-	-	-	-	-
HTC++ [3, 21]	Swin-L	-	57.7	-	-	-	-	-	-
$\mathcal{H}$-Deformable-DETR [13]	Swin-L	-	58.3	**77.1**	63.9	**39.8**	61.5	72.7	4.6
DINO [36]	Swin-L	-	**58.6**	76.9	64.1	39.4	61.6	73.2	2.7
Improved Deformable-DETR	Swin-L	-	56.6	75.6	61.9	38.8	60.4	73.5	4.3
DETA (Ours)	Swin-L	-	58.5	76.5	**64.4**	38.5	**62.6**	**73.8**	4.2
DINO [36]	Swin-L	Objects365	63.3	-	-	-	-	-	2.7
DETA (Ours)	Swin-L	Objects365	**63.5**	**80.4**	**70.2**	**46.1**	**66.9**	**76.9**	4.2

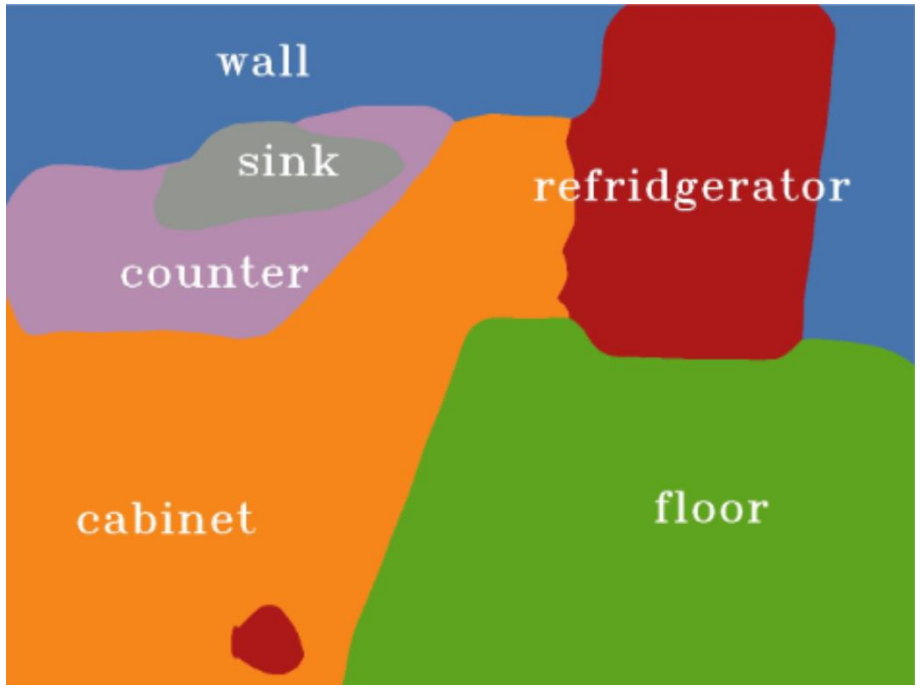
Table 4. **Comparisons to prior work with larger backbones on COCO test-dev.** The results are taken from original works except $\mathcal{H}$-Deformable-DETR and DINO where models are taken from their repositories and evaluated on the official test server. We train our model with Swin-L backbone [21] with a 2× schedule. FPS are reported on the same V100 machine whenever possible. Italicized FPS are taken from original texts. Top block: traditional NMS-based detectors; Middle block: transformer-based end-to-end object detectors. Bottom block: transformer-based detectors with Objects365 pretraining.

NMS Strikes Back.
Jeffrey Ouyang-Zhang, Jang Hyun Cho, Xingyi Zhou, Philipp Krähenbühl. 2022

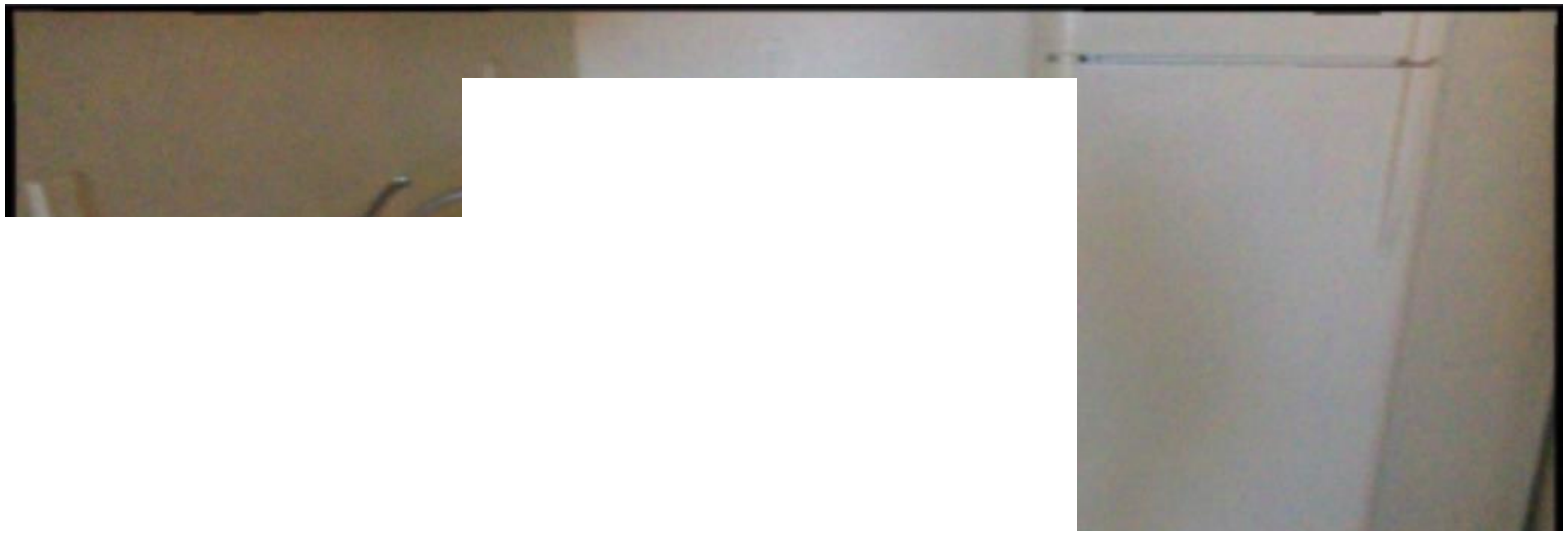
Outline

- Context and Receptive Field
- Going Beyond Convolutions in…
  - Text
  - Images
  - Point Clouds
- Notable Transformer “Foundation” models
  - CLIP-like models
  - DINO models

Ground truth



Prediction from Mseg



Language understanding

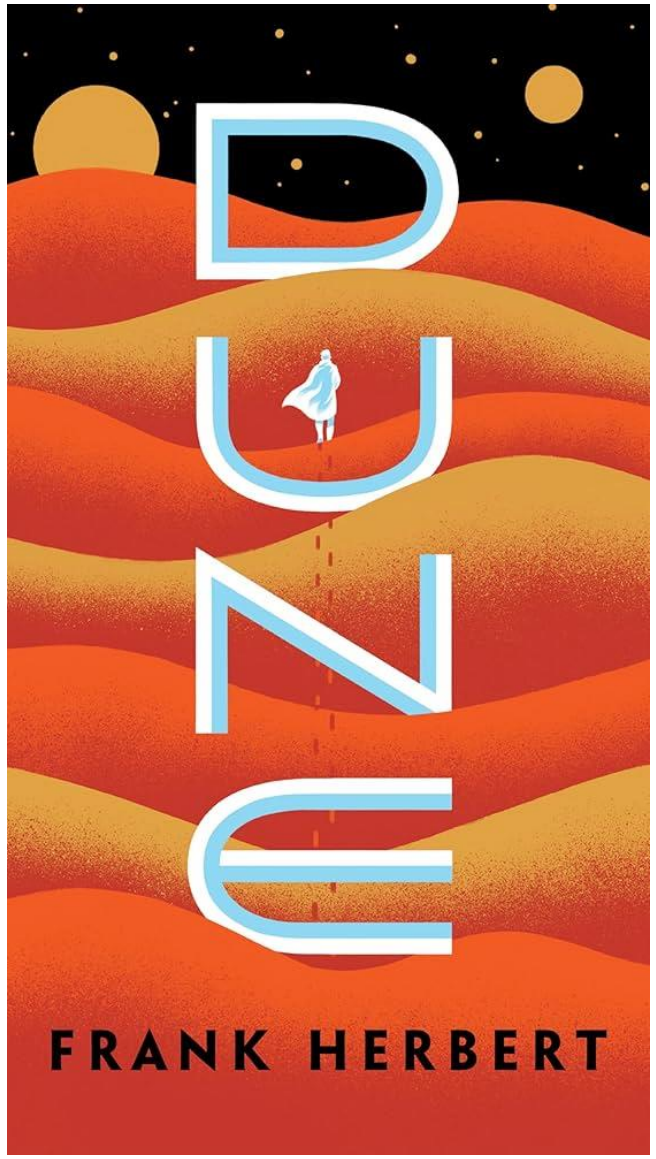
… serve …

Language understanding

… great serve from Djokovic …

Language understanding

… be right back after I serve these salads …

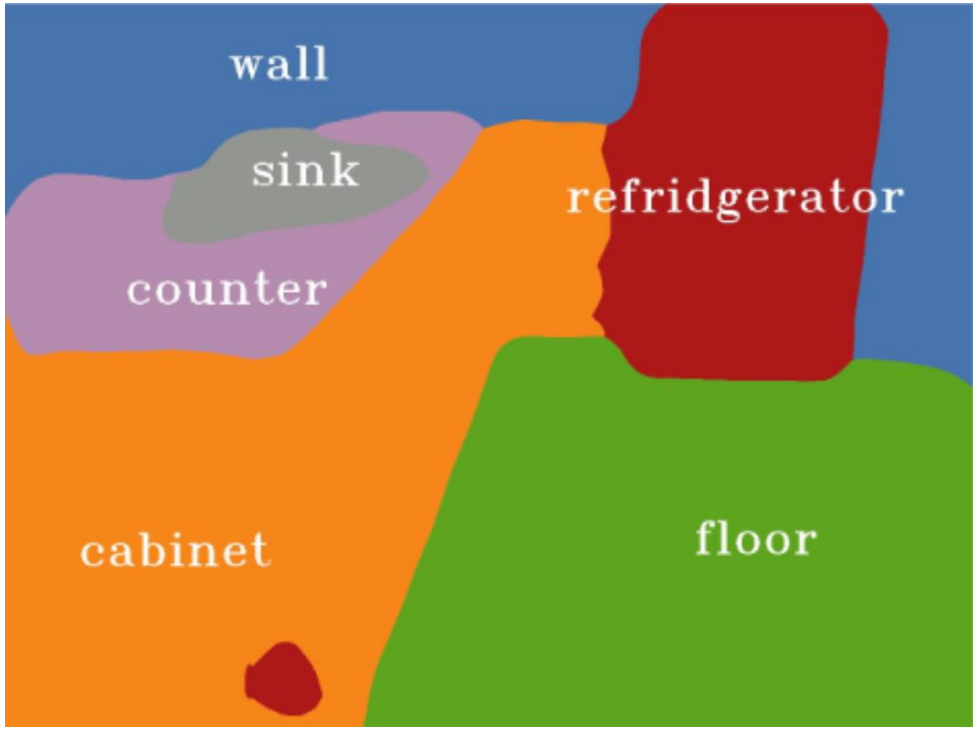


Longer Range Dependencies

The text of Dune + “Which characters possess the Atreides Ducal Signet?”

1. Duke Leto Atreides
As the head of House Atreides, Leto possesses the ring at the start of the series. During the Harkonnen/Sardaukar sneak attack on Arrakis, Leto is captured. Knowing he is going to his death, his primary concern is the succession. He manages to pass the ring to his loyal Mentat, Thufir Hawat, before being taken to the Baron.
2. Thufir Hawat
Hawat carries the ring briefly after the fall of House Atreides. Despite being captured by the Harkonnens and forced into their service (via "residual poison" coercion), he manages to keep the ring hidden or secured. His ultimate goal remains the preservation of the Atreides legacy.
3. Paul Atreides (Muad'Dib)
Paul receives the ring while living among the Fremen in the deep desert. In the novel, it is delivered to him via a message/courier from Hawat (who is being held by the Harkonnens). Taking the ring is a pivotal moment; it marks Paul's formal acceptance of the title **Duke of Arrakis** and signals his intent to challenge the Emperor and House Harkonnen.

So how do we fix these problems?



Receptive field could also be an issue in 3D

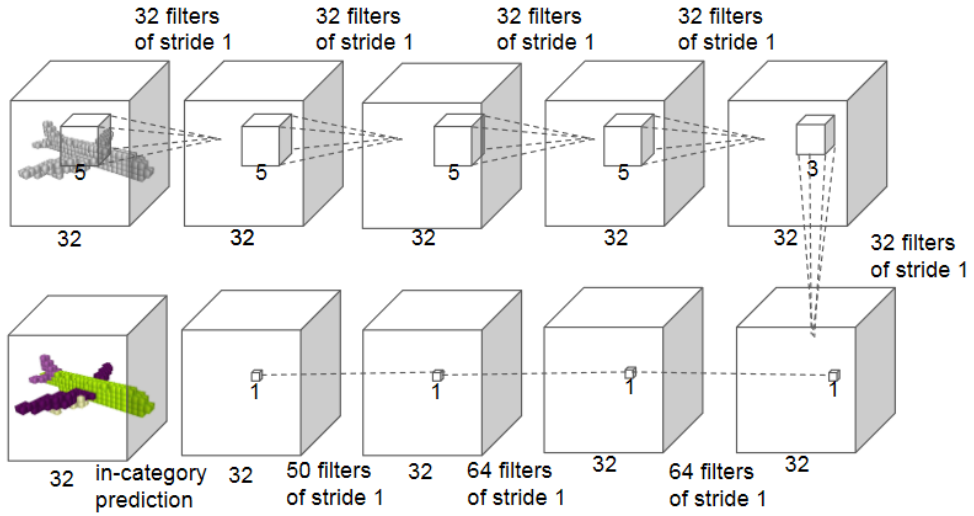


Figure 10. **Baseline 3D CNN segmentation network.** The network is fully convolutional and predicts part scores for each voxel.

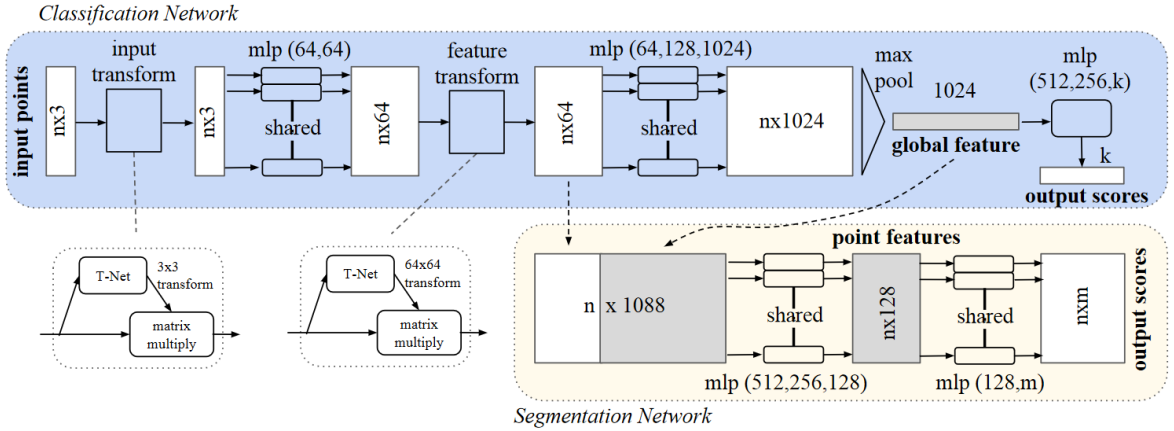
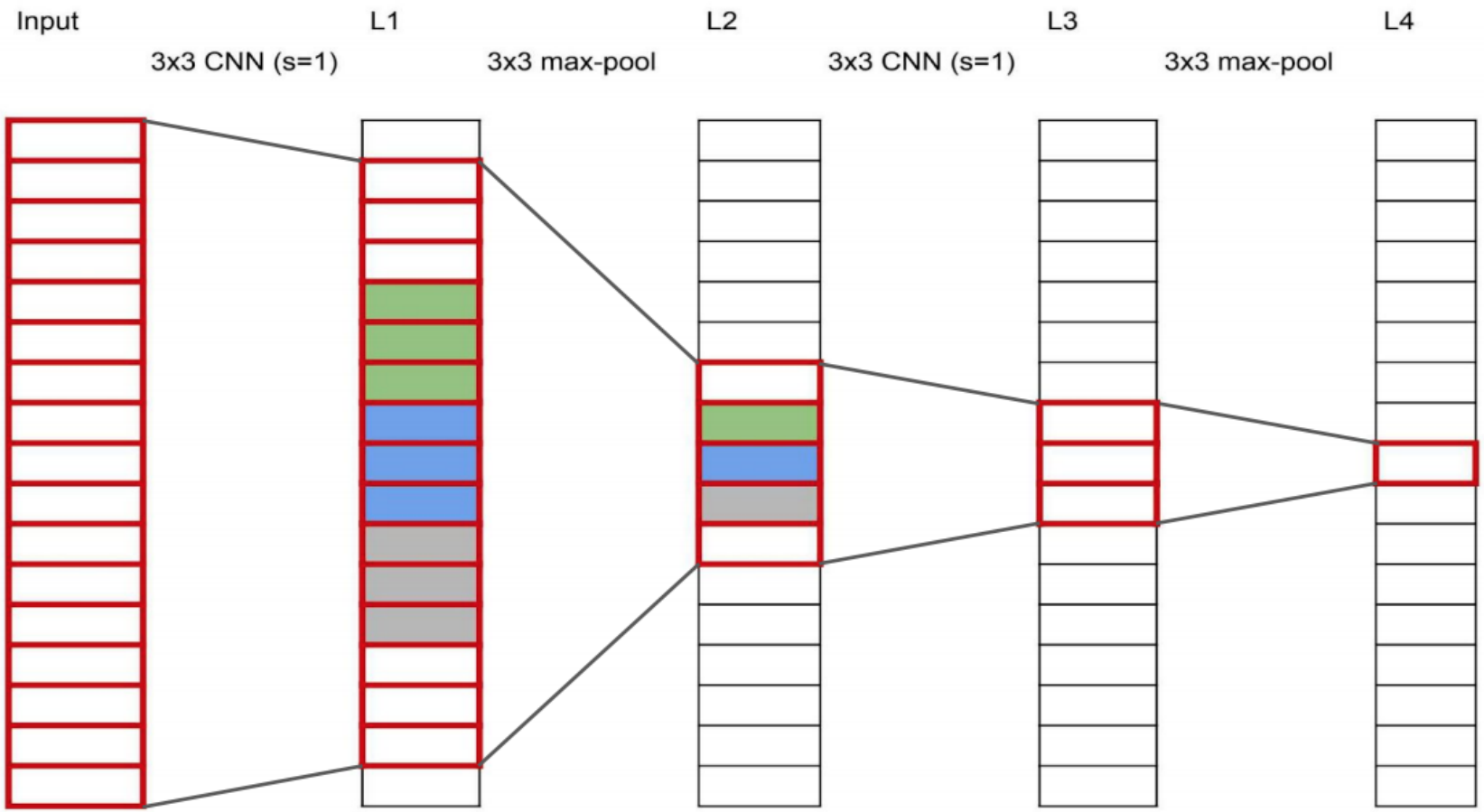


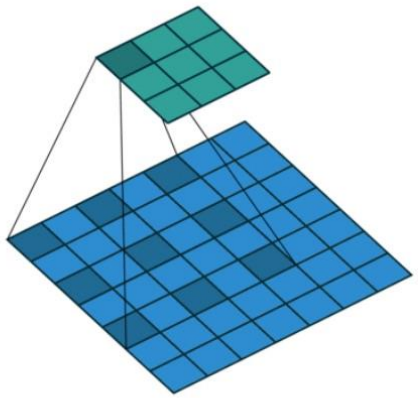
Figure 2. **PointNet Architecture.** The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net. It concatenates global and local features and outputs per point scores. "mlp" stands for multi-layer perceptron, numbers in bracket are layer sizes. Batchnorm is used for all layers with ReLU. Dropout layers are used for the last mlp in classification net.

