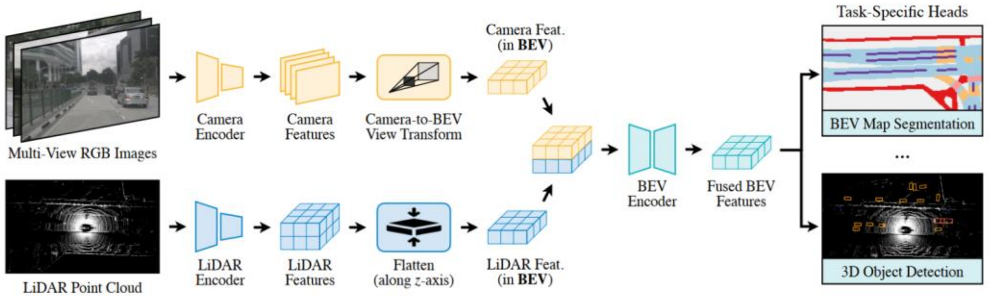
“Attention” and “Transformer” Architectures

James Hays

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).



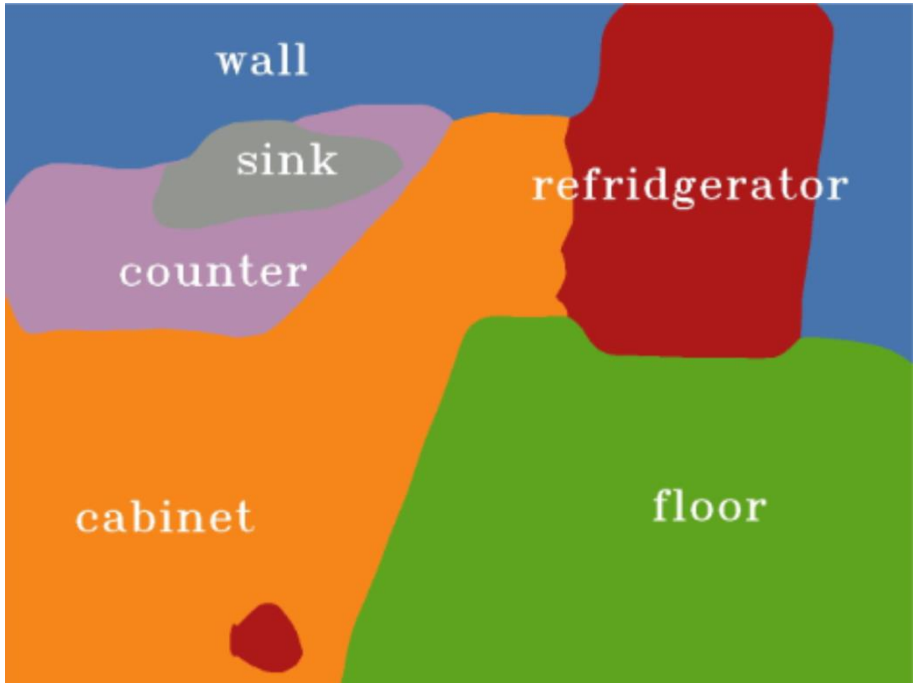
BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation.
Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L. Rus, Song Han

https://paperswithcode.com/sota/3d-object-detection-on-nuscenes

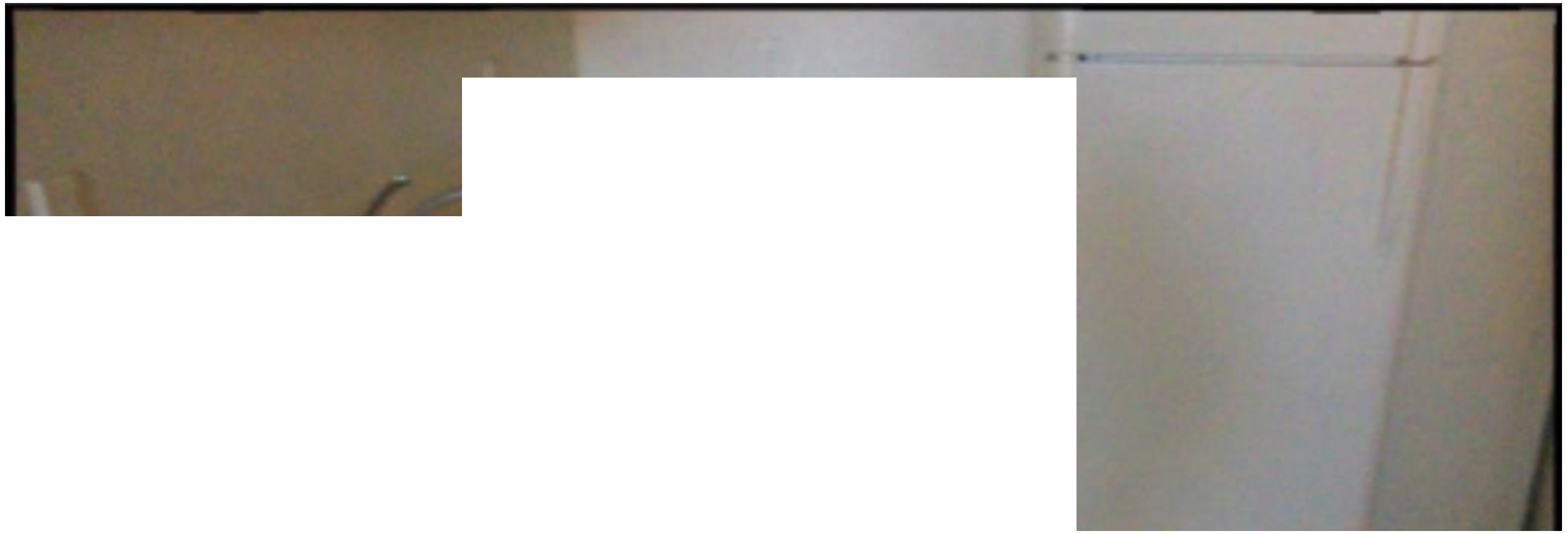
Outline

- Context and Receptive Field
- Going Beyond Convolutions in…
  - Text
  - Point Clouds
  - Images

Ground truth



Prediction from Mseg



Language understanding

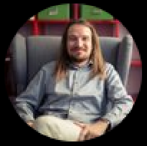
… serve …

Language understanding

… great serve from Djokovic …

Language understanding

… be right back after I serve these salads …



Brendan Dolan-Gavitt
@moyix

The latest generation of adversarial image attacks is, uh, somewhat simpler to carry out openai.com/blog/multimoda...

Attacks in the wild

We refer to these attacks as *typographic attacks*. We believe attacks such as those described above are far from simply an academic concern. By exploiting the model's ability to read text robustly, we find that even *photographs of hand-written text* can often fool the model. Like the Adversarial Patch,[22] this attack works in the wild; but unlike such attacks, it requires no more technology than pen and paper.

Attack text label iPod

Granny Smith	85.6%
iPod	0.4%
library	0.0%
pizza	0.0%
toaster	0.0%
dough	0.1%

Granny Smith	0.1%
iPod	99.7%
library	0.0%
pizza	0.0%
toaster	0.0%
dough	0.0%

When we put a label saying "iPod" on this Granny Smith apple, the model erroneously classifies it as an iPod in the zero-shot setting.

3:38 PM · Mar 4, 2021 · Twitter Web App

Mark O. Riedl
@mark_riedl

Replying to @mark_riedl

In case of AI uprising...

6:42 PM · Mar 4, 2021 · Twitter for iPad

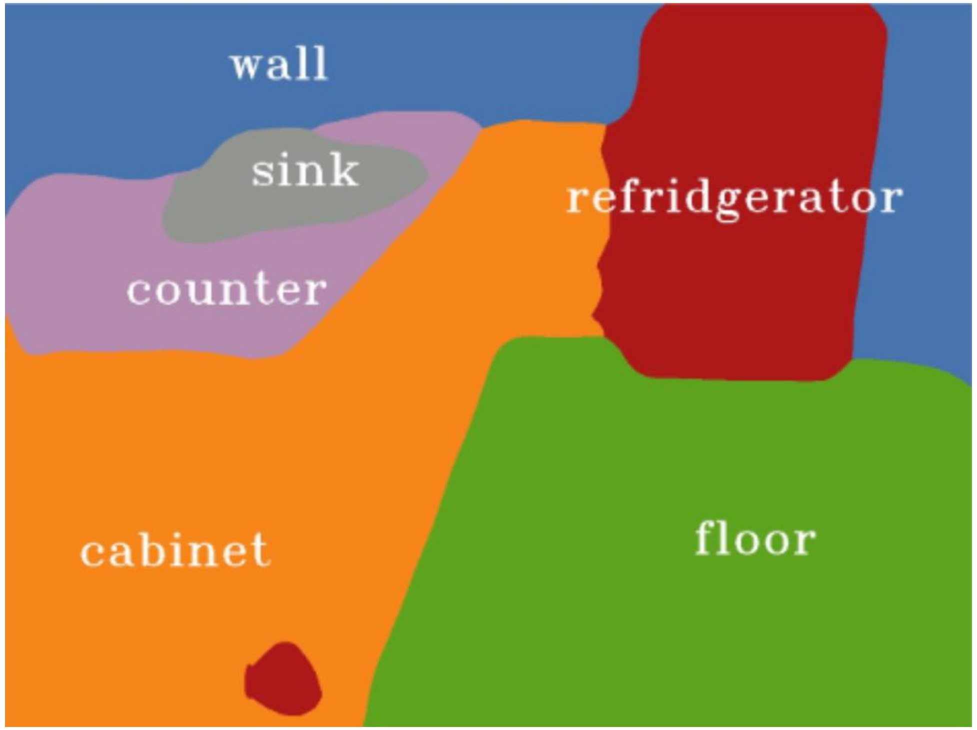
Mark O. Riedl
@mark_riedl

Replying to @mark_riedl

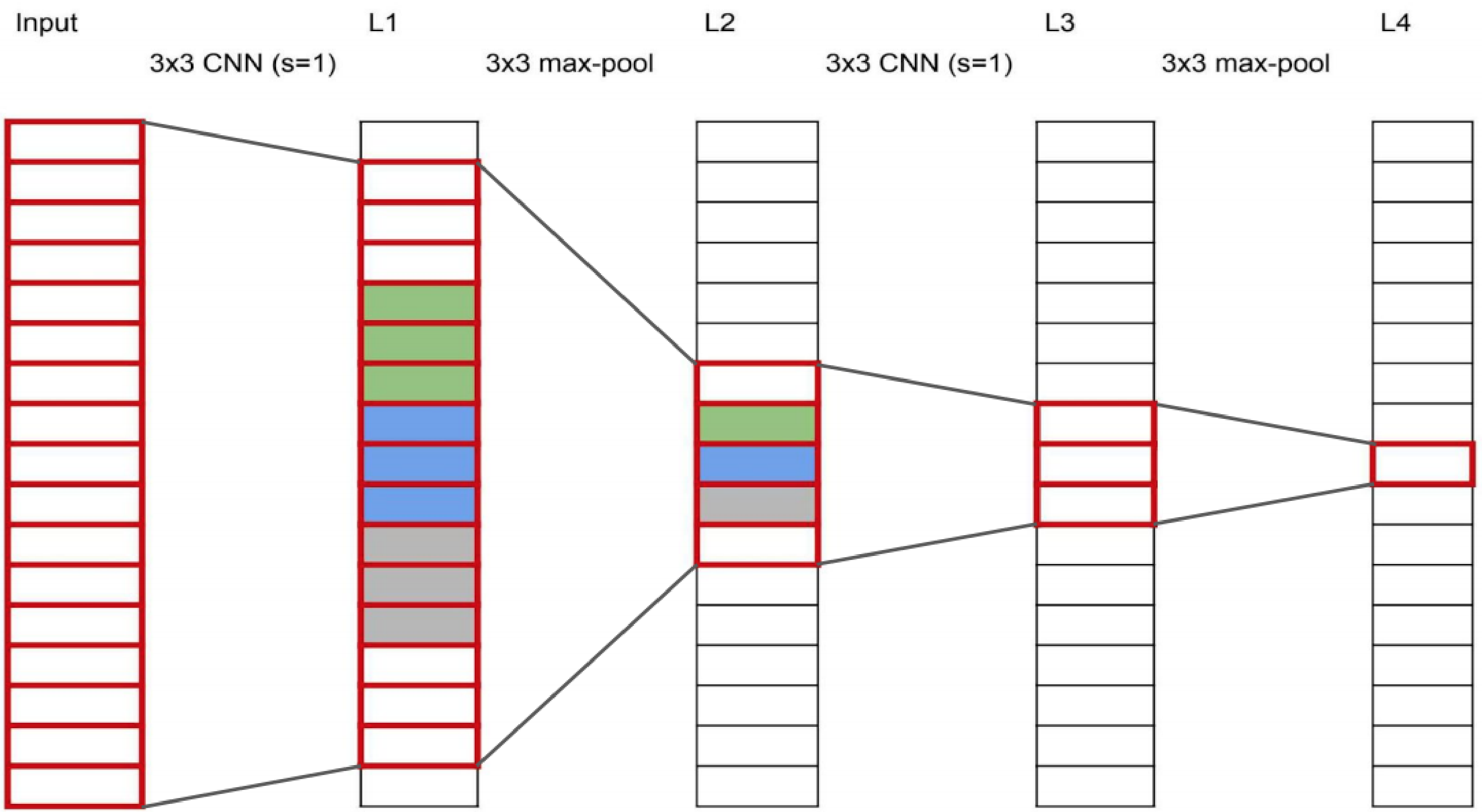
Upon further reflection, neural language models aren’t always so good with negations. I recommend this instead