Receptive field

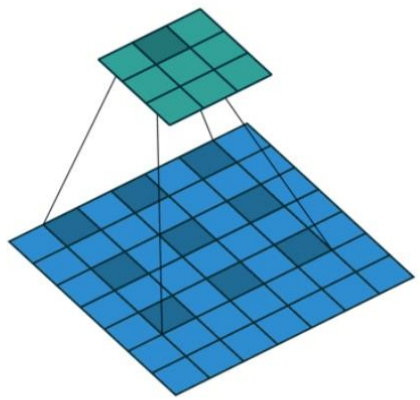


Slide Credit: Frank Dellaert https://dellaert.github.io/19F-4476/resources/receptiveField.pdf

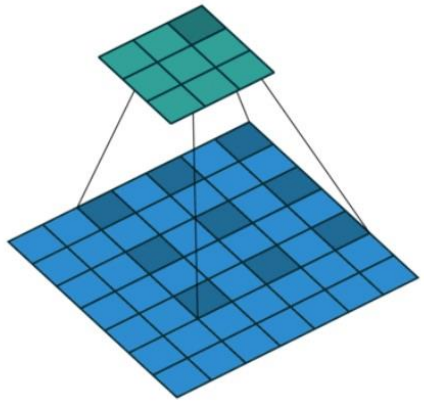
Dilated Convolution



No padding, no stride, dilation



No padding, no stride, dilation



No padding, no stride, dilation

Figure source: https://github.com/vdumoulin/conv_arithmetic

Outline

- Context and Receptive Field
- Going Beyond Convolutions in…
  - Text
  - Point Clouds
  - Images
- Notable Transformer “Foundation” models
  - CLIP-like models
  - DINO models

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer,

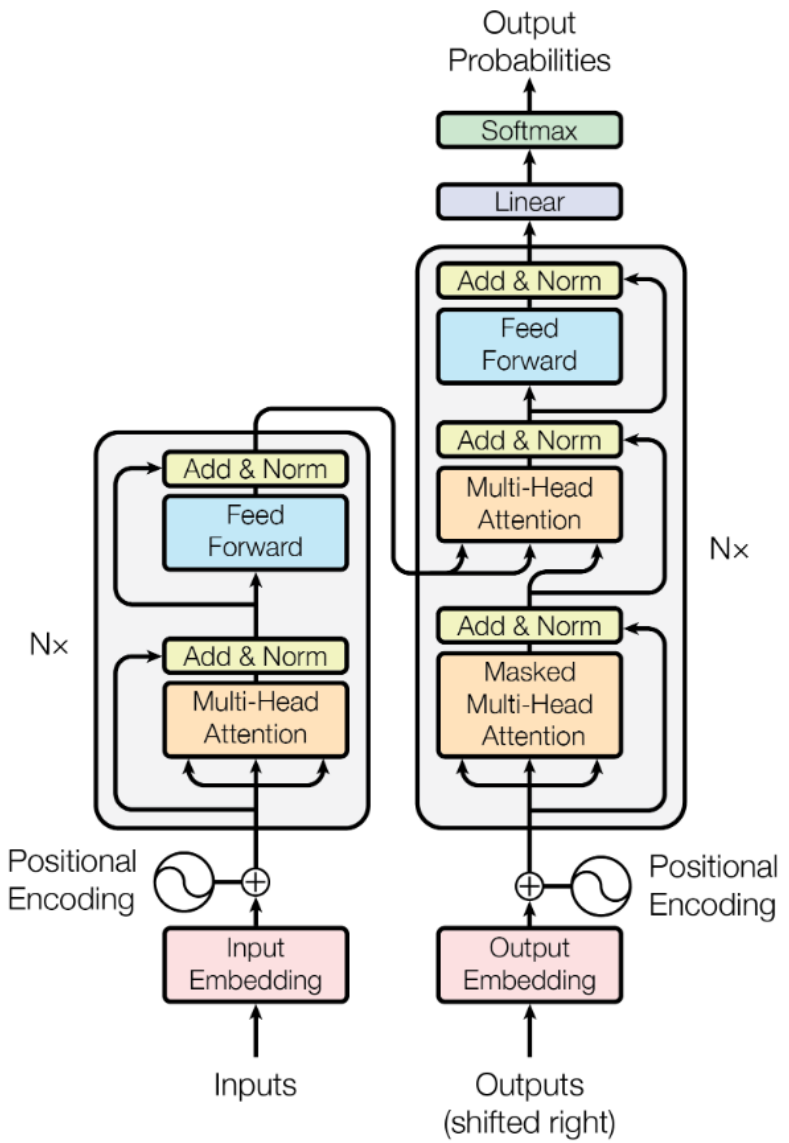
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

“Many readers will skim over formulas on their first reading of your exposition. Therefore, your sentences should flow smoothly when all but the simplest formulas are replaced by “blah” or some other grunting noise.” - Donald Knuth

Transformer Architecture

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

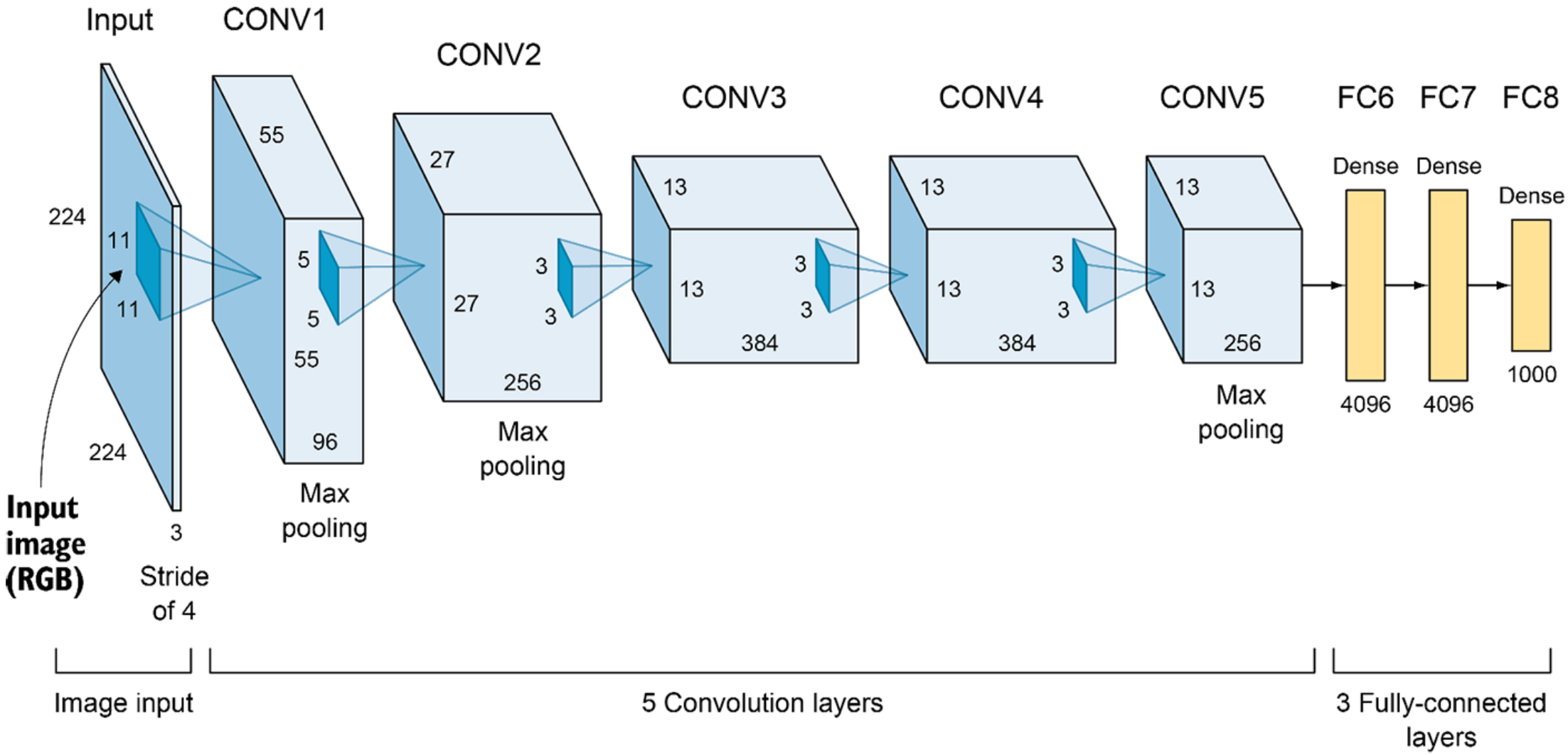
Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	



Outline

- Context and Receptive Field
- Going Beyond Convolutions in…
  - Text
  - Images
  - Point Clouds
- Notable Transformer “Foundation” models
  - CLIP-like models
  - DINO models

Recall: ConvNets



Elgendy, *Deep Learning for Vision Systems,* https://livebook.manning.com/book/grokking-deep-learning-for-computer-vision/chapter-5/v-3/

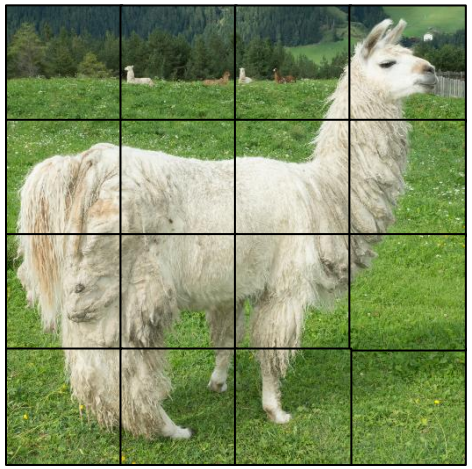
ConvNets assume spatial locality

- Assume nearby pixels are more important to making decisions than far away pixels (an example of an “inductive bias”)
- Only after stacking together several convolutional layers with spatial downsampling can distant pixels “talk” to each other
- As image datasets grow, maybe we can do better by removing the spatial locality assumption and *learning how to process images from scratch*

An alternative to convolution: Attention

- Goal: consider long-range relationships between pixels

An alternative to convolution: Attention



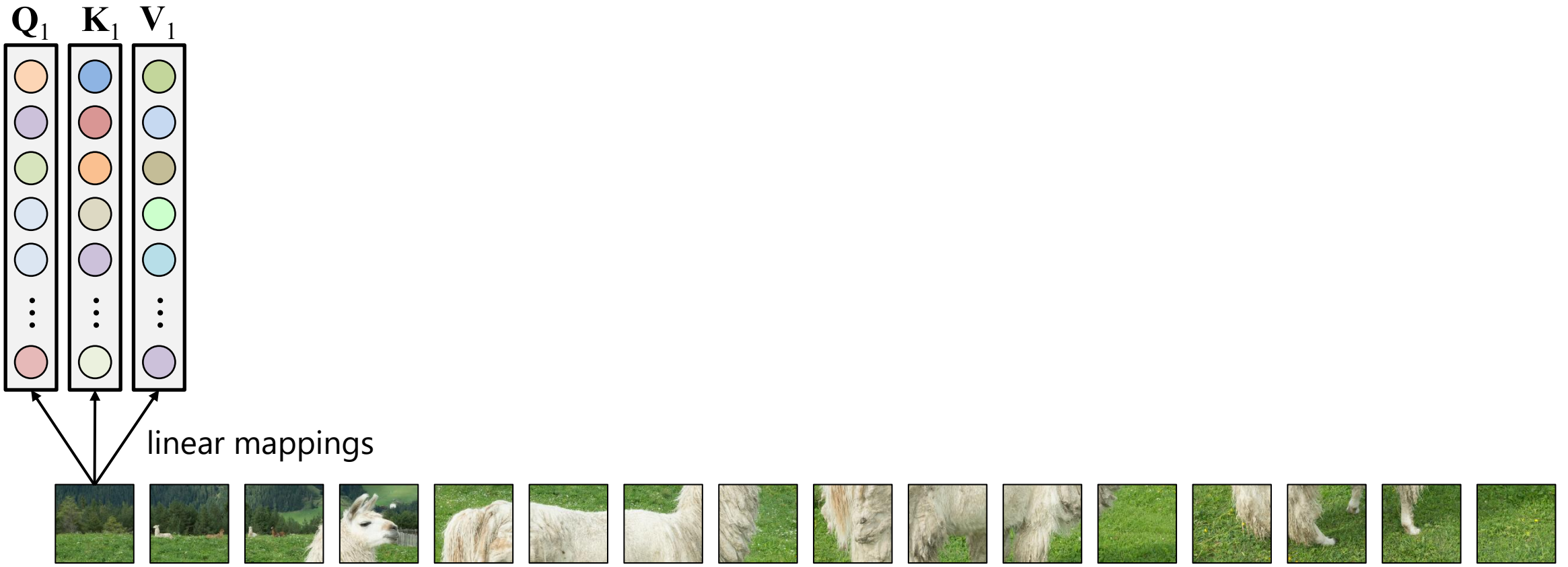
Step 1: Break image into patches

An alternative to convolution: Attention

Step 1: Break image into patches

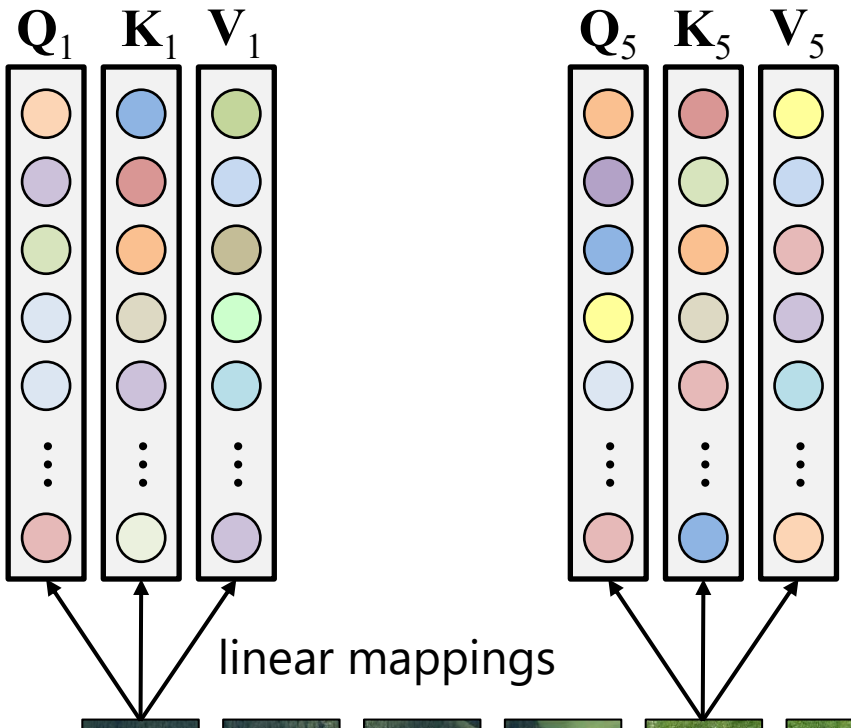
An alternative to convolution: Attention

Step 2: Map each patch to three vectors:
Query (Q), Key (K), and Value (V)



An alternative to convolution: Attention

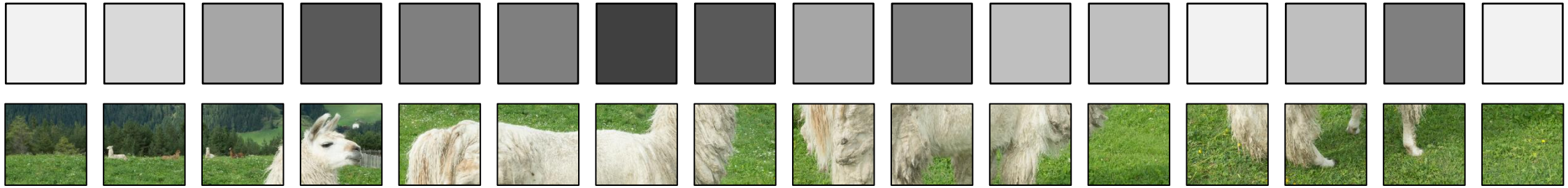
Step 3: For each patch, compare its query vector to all key vectors



$$w_{1,5} = \mathbf{Q}_1 \cdot \mathbf{K}_5 = \boxed{0.2}$$

An alternative to convolution: Attention

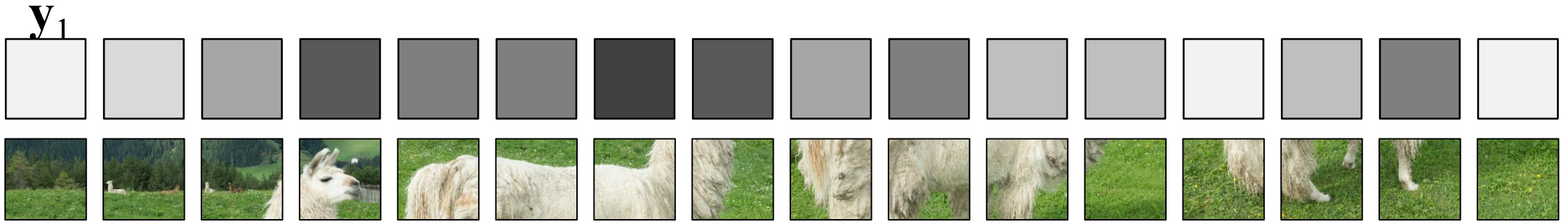
Step 3: For each patch, compare its query vector to all key vectors



An alternative to convolution: Attention

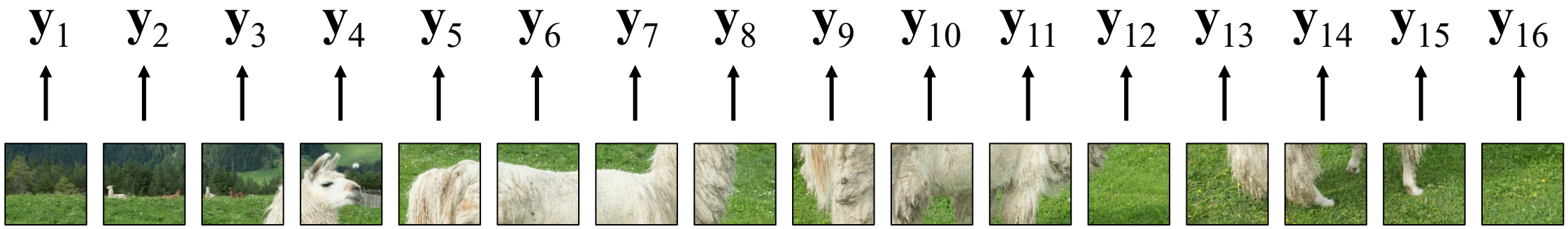
Step 4: Compute weighted sum of **value vectors**

$$\text{New vector } \mathbf{y}_1 = \sum_{i=1}^{n} \text{softmax}\left(\frac{\mathbf{Q}_1 \cdot \mathbf{K}_i}{D}\right) \mathbf{V}_i$$