9:28 PM · Mar 4, 2021 · Twitter for iPad

So how do we fix these problems?

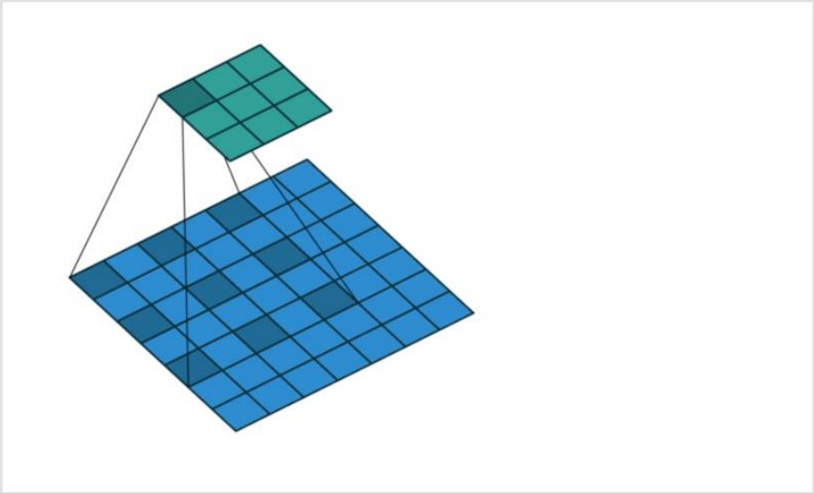


Receptive field

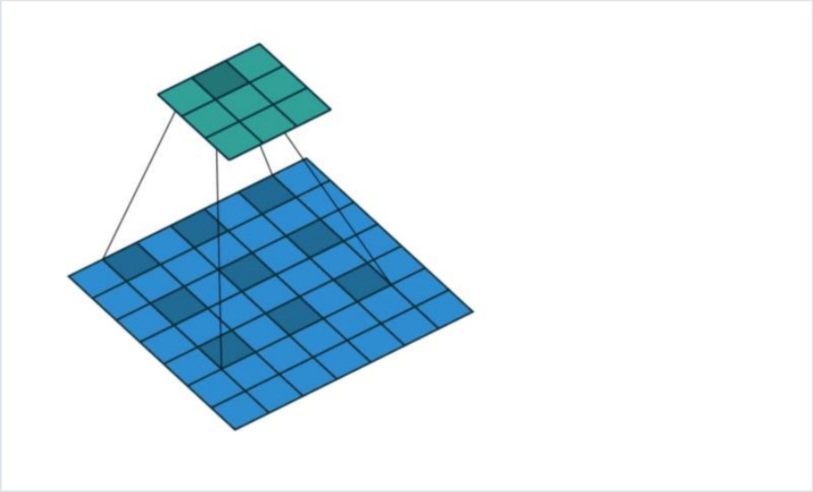


Slide Credit: Frank Dellaert https://dellaert.github.io/19F-4476/resources/receptiveField.pdf