An alternative to convolution: Attention

Step 5: Repeat for all patches



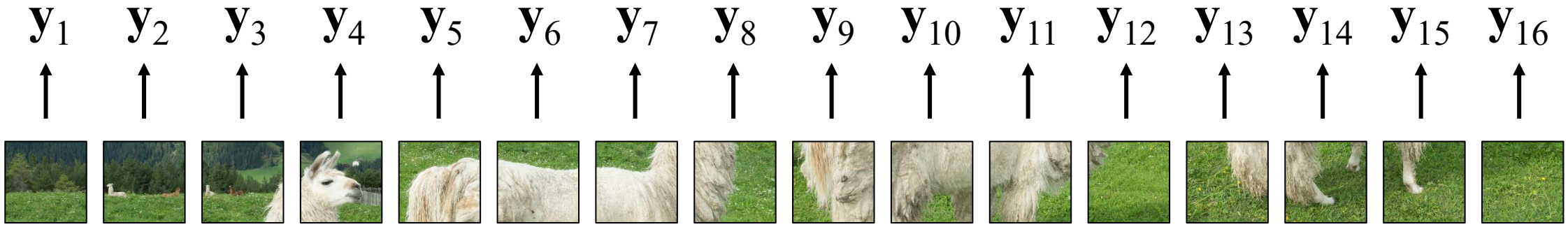
An alternative to convolution: Attention

Result: we've transformed all of the input patches into new vectors, by comparing vectors derived from all pairs of patches

This operation is called **attention** – the network can choose, for each patch, which other patches to attend to (i.e., give high weight to)

Unlike convolution, a patch is allowed to talk to the entire image

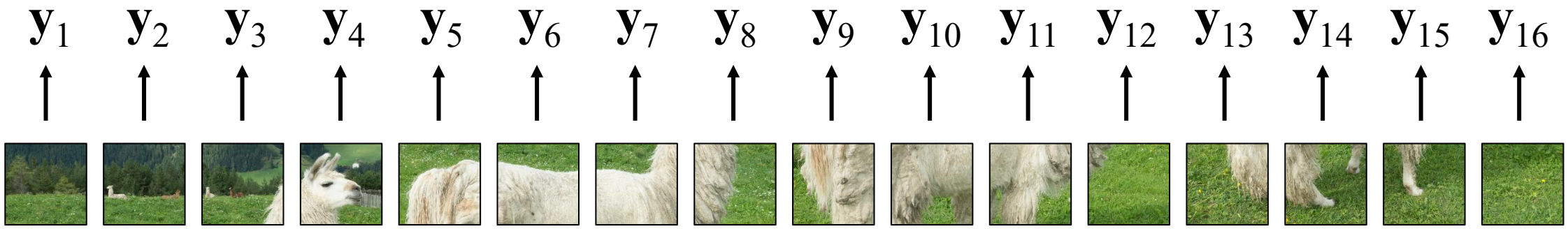
Attention is a *set-to-set* operation – resulting patch representations are covariant under permutation of patch ordering



An alternative to convolution: Attention

Parameters: weight matrices $\mathbf{W_q}$, $\mathbf{W_k}$, $\mathbf{W_v}$ that map input patches to query, key, and value vectors

$$\mathbf{Q}_i = \mathbf{W_q}\mathbf{x}_i, \mathbf{K}_i = \mathbf{W_k}\mathbf{x}_i, \mathbf{V}_i = \mathbf{W_v}\mathbf{x}_i$$



Details

- Rather than working with raw RGB image patches, the patches can themselves be features (e.g., produced by a linear mapping from RGB patches, or the output of a CNN). (These discrete vectors are called *tokens*)
- The feature vectors produced by the attention layer are often passed through an MLP (adding more parameters to the system)
- Each patch can be combined with a *positional encoding* indicating the spatial location of the patch, enabling spatial reasoning
- Instead of single $\mathbf{W}_{q'}$ $\mathbf{W}_{k'}$ $\mathbf{W}_{v}$ weight matrices, multiple linear mappings can be learned for an attention layer, and the

Transformers

- Just like any network layer, we can stack attention layers – the output of one becomes the input to the next – to form a bigger network, called a **transformer**
- Transformers are very large, powerful learners that transcend convolutional networks by representing a larger class of functions

Preprint. Under review.

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy[*,†], **Lucas Beyer**[*], **Alexander Kolesnikov**[*], **Dirk Weissenborn**[*], **Xiaohua Zhai**[*], **Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby**[*,†]

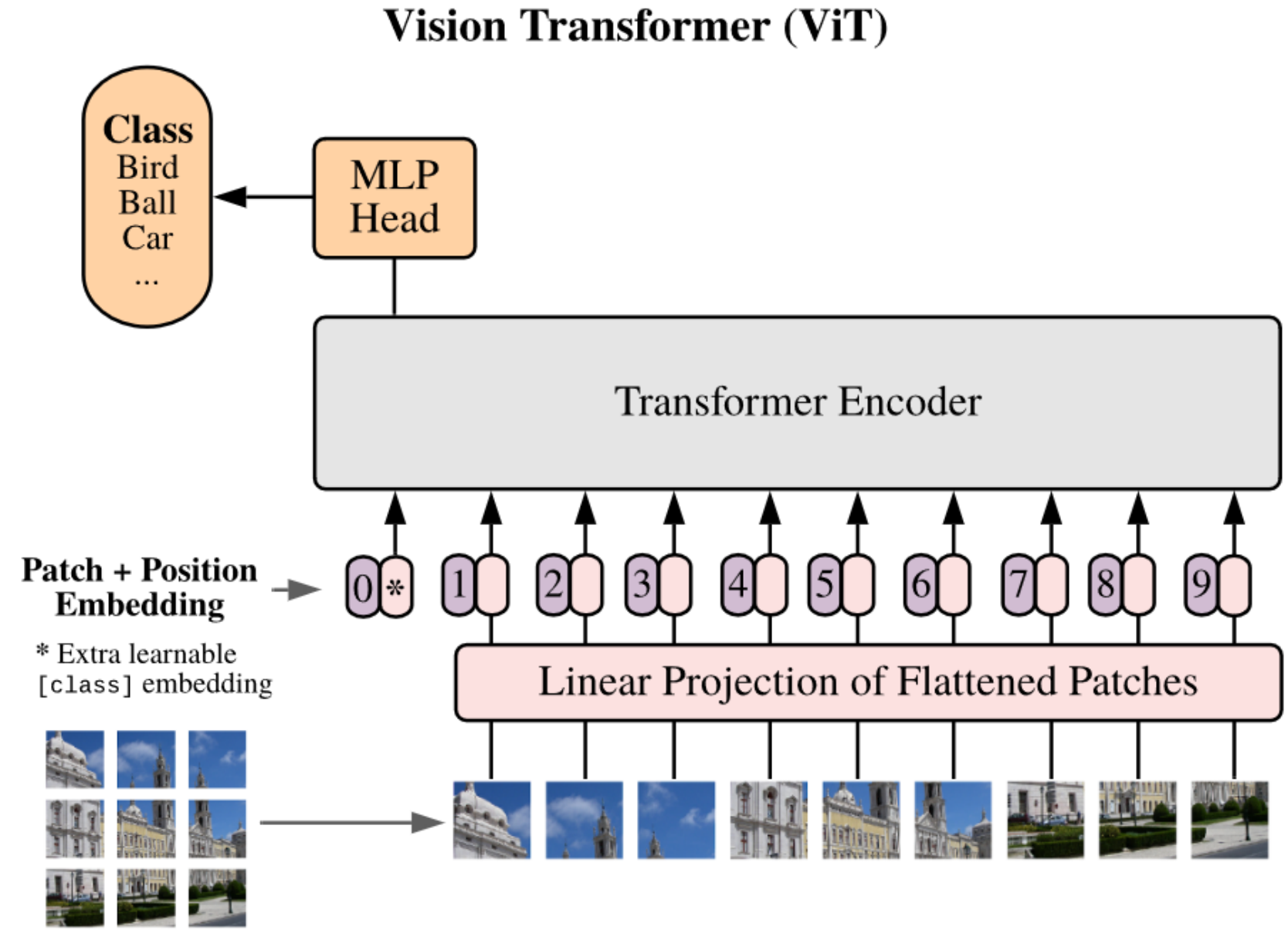
[*]equal technical contribution, [†]equal advising

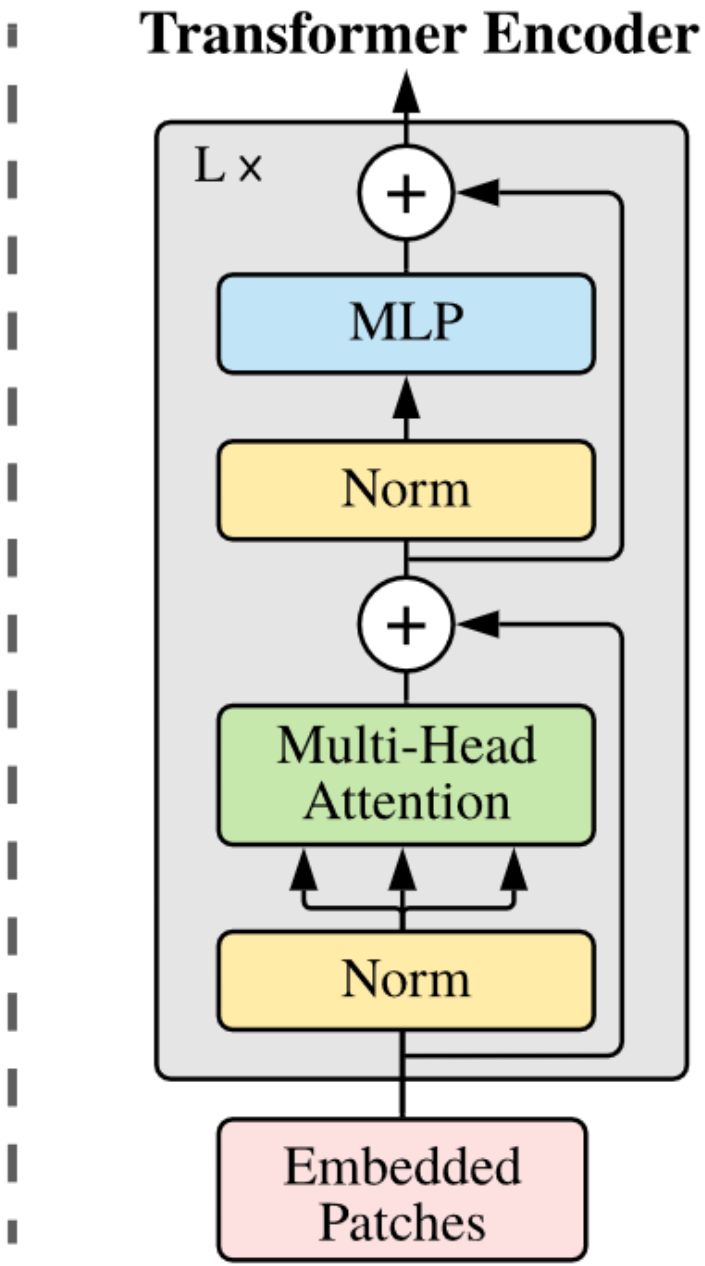
Google Research, Brain Team

{adosovitskiy, neilhoulsby}@google.com

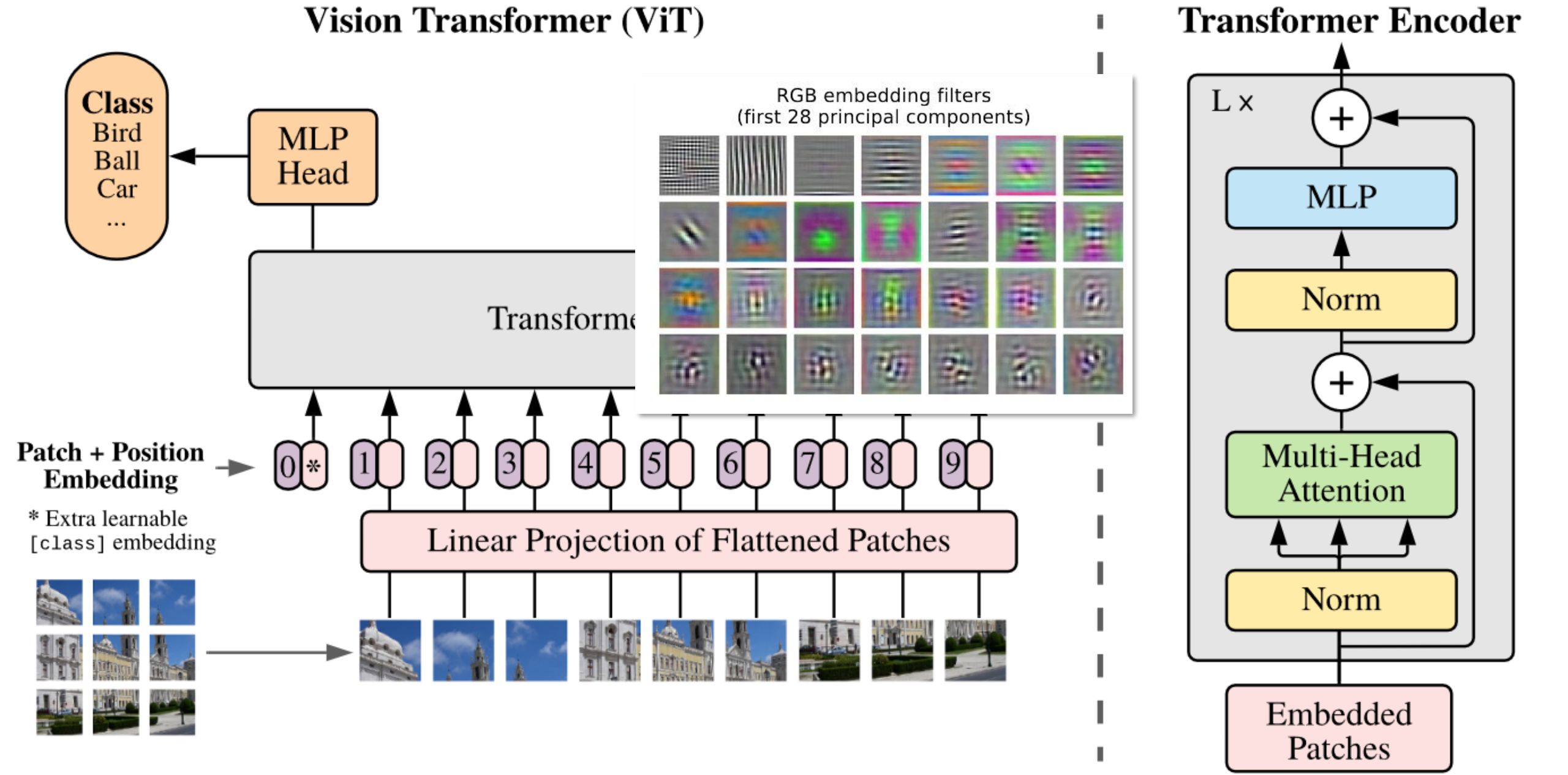
ABSTRACT

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.[1]





Supervised training on 300M labeled images


Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transforme
RGB embedding filters
(first 28 principal components)
Patch + Position Embedding
* Extra learnable [class] embedding
0 * 1 2 3 4 5 6 7 8 9
Linear Projection of Flattened Patches
Transformer Encoder
L x
MLP
Norm
Multi-Head Attention
Norm
Embedded Patches

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants.

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21K (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** $\pm$ 0.04	87.76 $\pm$ 0.03	85.30 $\pm$ 0.02	87.54 $\pm$ 0.02	88.4/88.5*
ImageNet ReaL	**90.72** $\pm$ 0.05	90.54 $\pm$ 0.03	88.62 $\pm$ 0.05	90.54	90.55
CIFAR-10	**99.50** $\pm$ 0.06	99.42 $\pm$ 0.03	99.15 $\pm$ 0.03	99.37 $\pm$ 0.06	–
CIFAR-100	**94.55** $\pm$ 0.04	93.90 $\pm$ 0.05	93.25 $\pm$ 0.05	93.51 $\pm$ 0.08	–
Oxford-IIIT Pets	**97.56** $\pm$ 0.03	97.32 $\pm$ 0.11	94.67 $\pm$ 0.15	96.62 $\pm$ 0.23	–
Oxford Flowers-102	99.68 $\pm$ 0.02	**99.74** $\pm$ 0.00	99.61 $\pm$ 0.02	99.63 $\pm$ 0.03	–
VTAB (19 tasks)	**77.63** $\pm$ 0.23	76.28 $\pm$ 0.46	72.72 $\pm$ 0.21	76.29 $\pm$ 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

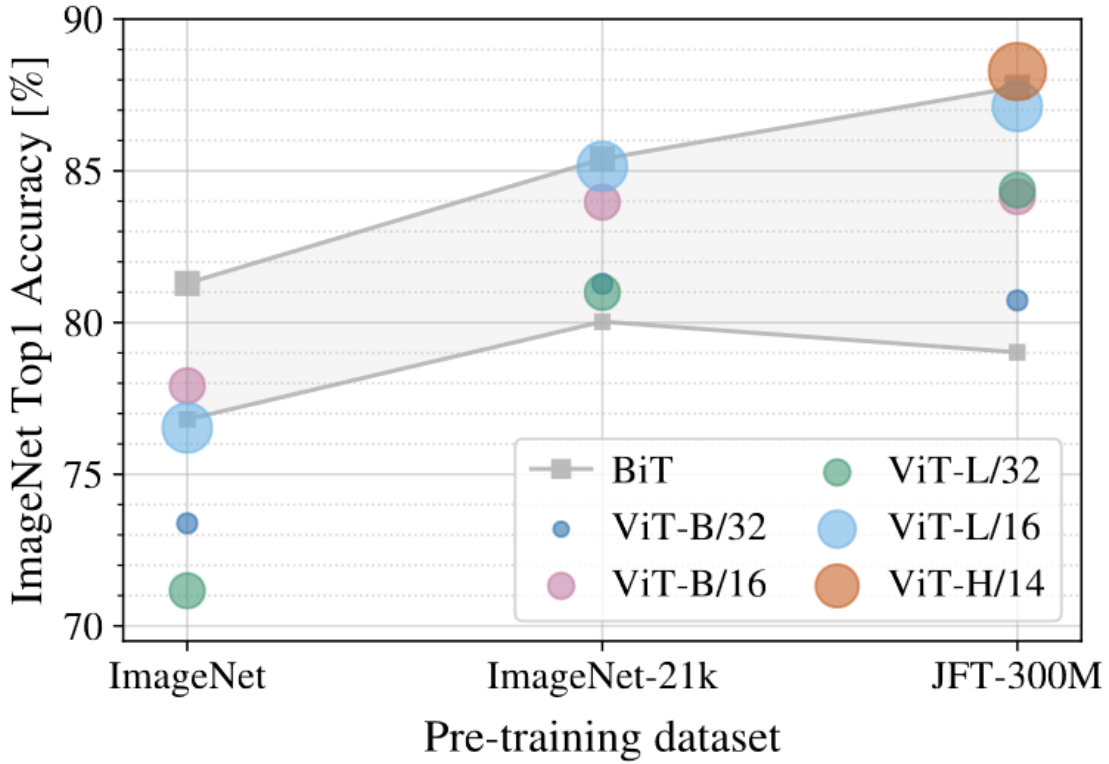


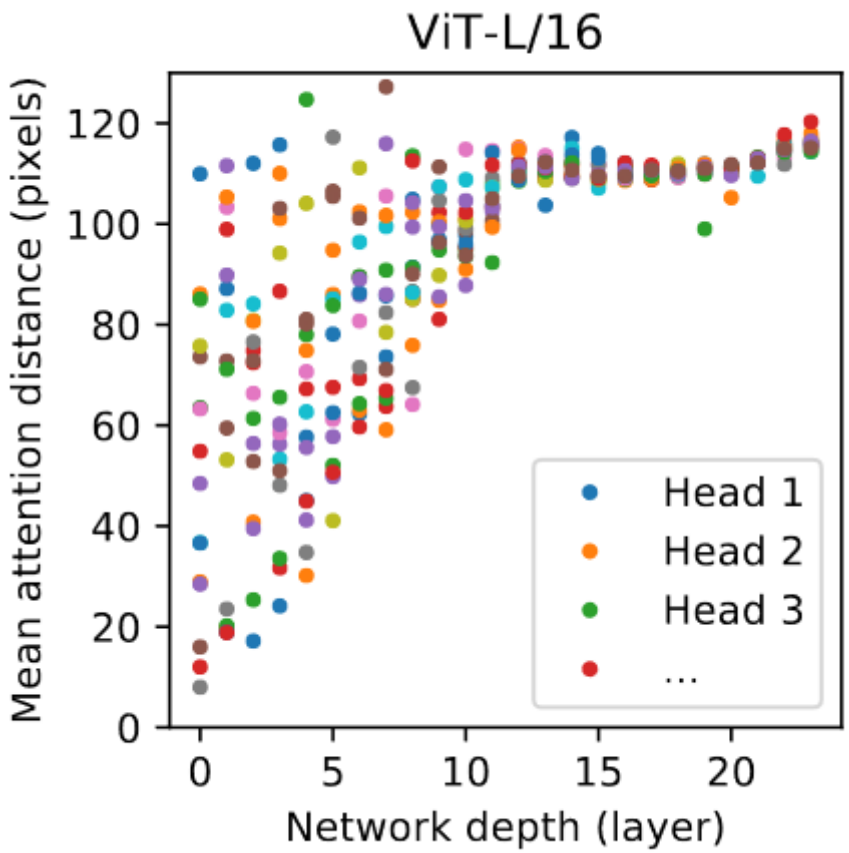
Figure 3: Transfer to ImageNet. While large ViT models perform worse than BiT ResNets (shaded area) when pre-trained on small datasets, they shine when pre-trained on larger datasets. Similarly, larger ViT variants overtake smaller ones as the dataset grows.

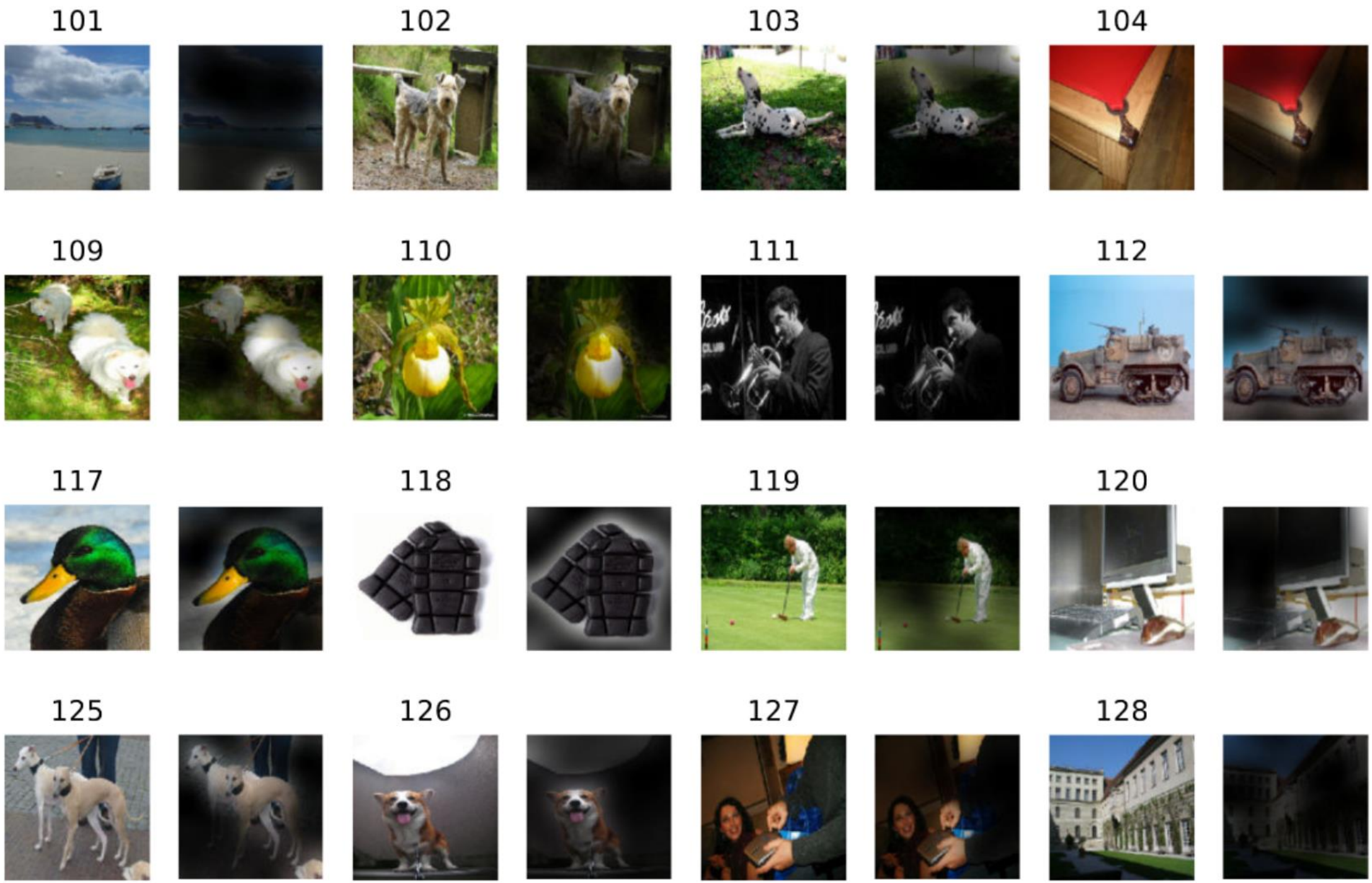
When trained on mid-sized datasets such as ImageNet, such models yield modest accuracies of a few percentage points below ResNets of comparable size. This seemingly discouraging outcome maybe expected: Transformers lack some of the inductive biases inherent to CNNs, such as translation equivariance and locality, and therefore do not generalize well when trained on insufficient amounts of data.

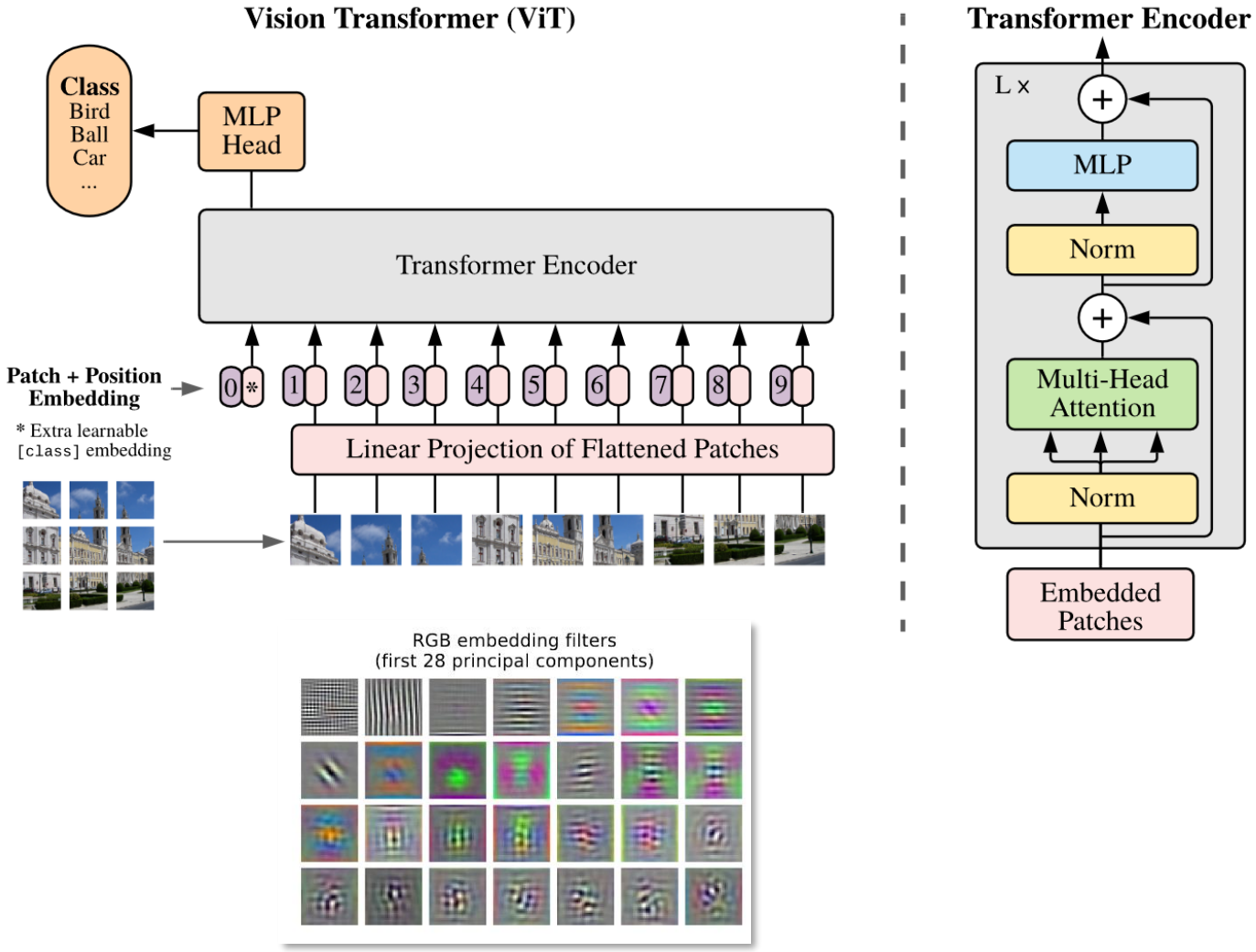
However, the picture changes if the models are trained on larger datasets (14M-300M images). We find that large scale training trumps inductive bias.

Dosovitskiy et al.

https://paperswithcode.com/sota/image-classification-on-imagenet


ViT-L/16
Mean attention distance (pixels)
0
20
40
60
80
100
120
0
5
10
15
20
Network depth (layer)
Head 1
Head 2
Head 3
...

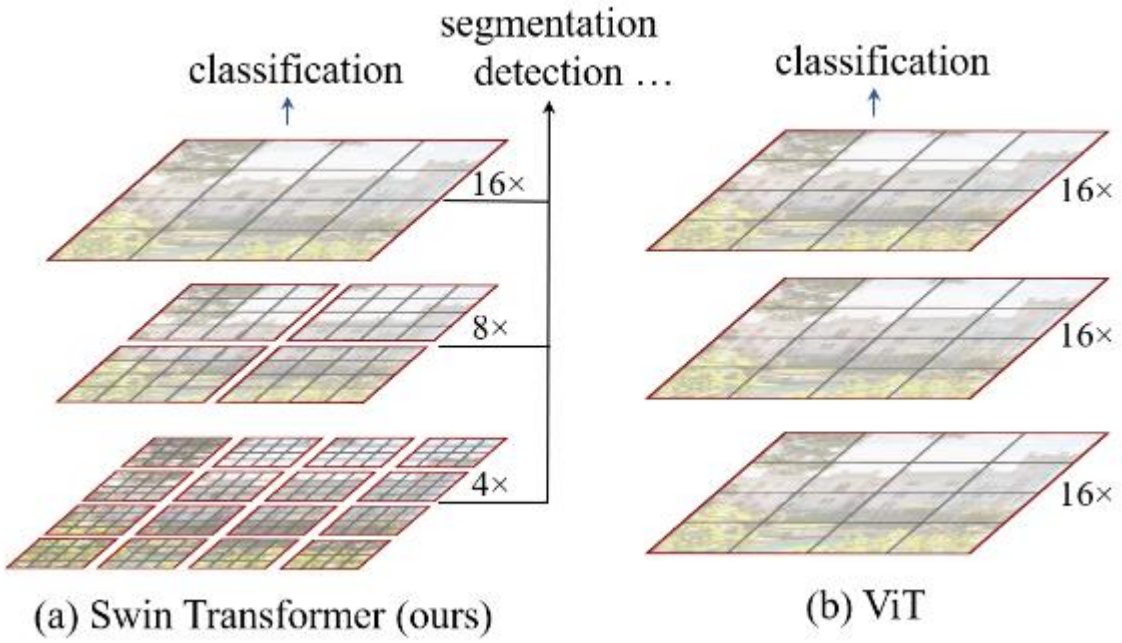

101
102
103
104
109
110
111
112
117
118
119
120
125
126
127
128



- This can’t be ideal, right?

Pos. Emb.	Default/Stem	Every Layer	Every Layer-Shared
No Pos. Emb.	0.61382	N/A	N/A
1-D Pos. Emb.	0.64206	0.63964	0.64292
2-D Pos. Emb.	0.64001	0.64046	0.64022
Rel. Pos. Emb.	0.64032	N/A	N/A

Table 8: Results of the ablation study on positional embeddings with ViT-B/16 model evaluated on ImageNet 5-shot linear.



Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, Baining Guo

Swin Transformer outperforms “vanilla” ViT by…

- Reintroducing the useful inductive biases of Convolutional Networks
  - Hierarchical feature maps
    - Lose spatial resolution as you go deeper in the network, gain channel depth
  - *Local* self attention
    - E.g. a 7x7 window of patches instead of global attention
    - Receptive field expands through the namesake window shifting from layer to layer
  - *Relative* positional encoding

(a) Regular ImageNet-1K trained models

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
RegNetY-4G [48]	224^2	21M	4.0G	1156.7	80.0
RegNetY-8G [48]	224^2	39M	8.0G	591.6	81.7
RegNetY-16G [48]	224^2	84M	16.0G	334.7	82.9
EffNet-B3 [58]	300^2	12M	1.8G	732.1	81.6
EffNet-B4 [58]	380^2	19M	4.2G	349.4	82.9
EffNet-B5 [58]	456^2	30M	9.9G	169.1	83.6
EffNet-B6 [58]	528^2	43M	19.0G	96.9	84.0
EffNet-B7 [58]	600^2	66M	37.0G	55.1	84.3
ViT-B/16 [20]	384^2	86M	55.4G	85.9	77.9
ViT-L/16 [20]	384^2	307M	190.7G	27.3	76.5
DeiT-S [63]	224^2	22M	4.6G	940.4	79.8
DeiT-B [63]	224^2	86M	17.5G	292.3	81.8
DeiT-B [63]	384^2	86M	55.4G	85.9	83.1
Swin-T	224^2	29M	4.5G	755.2	81.3
Swin-S	224^2	50M	8.7G	436.9	83.0
Swin-B	224^2	88M	15.4G	278.1	83.5
Swin-B	384^2	88M	47.0G	84.7	84.5

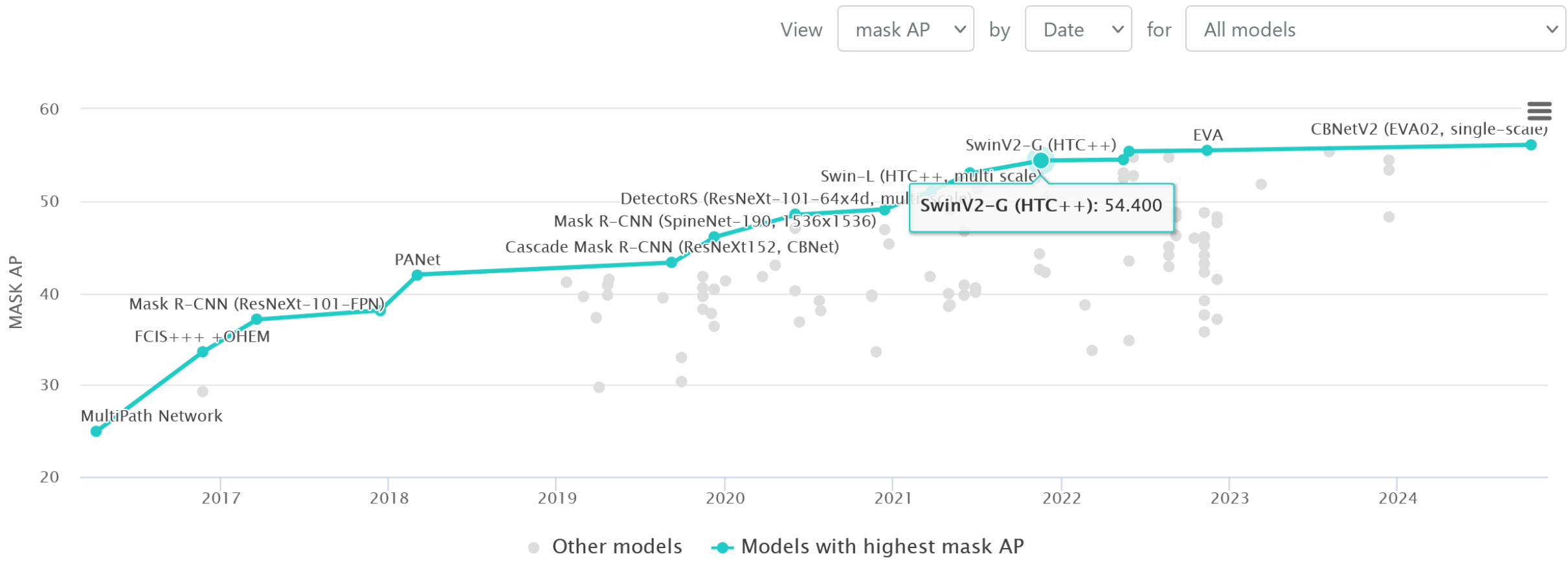
(b) ImageNet-22K pre-trained models

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
R-101x3 [38]	384^2	388M	204.6G	-	84.4
R-152x4 [38]	480^2	937M	840.5G	-	85.4
ViT-B/16 [20]	384^2	86M	55.4G	85.9	84.0
ViT-L/16 [20]	384^2	307M	190.7G	27.3	85.2
Swin-B	224^2	88M	15.4G	278.1	85.2
Swin-B	384^2	88M	47.0G	84.7	86.4
Swin-L	384^2	197M	103.9G	42.1	87.3

Table 1. Comparison of different backbones on ImageNet-1K classification. Throughput is measured using the GitHub repository of [68] and a V100 GPU, following [63].

Instance Segmentation on COCO test-dev

Leaderboard Dataset



A ConvNet for the 2020s

Zhuang Liu[1,2*] Hanzi Mao[1] Chao-Yuan Wu[1] Christoph Feichtenhofer[1] Trevor Darrell[2] Saining Xie[1†]

[1]Facebook AI Research (FAIR) [2]UC Berkeley

Code: https://github.com/facebookresearch/ConvNeXt

Abstract

The "Roaring 20s" of visual recognition began with the introduction of Vision Transformers (ViTs), which quickly superseded ConvNets as the state-of-the-art image classification model. A vanilla ViT, on the other hand, faces difficulties when applied to general computer vision tasks such as object detection and semantic segmentation. It is the hierarchical Transformers (e.g., Swin Transformers) that reintroduced several ConvNet priors, making Transformers practically viable as a generic vision backbone and demonstrating remarkable performance on a wide variety of vision tasks. However, the effectiveness of such hybrid approaches is still largely credited to the intrinsic superiority of Transformers, rather than the inherent inductive biases of convolutions. In this work, we reexamine the design spaces and test the limits of what a pure ConvNet can achieve. We gradually "modernize" a standard ResNet toward the design of a vision Transformer, and discover several key components that contribute to the performance difference along the way. The outcome of this exploration is a family of pure ConvNet models dubbed ConvNeXt. Constructed entirely from standard ConvNet modules, ConvNeXts compete favorably with Transformers in terms of accuracy and scalability, achieving 87.8% ImageNet top-1 accuracy and outperforming Swin Transformers on COCO detection and ADE20K segmentation, while maintaining the simplicity and efficiency of standard ConvNets.