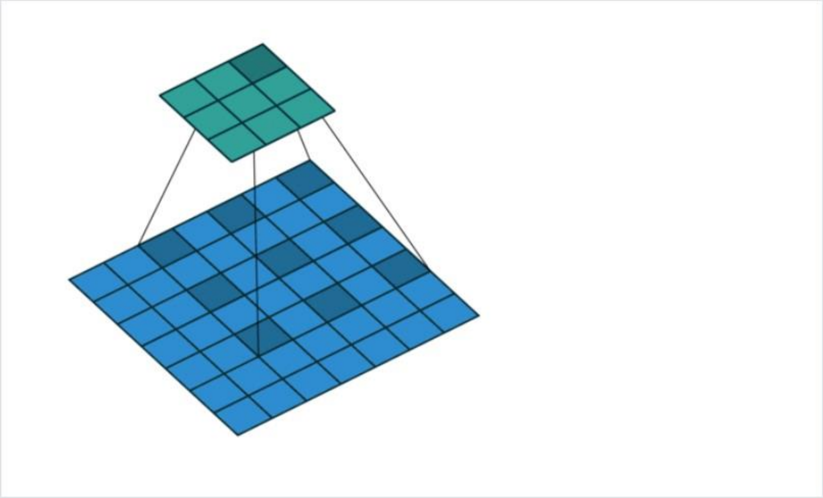
Dilated Convolution



No padding, no stride, dilation



No padding, no stride, dilation



No padding, no stride, dilation

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

Receptive field could also be an issue in 3D

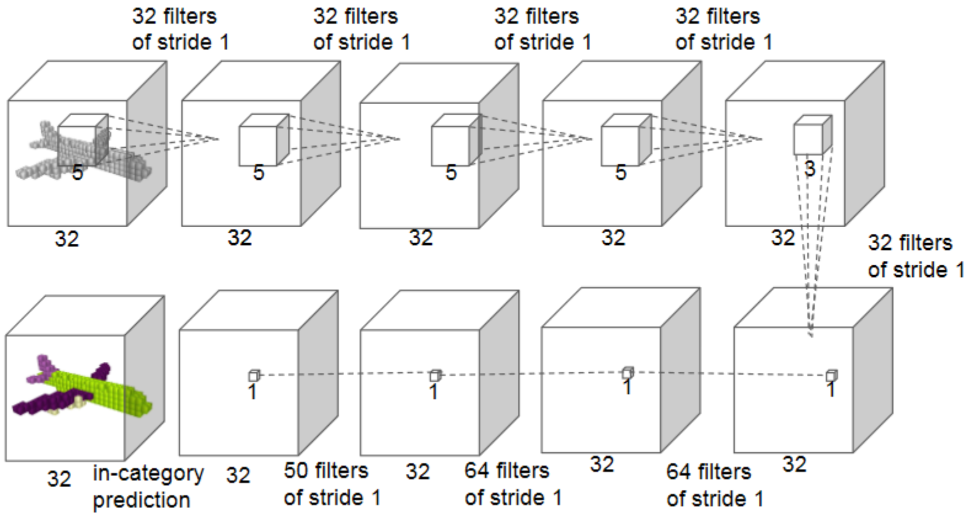


Figure 10. **Baseline 3D CNN segmentation network.** The network is fully convolutional and predicts part scores for each voxel.

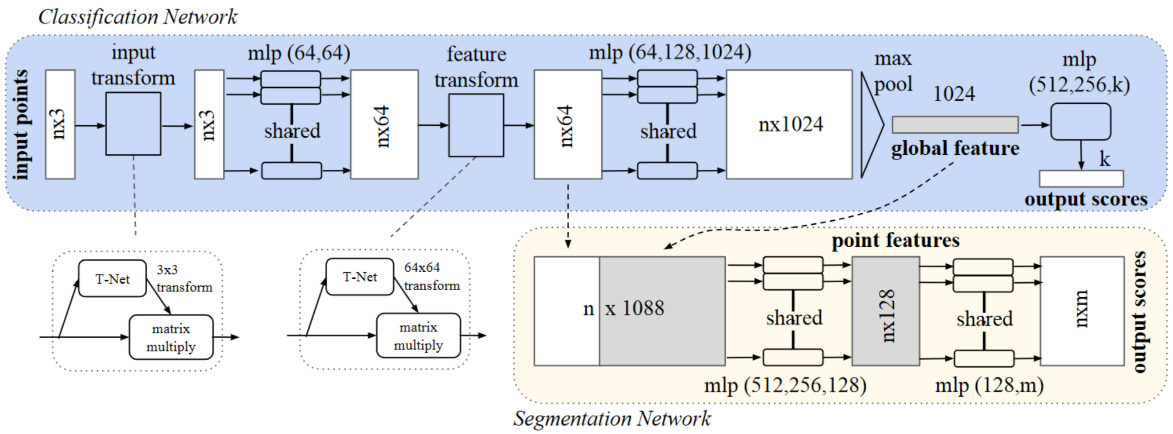


Figure 2. **PointNet Architecture.** The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net. 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.

Outline

- Context and Receptive Field
- Going Beyond Convolutions in…
  - Text
  - Point Clouds
  - Images

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

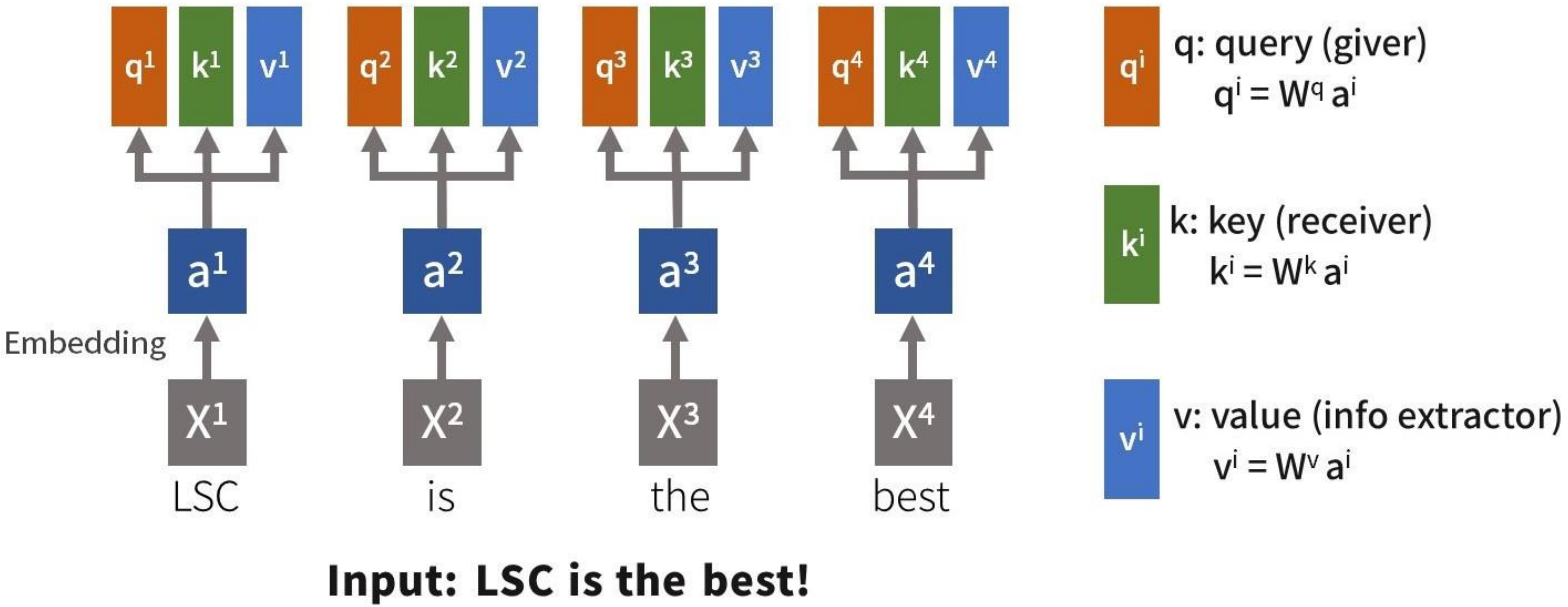
Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