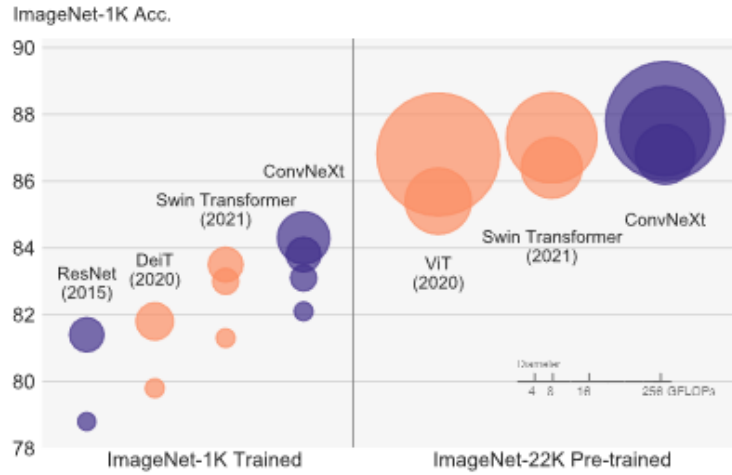


Figure 1. **ImageNet-1K classification** results for • ConvNets and ○ vision Transformers. Each bubble's area is proportional to FLOPs of a variant in a model family. ImageNet-1K/22K models here take 224^2/384^2 images respectively. ResNet and ViT results were obtained with improved training procedures over the original papers. We demonstrate that a standard ConvNet model can achieve the same level of scalability as hierarchical vision Transformers while being much simpler in design.

visual feature learning. The introduction of AlexNet [40] precipitated the "ImageNet moment" [59], ushering in a new era of computer vision. The field has since evolved at a rapid speed. Representative ConvNets like VGGNet [64], Inceptions [68], ResNe(X)t [28, 87], DenseNet [36], MobileNet [34], EfficientNet [71] and RegNet [54] focused on different aspects of accuracy, efficiency and scalability, and popularized many useful design principles.

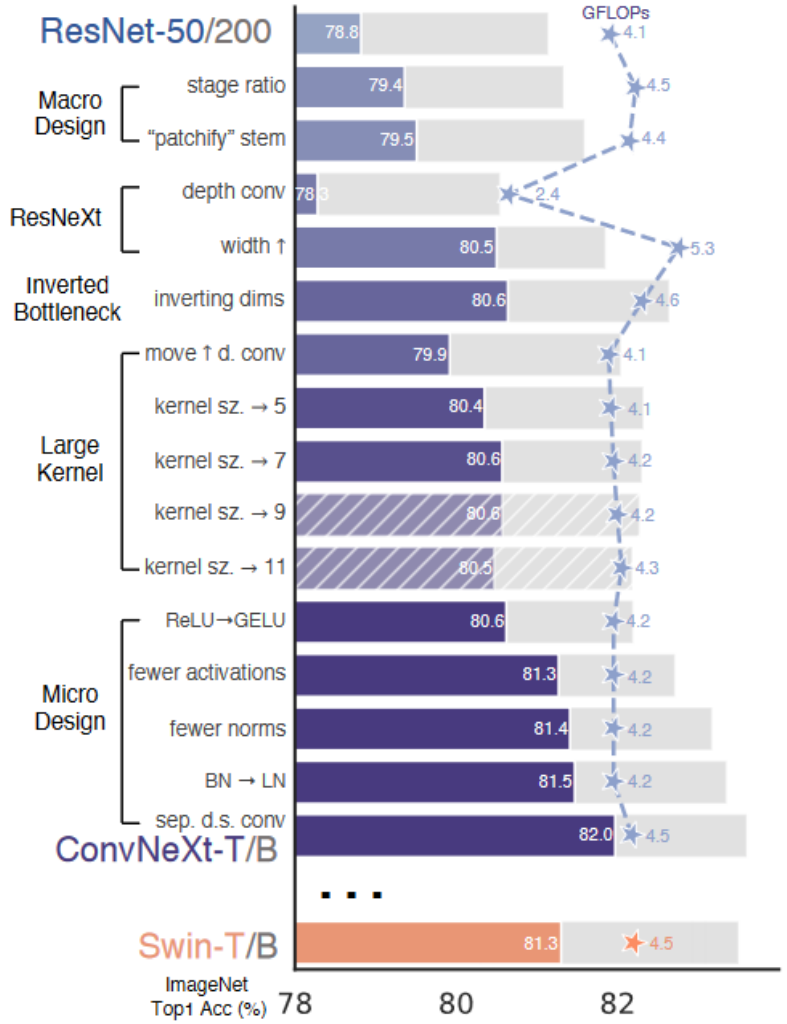


Figure 2. We modernize a standard ConvNet (ResNet) towards the design of a hierarchical vision Transformer (Swin), without introducing any attention-based modules. The foreground bars are model accuracies in the ResNet-50/Swin-T FLOP regime; results for the ResNet-200/Swin-B regime are shown with the gray bars. A hatched bar means the modification is not adopted. Detailed results for both regimes are in the appendix. Many Transformer architectural choices can be incorporated in a ConvNet, and they lead to increasingly better performance. In the end, our pure ConvNet model, named ConvNeXt, can outperform the Swin Transformer.

backbone	FLOPs	FPS	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
Mask-RCNN 3× schedule								
○ Swin-T	267G	23.1	46.0	68.1	50.3	41.6	65.1	44.9
● ConvNeXt-T	262G	25.6	**46.2**	67.9	50.8	**41.7**	65.0	44.9
Cascade Mask-RCNN 3× schedule								
● ResNet-50	739G	16.2	46.3	64.3	50.5	40.1	61.7	43.4
● X101-32	819G	13.8	48.1	66.5	52.4	41.6	63.9	45.2
● X101-64	972G	12.6	48.3	66.4	52.3	41.7	64.0	45.1
○ Swin-T	745G	12.2	50.4	69.2	54.7	43.7	66.6	47.3
● ConvNeXt-T	741G	13.5	**50.4**	69.1	54.8	**43.7**	66.5	47.3
○ Swin-S	838G	11.4	51.9	70.7	56.3	45.0	68.2	48.8
● ConvNeXt-S	827G	12.0	**51.9**	70.8	56.5	**45.0**	68.4	49.1
○ Swin-B	982G	10.7	51.9	70.5	56.4	45.0	68.1	48.9
● ConvNeXt-B	964G	11.4	**52.7**	71.3	57.2	**45.6**	68.9	49.5
○ Swin-B‡	982G	10.7	53.0	71.8	57.5	45.8	69.4	49.7
● ConvNeXt-B‡	964G	11.5	**54.0**	73.1	58.8	**46.9**	70.6	51.3
○ Swin-L‡	1382G	9.2	53.9	72.4	58.8	46.7	70.1	50.8
● ConvNeXt-L‡	1354G	10.0	**54.8**	73.8	59.8	**47.6**	71.3	51.7
● ConvNeXt-XL‡	1898G	8.6	**55.2**	74.2	59.9	**47.7**	71.6	52.2

Table 3. **COCO object detection and segmentation results** using Mask-RCNN and Cascade Mask-RCNN. ‡ indicates that the model is pre-trained on ImageNet-22K. ImageNet-1K pre-trained Swin results are from their Github repository [3]. AP numbers of the ResNet-50 and X101 models are from [45]. We measure FPS on an A100 GPU. FLOPs are calculated with image size (1280, 800).

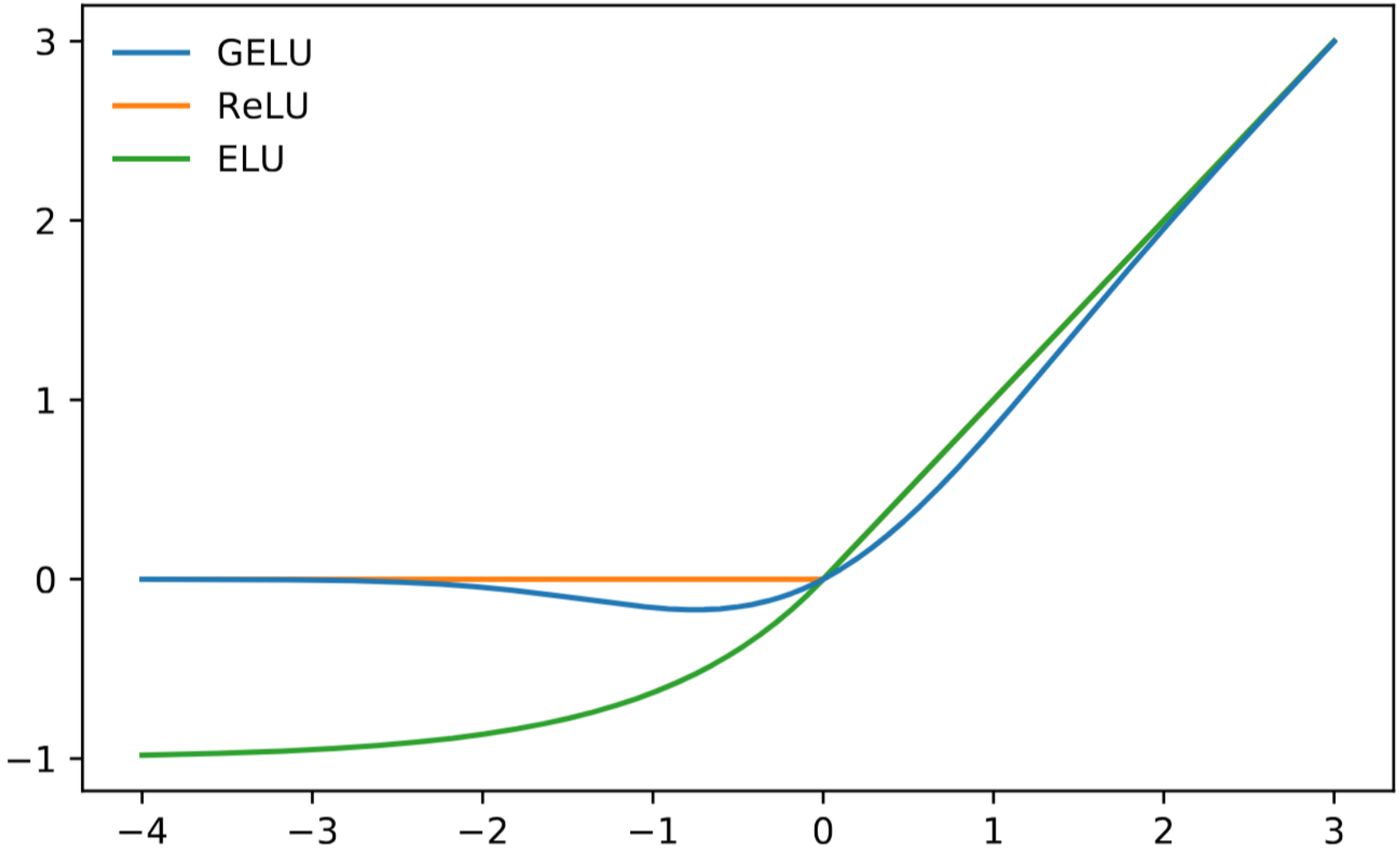


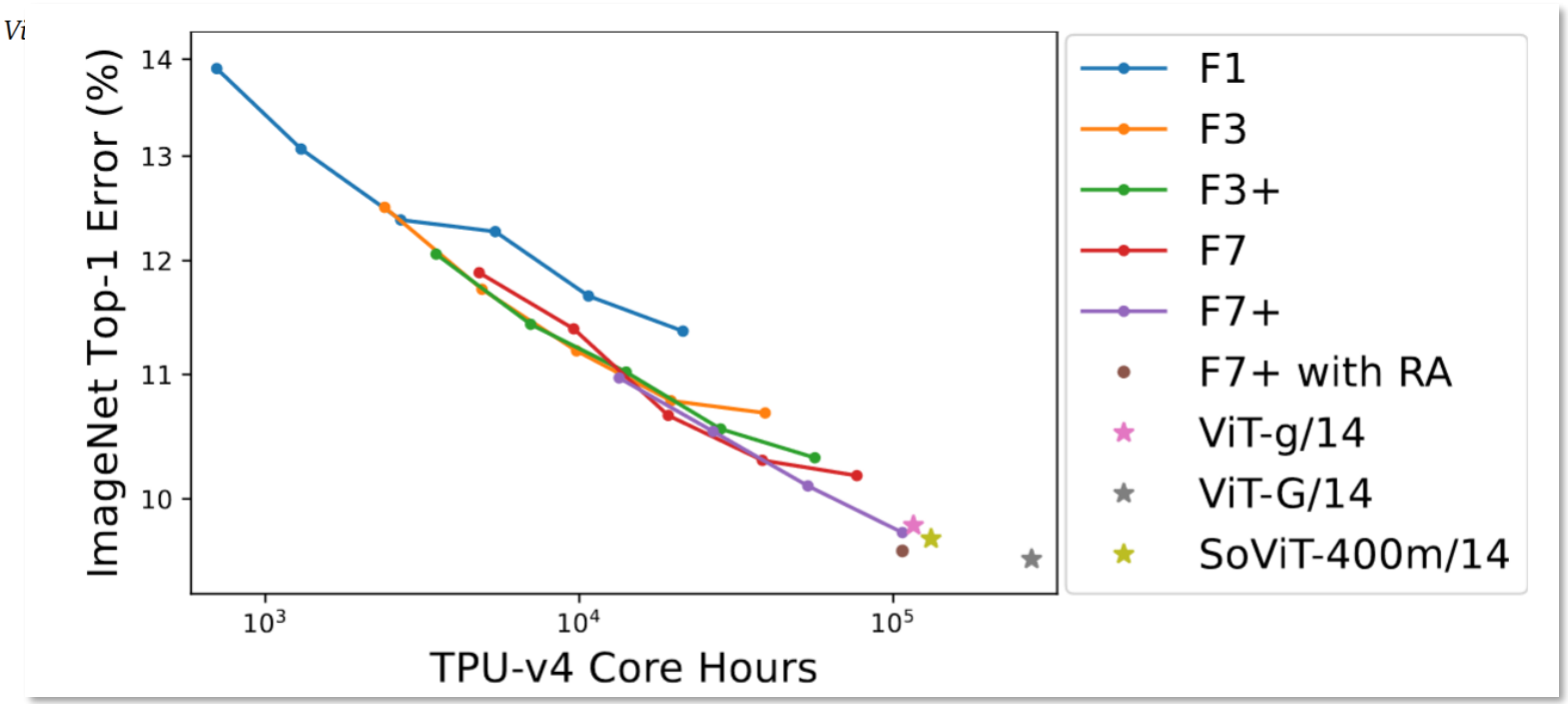
Figure 1: The GELU ($\mu = 0, \sigma = 1$), ReLU, and ELU ($\alpha = 1$).

ConvNets Match Vision Transformers at Scale

Samuel L Smith[1], Andrew Brock[1], Leonard Berrada[1] and Soham De[1]
[1]Google DeepMind

Many researchers believe that ConvNets perform well on small or moderately sized datasets, but are not competitive with Vision Transformers when given access to datasets on the web-scale. We challenge this belief by evaluating a performant ConvNet architecture pre-trained on JFT-4B, a large labelled dataset of images often used for training foundation models. We consider pre-training compute budgets between 0.4k and 110k TPU-v4 core compute hours, and train a series of networks of increasing depth and width from the NFNet model family. We observe a log-log scaling law between held out loss and compute budget. After fine-tuning on ImageNet, NFNets match the reported performance of Vision Transformers with comparable compute budgets. Our strongest fine-tuned model achieves a Top-1 accuracy of 90.4%.

Keywords: ConvNets, CNN, Convolution, Transformer, Vi



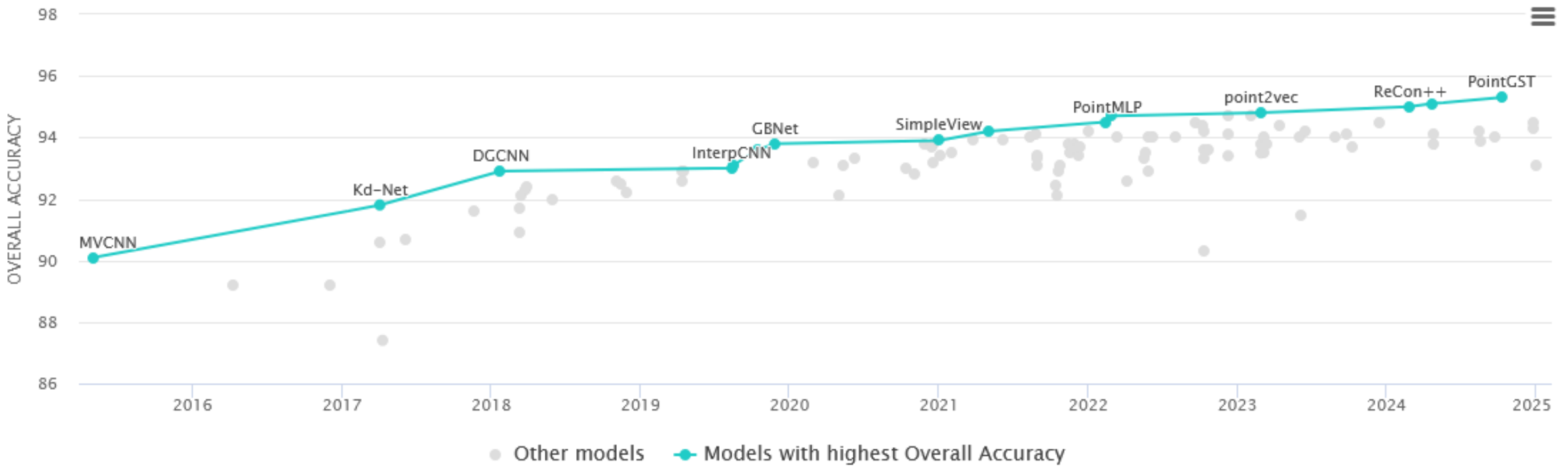
Outline

- Context and Receptive Field
- Going Beyond Convolutions in…
  - Text
  - Images
  - Point Clouds
- Notable Transformer “Foundation” models
  - CLIP-like models
  - DINO models

3D Point Cloud Classification on ModelNet40

Leaderboard Dataset

View Overall Accuracy by Date for All models



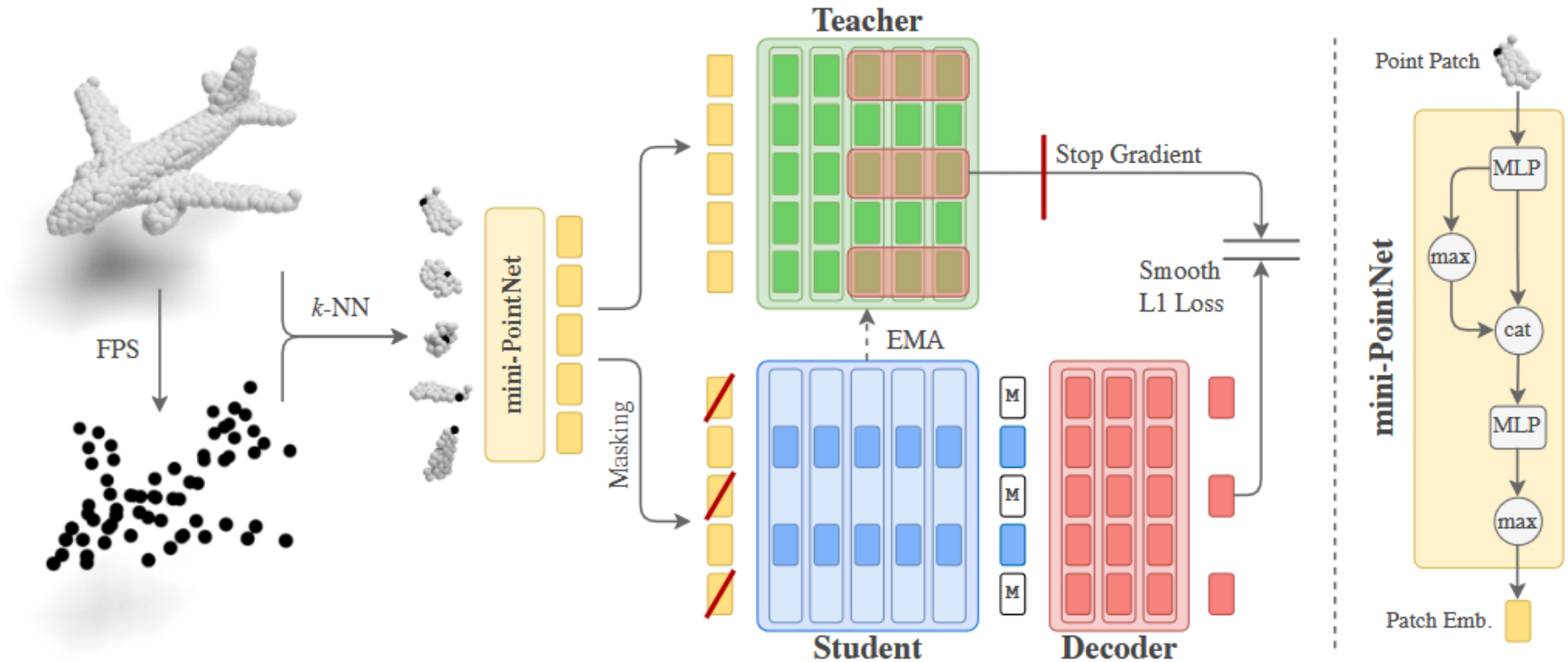


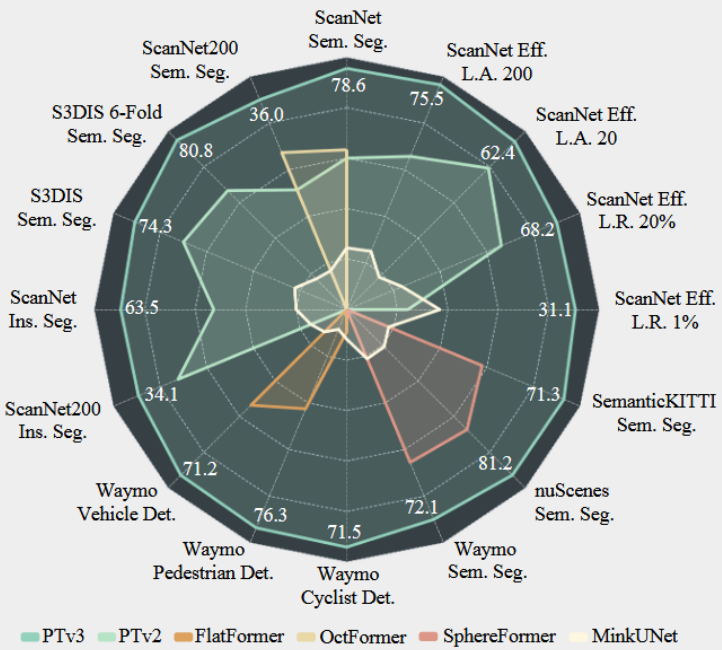
Fig. 2: Point2Vec pre-training. Our model divides the input point cloud into point patches using farthest point sampling (FPS) and k-NN aggregation. We obtain patch embeddings by applying a mini-PointNet ▢ to each point patch (*right*). The teacher Transformer encoder ▢ infers a contextualized representation for all patch embeddings which, after normalization and averaging over the last K Transformer layers, serve as training targets. The student's input is a masked view on the input data, *i.e.* we randomly mask out a ratio of patch embeddings and only pass the remaining embeddings into the student Transformer encoder ▢. After applying a shallow decoder ▢ on the outputs of the student, padded with learned mask embeddings [M], we train the student and decoder to predict the latent teacher representation of the patch embeddings.

Point Transformer V3: Simpler, Faster, Stronger

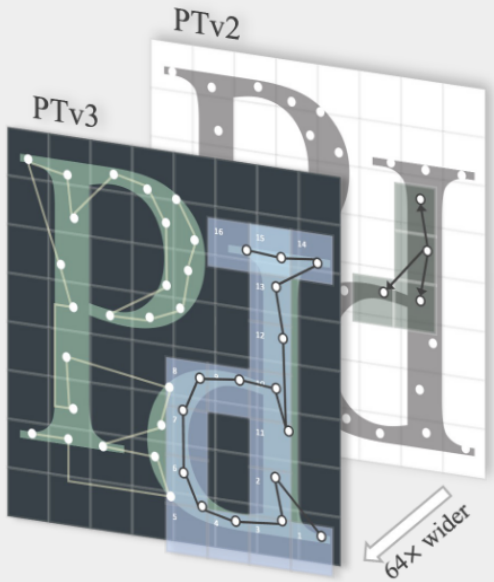
Xiaoyang Wu[1,2] Li Jiang[3] Peng-Shuai Wang[4]
Zhijian Liu[5] Xihui Liu[1] Yu Qiao[2] Wanli Ouyang[2] Tong He[2*] Hengshuang Zhao[1*]

[1]HKU [2]SH AI Lab [3]CUHK(SZ) [4]PKU [5]MIT

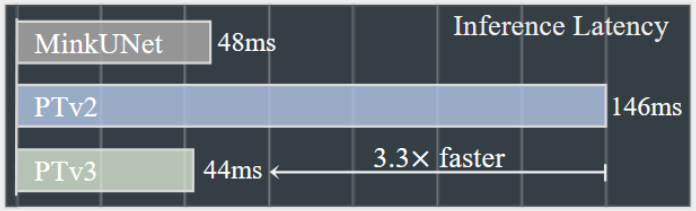
https://github.com/Pointcept/PointTransformerV3



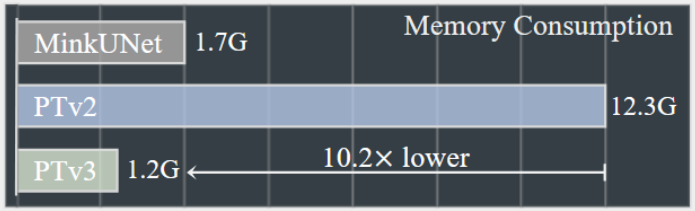
Stronger Performance



Wider Receptive Field



Faster Speed



Lower Memory Consumption

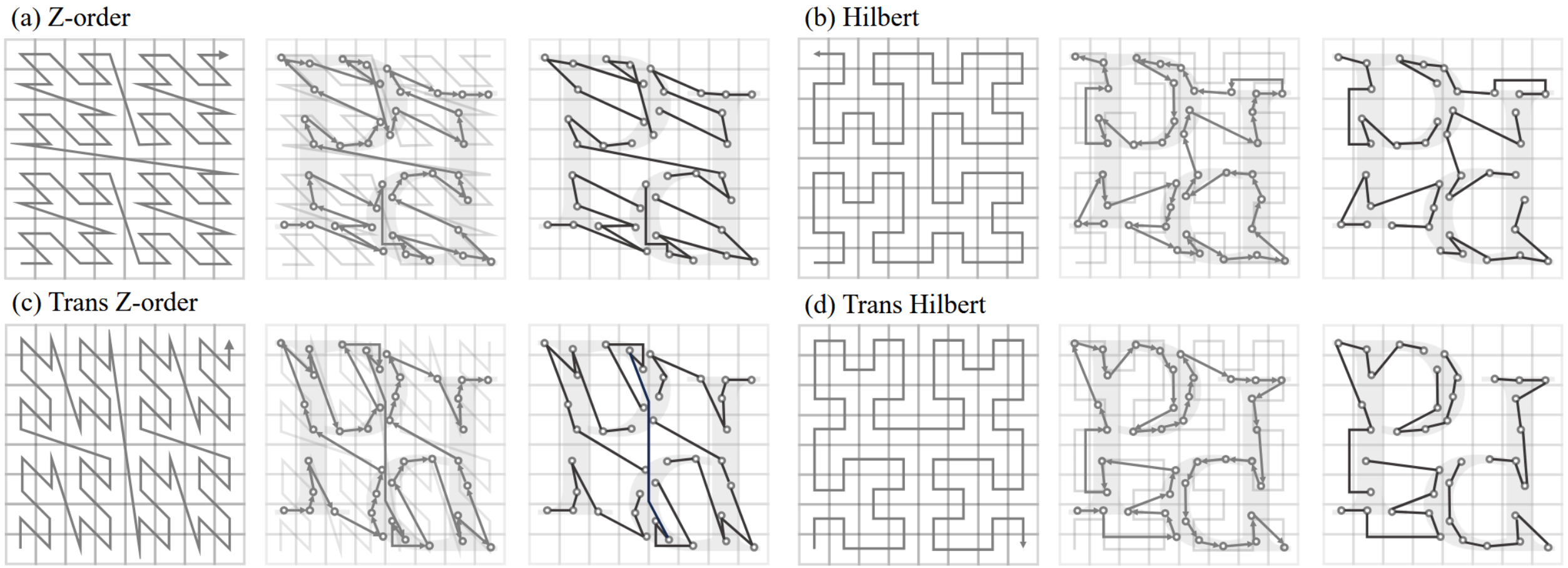


Figure 3. **Point cloud serialization.** We show the four patterns of serialization with a triplet visualization. For each triplet, we show the space-filling curve for serialization (left), point cloud serialization var sorting order within the space-filling curve (middle), and grouped patches of the serialized point cloud for local attention (right). Shifting across the four serialization patterns allows the attention mechanism to capture various spatial relationships and contexts, leading to an improvement in model accuracy and generalization capacity.

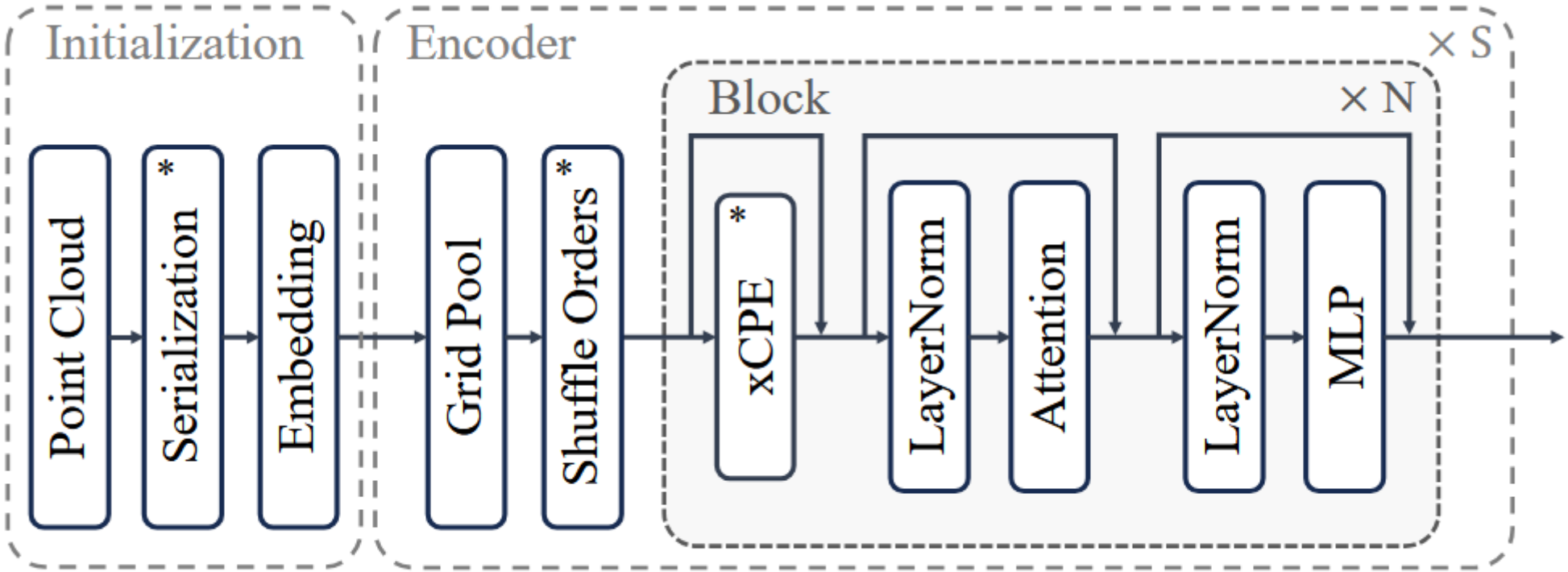


Figure 6. **Overall architecture.**

Patterns	S.O.	+ S.D.	+ S.P.	+ Shuffle O.
Z	74.3 54ms	75.5 89ms	75.8 86ms	74.3 54ms
Z + TZ	76.0 55ms	76.3 92ms	76.1 89ms	76.9 55ms
H + TH	76.2 60ms	76.1 98ms	76.2 94ms	76.8 60ms
Z + TZ + H + TH	76.5 61ms	76.8 99ms	76.6 97ms	**77.3** 61ms

Table 2. **Serialization patterns and patch interaction.** The first column indicates serialization patterns: Z for Z-order, TZ for Trans Z-order, H for Hilbert, and TH for Trans Hilbert. In the first row, S.O. represents Shift Order, which is the default setting also applied to other interaction strategies. S.D. stands for Shift Dilation, and S.P. signifies Shift Patch.

PE	APE	RPE	cRPE	CPE	xCPE
Perf. (%)	72.1 50ms	75.9 72ms	76.8 101ms	76.6 58ms	**77.3** 61ms

Table 3. **Positional encoding.** We compare the proposed CPE+ with APE, RPE, cRPE, and CPE. RPE and CPE are discussed in OctFormer [83], while cRPE is deployed by Swin3D [101].

P.S.	16	32	64	128	256	1024	4096
Perf. (%)	75.0	75.6	76.3	76.6	76.8	**77.3**	77.1
Std. Dev.	**0.15**	0.22	0.31	0.36	0.28	0.22	0.39

Table 4. **Patch size.** Leveraging the inherent simplicity and efficiency of our approach, we expand the receptive field of attention well beyond the conventional scope, surpassing sizes used in previous works such as PTv2 [90], which adopts a size of 16, and OctFormer [83], which uses 24.

Outline

- Context and Receptive Field
- Going Beyond Convolutions in…
  - Text
  - Images
  - Point Clouds
- Notable Transformer “Foundation” models
  - CLIP-like models
  - DINO models

CLIP

Learning Transferable Visual Models From Natural Language Supervision

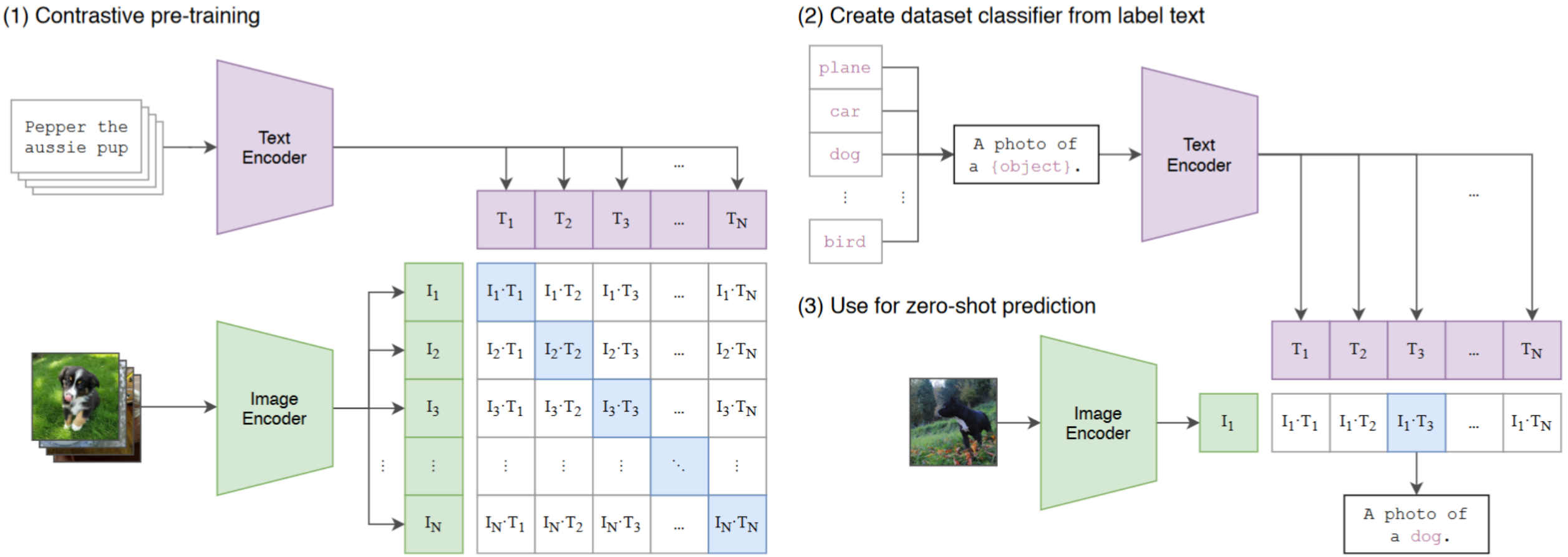
Alec Radford [*1] Jong Wook Kim [*1] Chris Hallacy [1] Aditya Ramesh [1] Gabriel Goh [1] Sandhini Agarwal [1]
Girish Sastry [1] Amanda Askell [1] Pamela Mishkin [1] Jack Clark [1] Gretchen Krueger [1] Ilya Sutskever [1]

Abstract

State-of-the-art computer vision systems are trained to predict a fixed set of predetermined object categories. This restricted form of supervision limits their generality and usability since additional labeled data is needed to specify any other visual concept. Learning directly from raw text about images is a promising alternative which leverages a much broader source of supervision. We demonstrate that the simple pre-training task of predicting which caption goes with which image is an efficient and scalable way to learn SOTA image representations from scratch on a dataset of 400 million (image, text) pairs collected from the internet. After pre-training, natural language is used to reference learned visual concepts (or describe new ones) enabling zero-shot transfer of the model to downstream tasks. We study the performance of this approach by benchmarking on over 30 different existing computer vision datasets, spanning tasks such as OCR, ac-

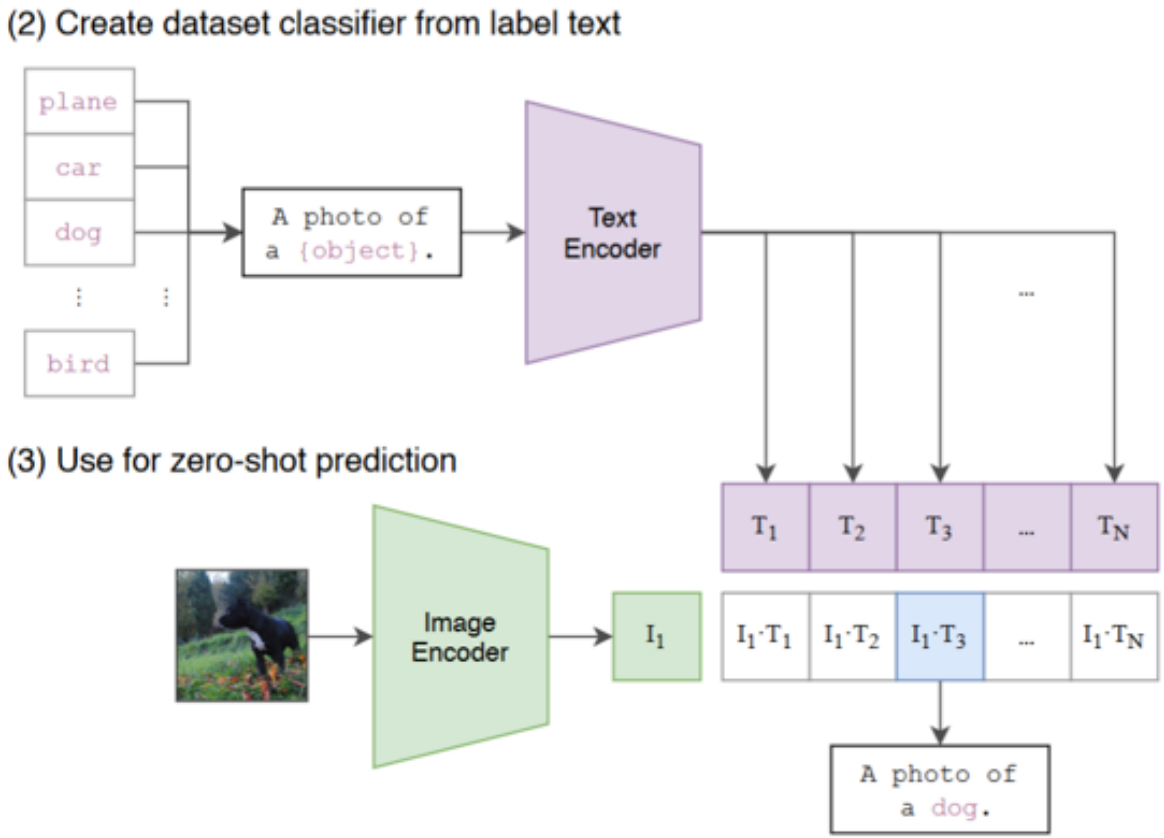
Task-agnostic objectives such as autoregressive and masked language modeling have scaled across many orders of magnitude in compute, model capacity, and data, steadily improving capabilities. The development of "text-to-text" as a standardized input-output interface (McCann et al., 2018; Radford et al., 2019; Raffel et al., 2019) has enabled task-agnostic architectures to zero-shot transfer to downstream datasets removing the need for specialized output heads or dataset specific customization. Flagship systems like GPT-3 (Brown et al., 2020) are now competitive across many tasks with bespoke models while requiring little to no dataset specific training data.