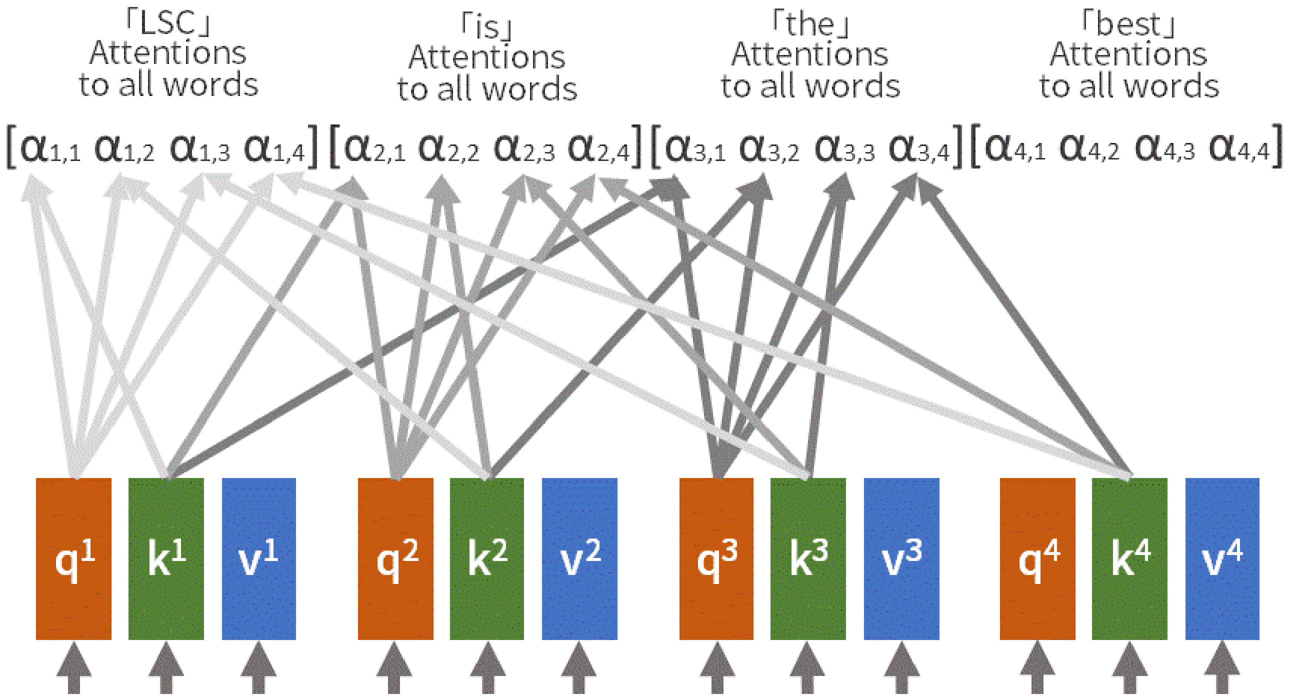
Abstract

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,

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



From https://medium.com/lsc-psd/introduction-of-self-attention-layer-in-transformer-fc7bff63f3bc

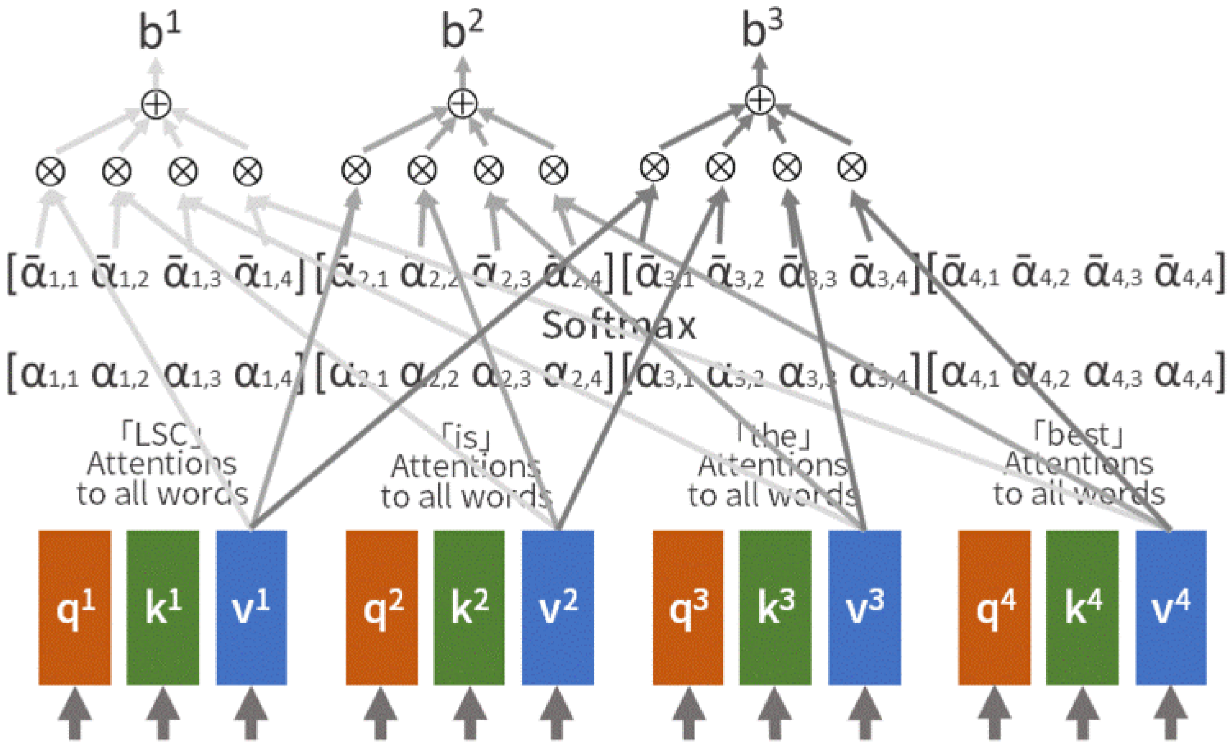


$$\alpha_{i,j} = \frac{q^i \cdot k^j}{\sqrt{d}}$$

d: dimension of q, k

$$\underset{\text{Attention Matrix}}{A} = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \alpha_{1,3} & \alpha_{1,4} \\ \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} & \alpha_{2,4} \\ \alpha_{3,1} & \alpha_{3,2} & \alpha_{3,3} & \alpha_{3,4} \\ \alpha_{4,1} & \alpha_{4,2} & \alpha_{4,3} & \alpha_{4,4} \end{bmatrix}$$

From https://medium.com/lsc-psd/introduction-of-self-attention-layer-in-transformer-fc7bff63f3bc

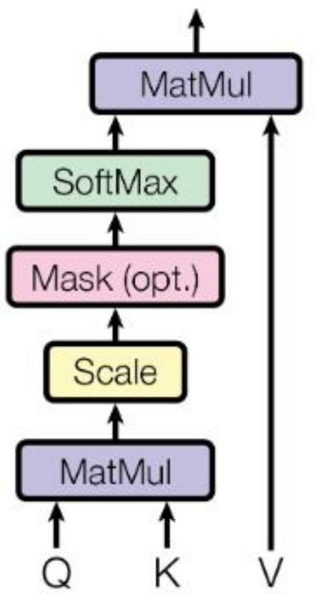


$$b^i = \sum_j \bar{\alpha}_{i,j} v^j$$

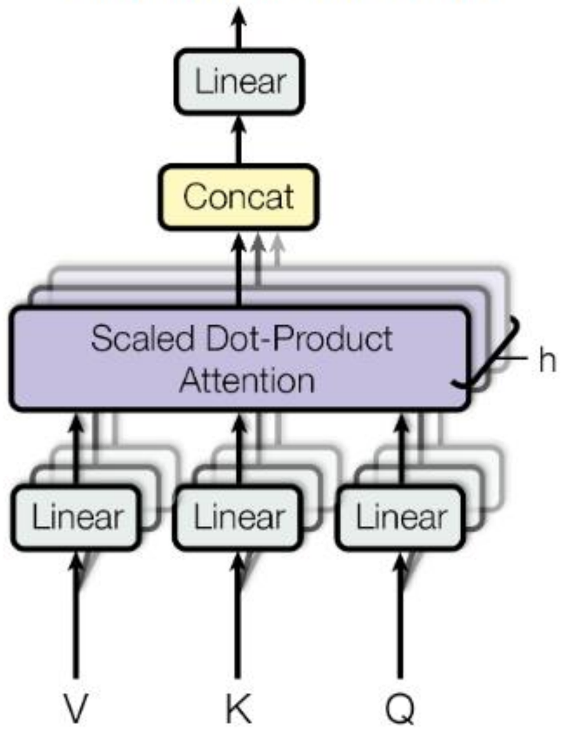
Complexity Comparison

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$

Scaled Dot-Product Attention



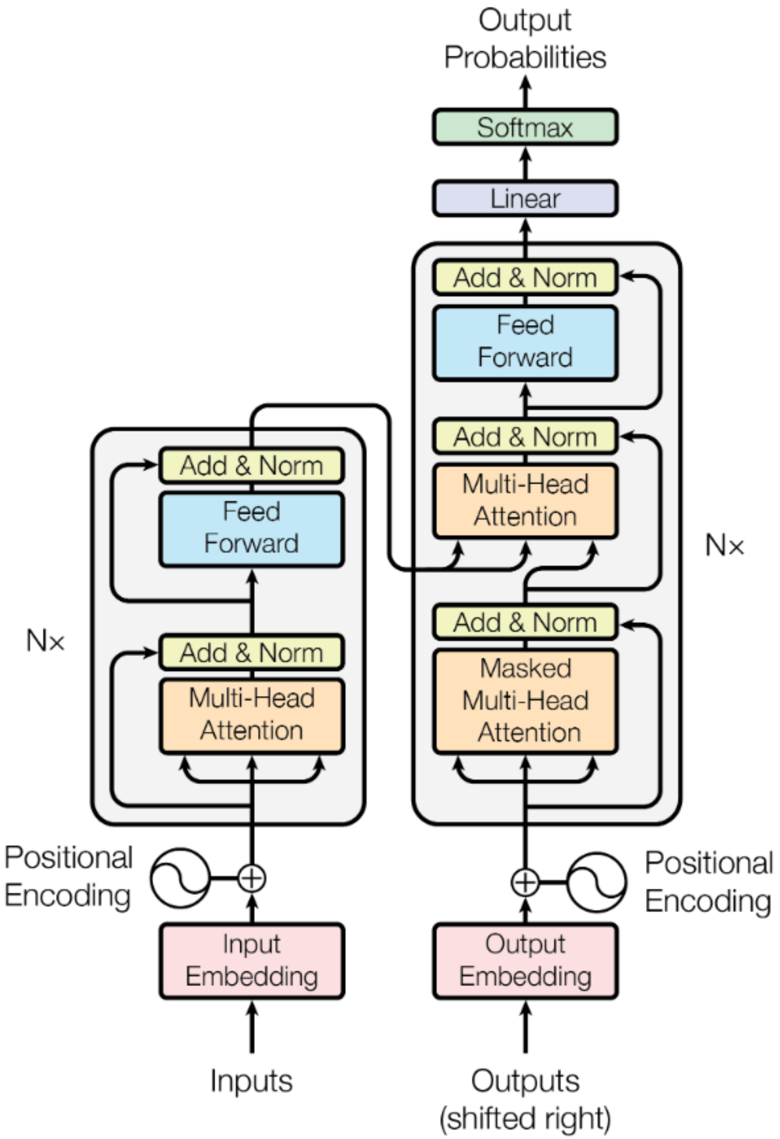
Multi-Head Attention



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	$\mathbf{3.3 \cdot 10^{18}}$	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	