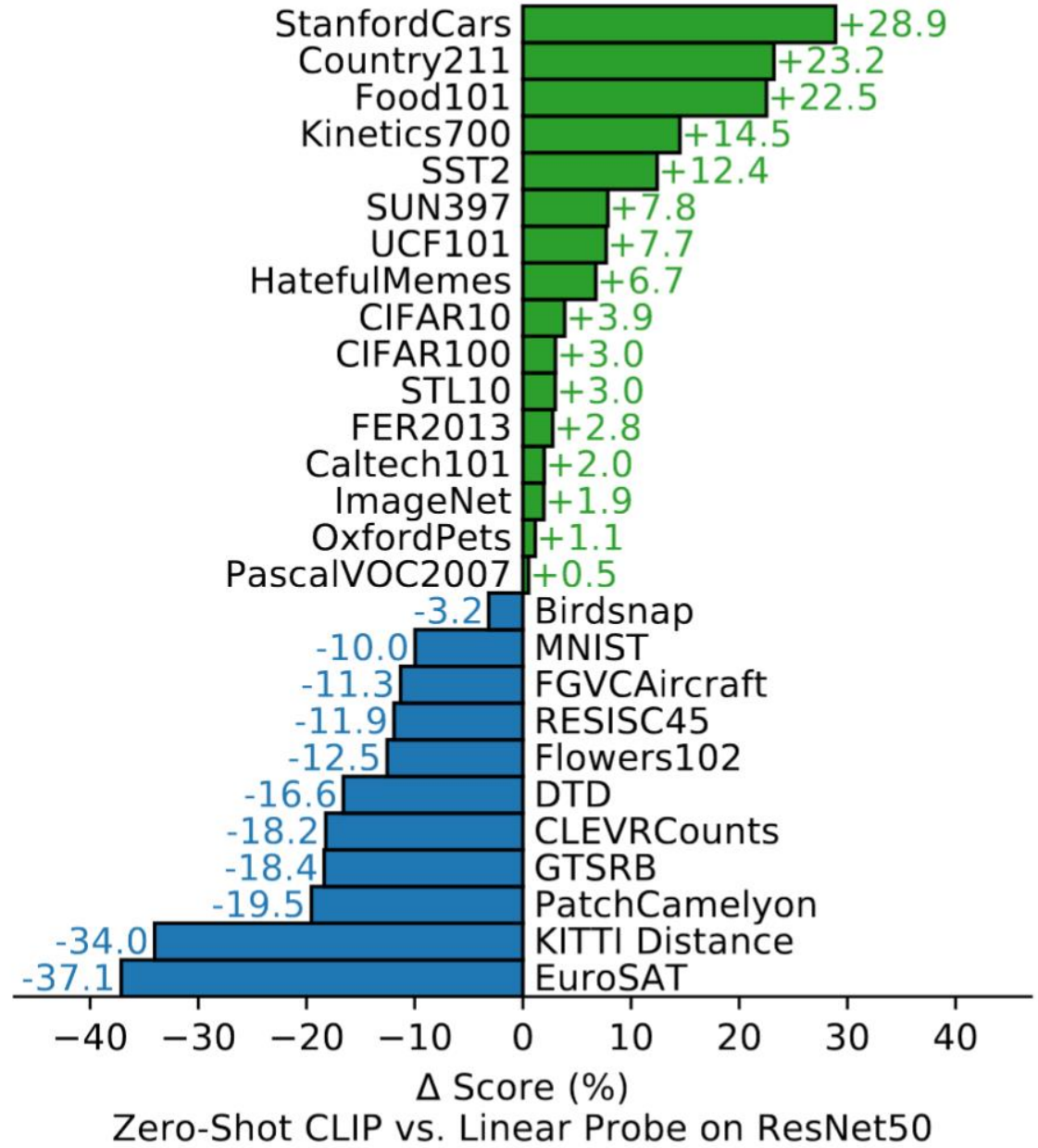
These results suggest that the aggregate supervision accessible to modern pre-training methods within web-scale collections of text surpasses that of high-quality crowd-labeled NLP datasets. However, in other fields such as computer vision it is still standard practice to pre-train models on crowd-labeled datasets such as ImageNet (Deng et al., 2009). Could scalable pre-training methods which learn directly from web text result in a similar breakthrough in computer vision? Prior work is encouraging.

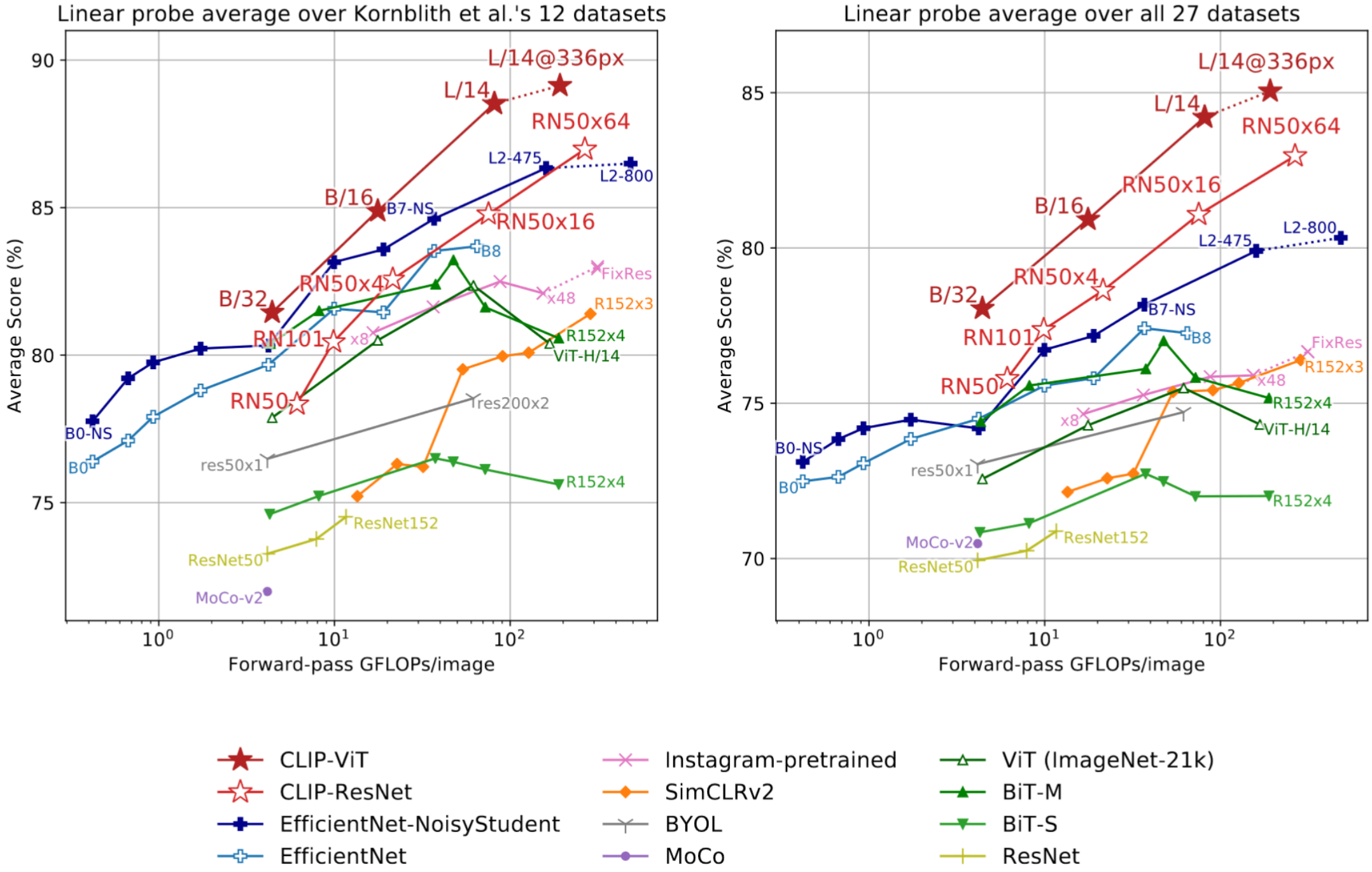


- “weakly supervised” contrastive learning on 400M text / image pairs
- ViT architecture trained from scratch

Dataset	Classes	Train size	Test size	Evaluation metric
Food-101	102	75,750	25,250	accuracy
CIFAR-10	10	50,000	10,000	accuracy
CIFAR-100	100	50,000	10,000	accuracy
Birdsnap	500	42,283	2,149	accuracy
SUN397	397	19,850	19,850	accuracy
Stanford Cars	196	8,144	8,041	accuracy
FGVC Aircraft	100	6,667	3,333	mean per class
Pascal VOC 2007 Classification	20	5,011	4,952	11-point mAP
Describable Textures	47	3,760	1,880	accuracy
Oxford-IIIT Pets	37	3,680	3,669	mean per class
Caltech-101	102	3,060	6,085	mean-per-class
Oxford Flowers 102	102	2,040	6,149	mean per class
MNIST	10	60,000	10,000	accuracy
Facial Emotion Recognition 2013	8	32,140	3,574	accuracy
STL-10	10	1000	8000	accuracy
EuroSAT	10	10,000	5,000	accuracy
RESISC45	45	3,150	25,200	accuracy
GTSRB	43	26,640	12,630	accuracy
KITTI	4	6,770	711	accuracy
Country211	211	43,200	21,100	accuracy
PatchCamelyon	2	294,912	32,768	accuracy
UCF101	101	9,537	1,794	accuracy
Kinetics700	700	494,801	31,669	mean(top1, top5)
CLEVR Counts	8	2,000	500	accuracy
Hateful Memes	2	8,500	500	ROC AUC
Rendered SST2	2	7,792	1,821	accuracy
ImageNet	1000	1,281,167	50,000	accuracy


(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
T_1 T_2 T_3 ... T_N
(3) Use for zero-shot prediction
Image Encoder
I_1
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
A photo of a dog.


StanfordCars +28.9
Country211 +23.2
Food101 +22.5
Kinetics700 +14.5
SST2 +12.4
SUN397 +7.8
UCF101 +7.7
HatefulMemes +6.7
CIFAR10 +3.9
CIFAR100 +3.0
STL10 +3.0
FER2013 +2.8
Caltech101 +2.0
ImageNet +1.9
OxfordPets +1.1
PascalVOC2007 +0.5
-3.2 Birdsnap
-10.0 MNIST
-11.3 FGVCAircraft
-11.9 RESISC45
-12.5 Flowers102
-16.6 DTD
-18.2 CLEVRCounts
-18.4 GTSRB
-19.5 PatchCamelyon
-34.0 KITTI Distance
-37.1 EuroSAT
−40 −30 −20 −10 0 10 20 30 40
Δ Score (%)
Zero-Shot CLIP vs. Linear Probe on ResNet50


Linear probe average over Kornblith et al.'s 12 datasets
Linear probe average over all 27 datasets
Average Score (%)
Forward-pass GFLOPs/image
L/14@336px
L/14
RN50x64
RN50x16
RN50x4
B/16
B/32
RN101
RN50
L2-475
L2-800
B7-NS
B8
B0-NS
B0
FixRes
x48
x8
R152x3
R152x4
ViT-H/14
res200x2
res50x1
ResNet152
ResNet50
MoCo-v2
CLIP-ViT
CLIP-ResNet
EfficientNet-NoisyStudent
EfficientNet
Instagram-pretrained
SimCLRv2
BYOL
MoCo
ViT (ImageNet-21k)
BiT-M
BiT-S
ResNet

Outline

- Context and Receptive Field
- Going Beyond Convolutions in…
  - Text
  - Images
  - Point Clouds
- Notable Transformer “Foundation” models
  - CLIP-like models
  - DINO models

DINOv2: Learning Robust Visual Features without Supervision

Maxime Oquab, Timothée Darcet**, Théo Moutakanni**, Huy V. Vo*, Marc Szafraniec*, Vasil Khalidov*, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal[1], Patrick Labatut*, Armand Joulin*, Piotr Bojanowski***

Meta AI Research [1]*Inria*

*core team **equal contribution

Reviewed on OpenReview: https://openreview.net/forum?id=a68SUt6zFt

Abstract

The recent breakthroughs in natural language processing for model pretraining on large quantities of data have opened the way for similar foundation models in computer vision. These models could greatly simplify the use of images in any system by producing general-purpose visual features, i.e., features that work across image distributions and tasks without finetuning. This work shows that existing pretraining methods, especially self-supervised methods, can produce such features if trained on enough curated data from diverse sources. 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. In terms of models, we train a ViT model (Dosovitskiy et al., 2021) with 1B parameters and distill it into a series of smaller models that surpass the best available general-purpose features, OpenCLIP (Ilharco et al., 2021) on most of the benchmarks at image and pixel levels.

DinoV2 Datasets

- 140 M total images
- No *labels* used
- Self-supervised with image level strategies (like SimCLR) and patch level strategies (like MAE, masked auto-encoder)
- Smaller models are distilled from the largest model

Dataset	Pretraining (as is)	Retrieving pretraining data	Eval.	Task	Citation
ImageNet-1k	✗	✓	✓	Classif.	(Russakovsky et al., 2015)
ImageNet-22k	✓	✓	✗		(Deng et al., 2009)
ImageNet-V2	✗	✗	✓	Classif.	(Recht et al., 2019)
ImageNet-ReaL	✗	✗	✓	Classif.	(Beyer et al., 2020)
ImageNet-A	✗	✗	✓	Classif.	(Hendrycks et al., 2021b)
ImageNet-C	✗	✗	✓	Classif.	(Hendrycks & Dietterich, 2019)
ImageNet-R	✗	✗	✓	Classif.	(Hendrycks et al., 2021a)
ImageNet-Sk.	✗	✗	✓	Classif.	(Wang et al., 2019)
Food-101	✗	✓	✓	Classif.	(Bossard et al., 2014)
CIFAR-10	✗	✓	✓	Classif.	(Krizhevsky et al., 2009)
CIFAR-100	✗	✓	✓	Classif.	(Krizhevsky et al., 2009)
SUN397	✗	✓	✓	Classif.	(Xiao et al., 2010)
StanfordCars	✗	✓	✓	Classif.	(Krause et al., 2013)
FGVC-Aircraft	✗	✓	✓	Classif.	(Maji et al., 2013)
VOC 2007	✗	✓	✓	Classif.	(Everingham et al., 2010)
DTD	✗	✓	✓	Classif.	(Cimpoi et al., 2014)
Oxford Pets	✗	✓	✓	Classif.	(Parkhi et al., 2012)
Caltech101	✗	✓	✓	Classif.	(Fei-Fei et al., 2004)
Flowers	✗	✓	✓	Classif.	(Nilsback & Zisserman, 2008)
CUB200	✗	✓	✓	Classif.	(Welinder et al., 2010)
iNaturalist 2018	✗	✗	✓	Classif.	(Van Horn et al., 2018)
iNaturalist 2021	✗	✗	✓	Classif.	(Van Horn et al., 2021)
Places-205	✗	✗	✓	Classif.	(Zhou et al., 2014)
UCF101	✗	✗	✓	Video	(Soomro et al., 2012)
Kinetics-400	✗	✗	✓	Video	(Kay et al., 2017)
SSv2	✗	✗	✓	Video	(Goyal et al., 2017)
GLD v2	✓	✓	✗		(Weyand et al., 2020)
R-Paris	✗	✓	✓	Retrieval	(Radenović et al., 2018a)
R-Oxford	✗	✓	✓	Retrieval	(Radenović et al., 2018a)
Met	✗	✓	✓	Retrieval	(Ypsilantis et al., 2021)
Amstertime	✗	✓	✓	Retrieval	(Yildiz et al., 2022)
ADE20k	✗	✓	✓	Seg.	(Zhou et al., 2017)
Cityscapes	✗	✓	✓	Seg.	(Cordts et al., 2016)
VOC 2012	✗	✓	✓	Seg.	(Everingham et al., 2010)
Mapillary SLS	✓	✗	✗		(Warburg et al., 2020)
NYU-Depth V2	✗	✓	✓	Depth	(Silberman et al., 2012)
KITTI	✗	✓	✓	Depth	(Geiger et al., 2013)
SUN-RGBD	✗	✓	✓	Depth	(Song et al., 2015)
DollarStreet	✗	✗	✓	Fairness	(De Vries et al., 2019)
Casual Conv.	✗	✗	✓	Fairness	(Hazirbas et al., 2021)

	INet-1k k-NN	INet-1k linear
iBOT	72.9	82.3
+(our reproduction)	74.5 ↑ 1.6	83.2 ↑ 0.9
+LayerScale, Stochastic Depth	75.4 ↑ 0.9	82.0 ↓ 1.2
+128k prototypes	76.6 ↑ 1.2	81.9 ↓ 0.1
+KoLeo	78.9 ↑ 2.3	82.5 ↑ 0.6
+SwiGLU FFN	78.7 ↓ 0.2	83.1 ↑ 0.6
+Patch size 14	78.9 ↑ 0.2	83.5 ↑ 0.4
+Teacher momentum 0.994	79.4 ↑ 0.5	83.6 ↑ 0.1
+Tweak warmup schedules	80.5 ↑ 1.1	83.8 ↑ 0.2
+Batch size 3k	81.7 ↑ 1.2	84.7 ↑ 0.9
+Sinkhorn-Knopp	81.7 =	84.7 =
+Untying heads = DINOv2	82.0 ↑ 0.3	84.5 ↓ 0.2