Outline

- Context and Receptive Field
- Going Beyond Convolutions in…
  - Text
  - Point Clouds
  - Images

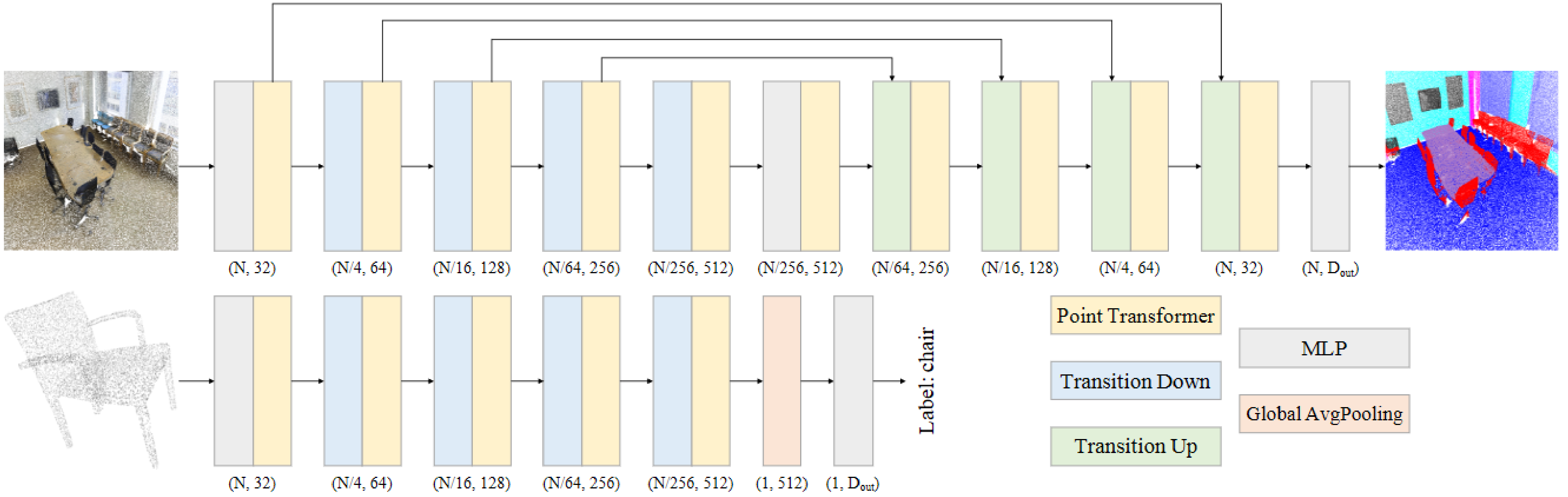
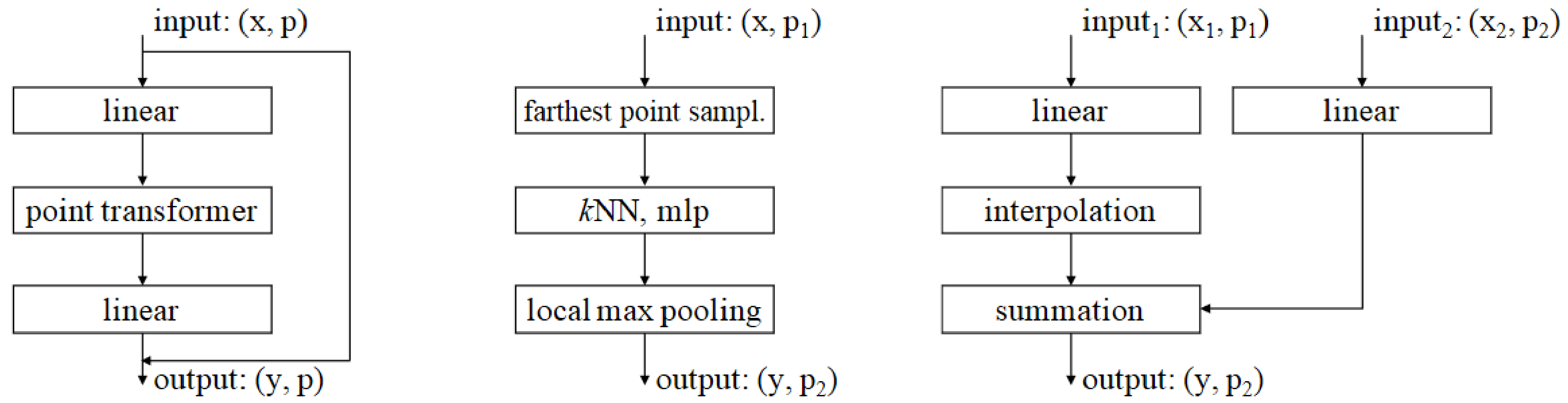
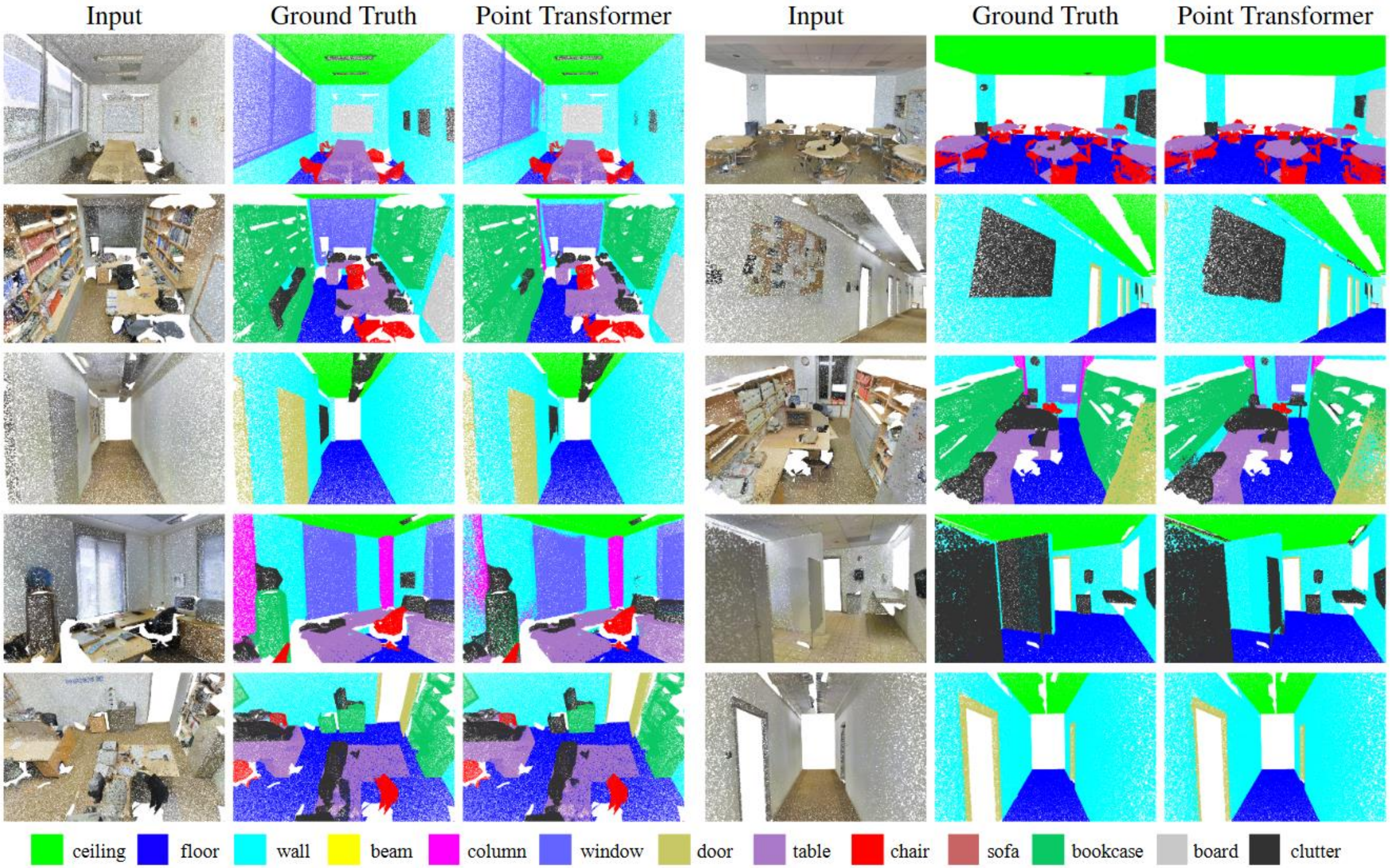