Method	Arch.	Data	Text sup.	kNN	linear		
				val	val	ReaL	V2
Weakly supervised							
CLIP	ViT-L/14	WIT-400M	✓	79.8	84.3	88.1	75.3
CLIP	ViT-L/14_{336}	WIT-400M	✓	80.5	85.3	88.8	75.8
SWAG	ViT-H/14	IG3.6B	✓	82.6	85.7	88.7	77.6
OpenCLIP	ViT-H/14	LAION-2B	✓	81.7	84.4	88.4	75.5
OpenCLIP	ViT-G/14	LAION-2B	✓	83.2	86.2	89.4	77.2
EVA-CLIP	ViT-g/14	custom*	✓	**83.5**	86.4	89.3	77.4
Self-supervised							
MAE	ViT-H/14	INet-1k	✗	49.4	76.6	83.3	64.8
DINO	ViT-S/8	INet-1k	✗	78.6	79.2	85.5	68.2
SEERv2	RG10B	IG2B	✗	–	79.8	–	–
MSN	ViT-L/7	INet-1k	✗	79.2	80.7	86.0	69.7
EsViT	Swin-B/W=14	INet-1k	✗	79.4	81.3	87.0	70.4
Mugs	ViT-L/16	INet-1k	✗	80.2	82.1	86.9	70.8
iBOT	ViT-L/16	INet-22k	✗	72.9	82.3	87.5	72.4
DINOv2	ViT-S/14	LVD-142M	✗	79.0	81.1	86.6	70.9
	ViT-B/14	LVD-142M	✗	82.1	84.5	88.3	75.1
	ViT-L/14	LVD-142M	✗	**83.5**	86.3	89.5	78.0
	ViT-g/14	LVD-142M	✗	**83.5**	**86.5**	**89.6**	**78.4**

Semantic Segmentation

		ADE20k (62.9)		CityScapes (86.9)		Pascal VOC (89.0)	
Method	Arch.	lin.	+ms	lin.	+ms	lin.	+ms
OpenCLIP	ViT-G/14	39.3	46.0	60.3	70.3	71.4	79.2
MAE	ViT-H/14	33.3	30.7	58.4	61.0	67.6	63.3
DINO	ViT-B/8	31.8	35.2	56.9	66.2	66.4	75.6
iBOT	ViT-L/16	44.6	47.5	64.8	74.5	82.3	84.3
DINOv2	ViT-S/14	44.3	47.2	66.6	77.1	81.1	82.6
	ViT-B/14	47.3	51.3	69.4	80.0	82.5	84.9
	ViT-L/14	47.7	**53.1**	70.3	80.9	82.1	86.0
	ViT-g/14	**49.0**	53.0	**71.3**	**81.0**	**83.0**	**86.2**

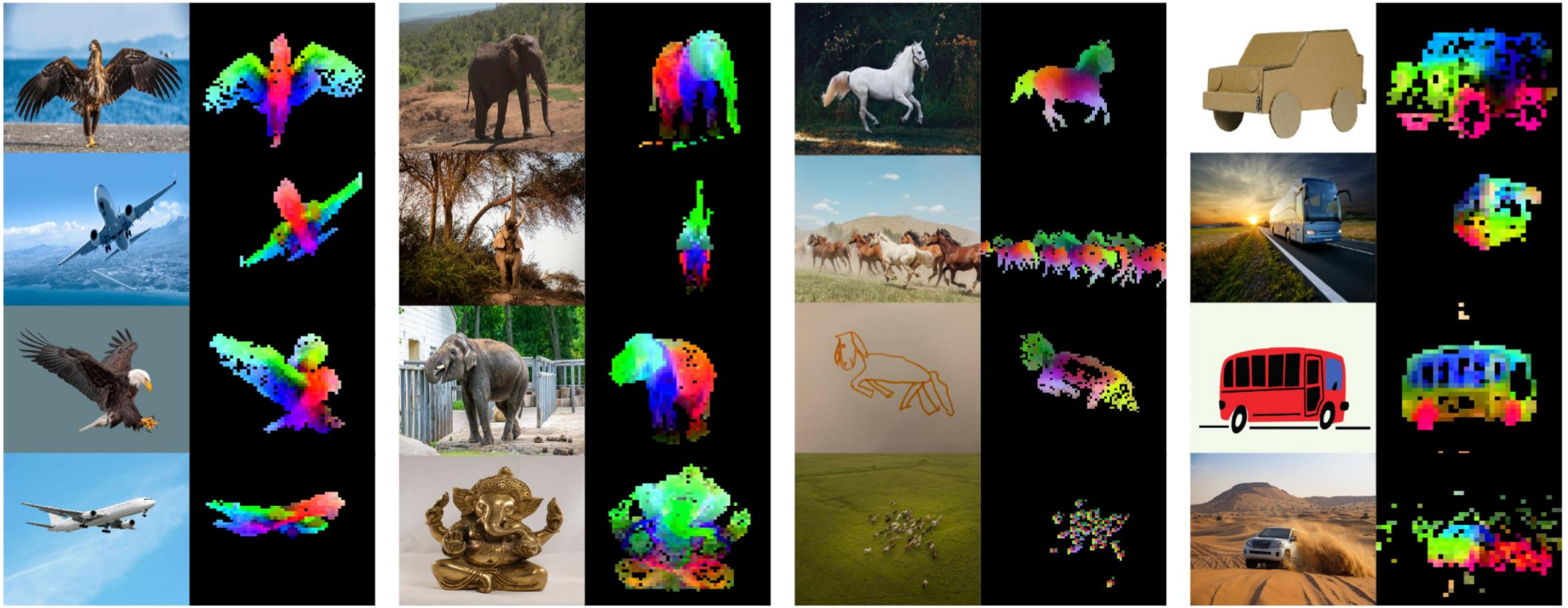


Figure 1: **Visualization of the first PCA components.** We compute a PCA between the patches of the images from the same column (a, b, c and d) and show their first 3 components. Each component is matched to a different color channel. Same parts are matched between related images despite changes of pose, style or even objects. Background is removed by thresholding the first PCA component.

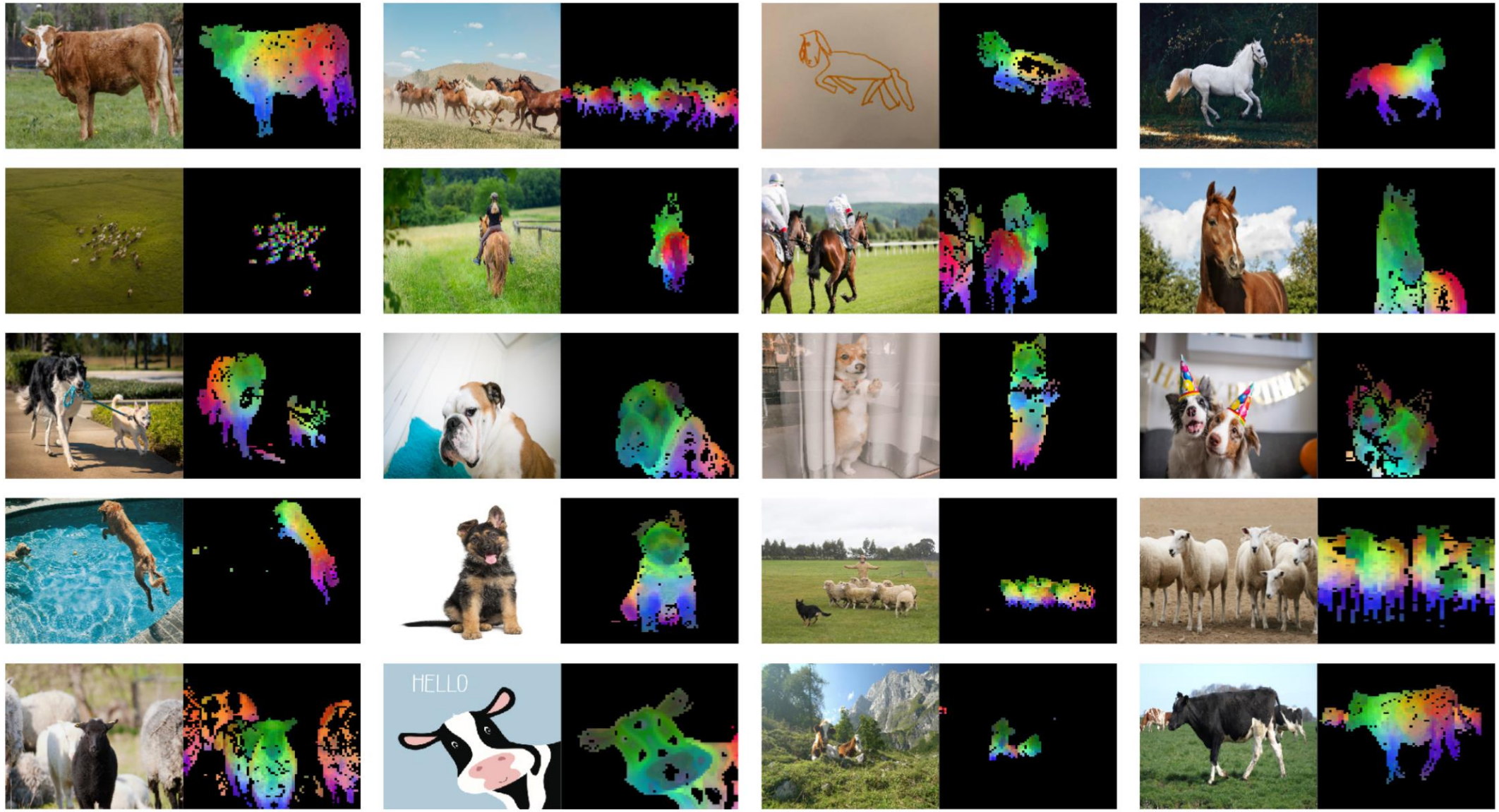


Figure 9: **More visualization of the first PCA components.** We compute the PCA between the patches from all of the images and show their first 3 components. Each component corresponds to a specific color channel. Same parts are matched between related images depsite changes of pose, style or even objects. Background is removed by removing patches with a negative score of the first PCA component.

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**[1] **Camille Couprie**
Julien Mairal[2] **Hervé Jégou** **Patrick Labatut** **Piotr Bojanowski**

Meta AI Research [1] *WRI* [2] *Inria*

* *corresponding authors: {osimeoni,huyvvo,seitzer,baldassarre,qas}@meta.com*

Abstract

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

Table 2: Comparison of the teacher architectures used in DINOv2 and DINOv3 models. We keep the model 40 blocks deep, and increase the embedding dimension to 4096. Importantly, we use a patch size of 16 pixels, changing the effective sequence length for a given resolution.

Teacher model	DINOv2	DINOv3
Backbone	ViT-giant	ViT-7B
#Params	1.1B	6.7B
#Blocks	40	40
Patch Size	14	16
Pos. Embeddings	Learnable	RoPE
Registers	4	4
Embed. Dim.	1536	4096
FFN Type	SwiGLU	SwiGLU
FFN Hidden Dim.	4096	8192
Attn. Heads	24	32
Attn. Heads Dim.	64	128
DINO Head MLP	4096-4096-256	8192-8192-512
DINO Prototypes	128k	256k
iBOT Head MLP	4096-4096-256	8192-8192-384
iBOT Prototypes	128k	96k

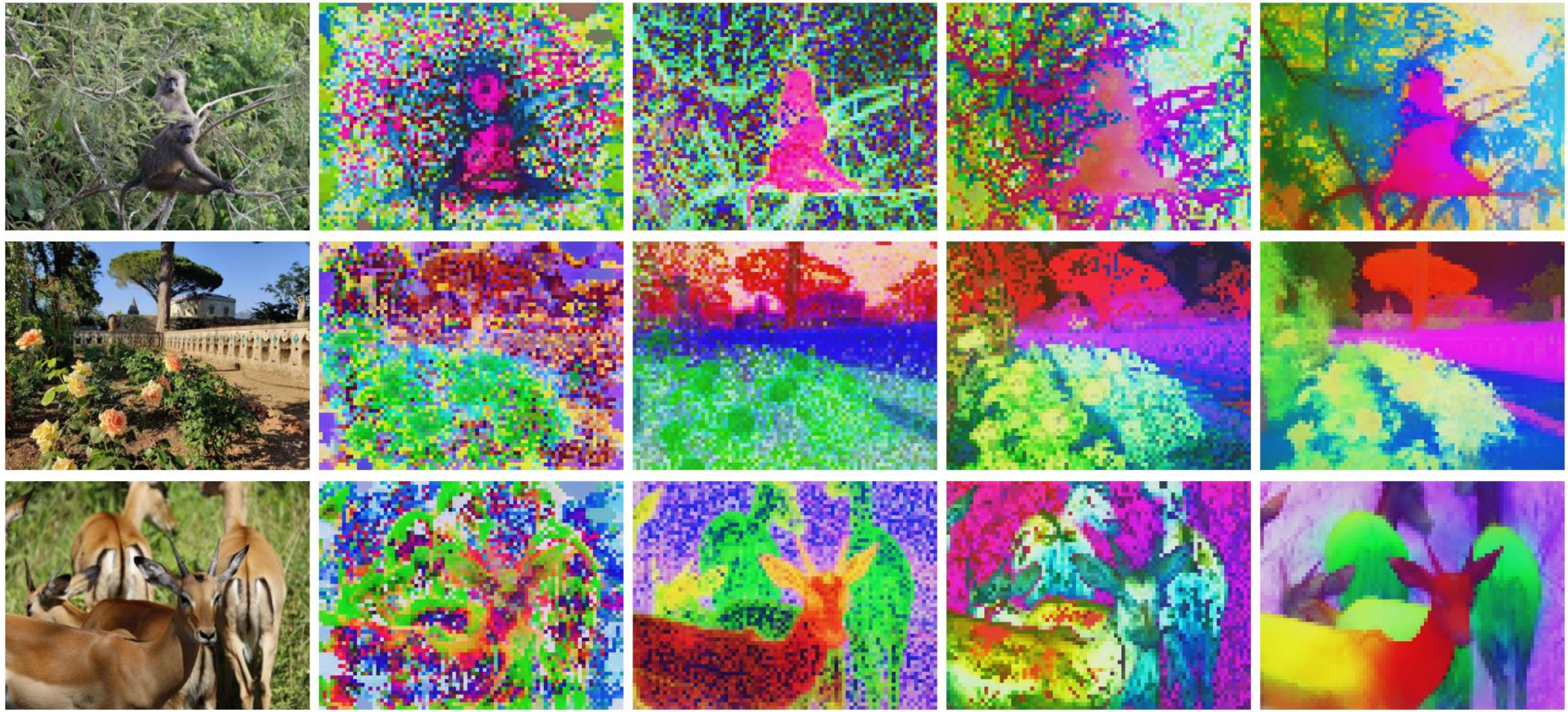
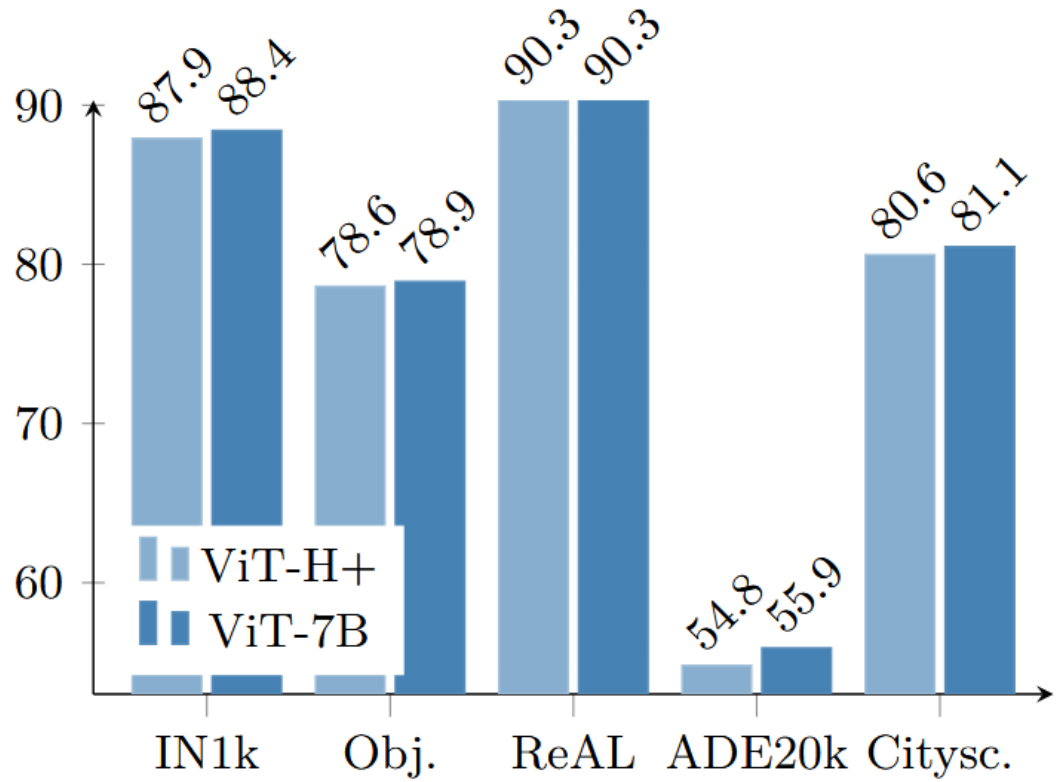


Figure 13: Comparison of dense features. We compare several vision backbones by projecting their dense outputs using PCA and mapping them to RGB. From left to right: SigLIP 2 ViT-g/16, PEspatial ViT-G/14, DINOv2 ViT-g/14 with registers, DINOv3 ViT-7B/16. Images are forwarded at resolution 1280×960 for models using patch 16 and 1120×840 for patch 14, *i.e.* all feature maps have size 80×60.

Model	#Params	Inference GFLOPs	
		Res. 256	Res. 512
CNX-Tiny	29M	5	20
CNX-Small	50M	11	46
CNX-Base	89M	20	81
CNX-Large	198M	38	152
ViT-S	21M	12	63
ViT-S+	29M	16	79
ViT-B	86M	47	216
ViT-L	300M	163	721
ViT-H+	840M	450	1903
ViT-7B	6716M	3550	14515

(a) DINOv3 family of models.



(b) ViT-H+ v.s. ViT-7B.

Figure 16: (a) Presentation of the distilled models' characteristics. CNX stands for ConvNeXT. We present per model the number of parameters and the GFLOPs estimated on images of size 256×256 and 512×512. (b) We compare DINOv3 ViT-H+ to its 7B-sized teacher; despite having almost 10× less parameters, the ViT-H+ is close to DINOv3 7B in performance.

Summary

- “Attention” models outperform recurrent models and convolutional models for **sequence** processing. They allow long range interactions.
- These models do best with LOTS of training data
- Naïve attention mechanisms have **quadratic** complexity with the number of input tokens, but there are often workarounds for this.
- Tokenization seems harmful, so we have to fight back against its with positional embeddings or overlapping tokenizations.
- Attentional models seem to succeed when they copy the inductive biases of convolutional models.
- For “traditional” image processing, it is not clear if Transformers outperform convolutional networks.
- You should start with one of these pre-trained models – CLIP if you want semantic / language support, Dino if you want dense spatial reasoning