Figure 3. Point transformer networks for semantic segmentation (top) and classification (bottom).



(a) point transformer block (b) transition down (c) transition up

Point Transformer. Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, Vladlen Koltun


Input
Ground Truth
Point Transformer
Input
Ground Truth
Point Transformer
ceiling
floor
wall
beam
column
window
door
table
chair
sofa
bookcase
board
clutter

Method	OA	mAcc	mIoU	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
PointNet [22]	–	49.0	41.1	88.8	97.3	69.8	0.1	3.9	46.3	10.8	59.0	52.6	5.9	40.3	26.4	33.2
SegCloud [32]	–	57.4	48.9	90.1	96.1	69.9	0.0	18.4	38.4	23.1	70.4	75.9	40.9	58.4	13.0	41.6
TangentConv [31]	–	62.2	52.6	90.5	97.7	74.0	0.0	20.7	39.0	31.3	77.5	69.4	57.3	38.5	48.8	39.8
PointCNN [18]	85.9	63.9	57.3	92.3	98.2	79.4	0.0	17.6	22.8	62.1	74.4	80.6	31.7	66.7	62.1	56.7
SPGraph [14]	86.4	66.5	58.0	89.4	96.9	78.1	0.0	42.8	48.9	61.6	84.7	75.4	69.8	52.6	2.1	52.2
PCCN [38]	–	67.0	58.3	92.3	96.2	75.9	0.3	6.0	69.5	63.5	66.9	65.6	47.3	68.9	59.1	46.2
PointWeb [50]	87.0	66.6	60.3	92.0	98.5	79.4	0.0	21.1	59.7	34.8	76.3	88.3	46.9	69.3	64.9	52.5
HPEIN [12]	87.2	68.3	61.9	91.5	98.2	81.4	0.0	23.3	65.3	40.0	75.5	87.7	58.5	67.8	65.6	49.4
MinkowskiNet [33]	–	71.7	65.4	91.8	98.7	86.2	0.0	34.1	48.9	62.4	81.6	89.8	47.2	74.9	74.4	58.6
KPConv [33]	–	72.8	67.1	92.8	97.3	82.4	0.0	23.9	58.0	69.0	81.5	91.0	75.4	75.3	66.7	58.9
PointTransformer	**90.8**	**76.5**	**70.4**	94.0	98.5	86.3	0.0	38.0	63.4	74.3	89.1	82.4	74.3	80.2	76.0	59.3

Table 1. Semantic segmentation results on the S3DIS dataset, evaluated on Area 5.

Method	input	mAcc	OA
3DShapeNets [43]	voxel	77.3	84.7
VoxNet [20]	voxel	83.0	85.9
Subvolume [23]	voxel	86.0	89.2
MVCNN [30]	image	–	90.1
PointNet [22]	point	86.2	89.2
PointNet++ [24]	point	–	91.9
SpecGCN [36]	point	–	92.1
PointCNN [18]	point	88.1	92.2
DGCNN [40]	point	90.2	92.2
PointWeb [50]	point	89.4	92.3
SpiderCNN [44]	point	–	92.4
PointConv [42]	point	–	92.5
KPConv [33]	point	–	92.9
InterpCNN [19]	point	–	93.0
PointTransformer	point	**90.6**	**93.7**

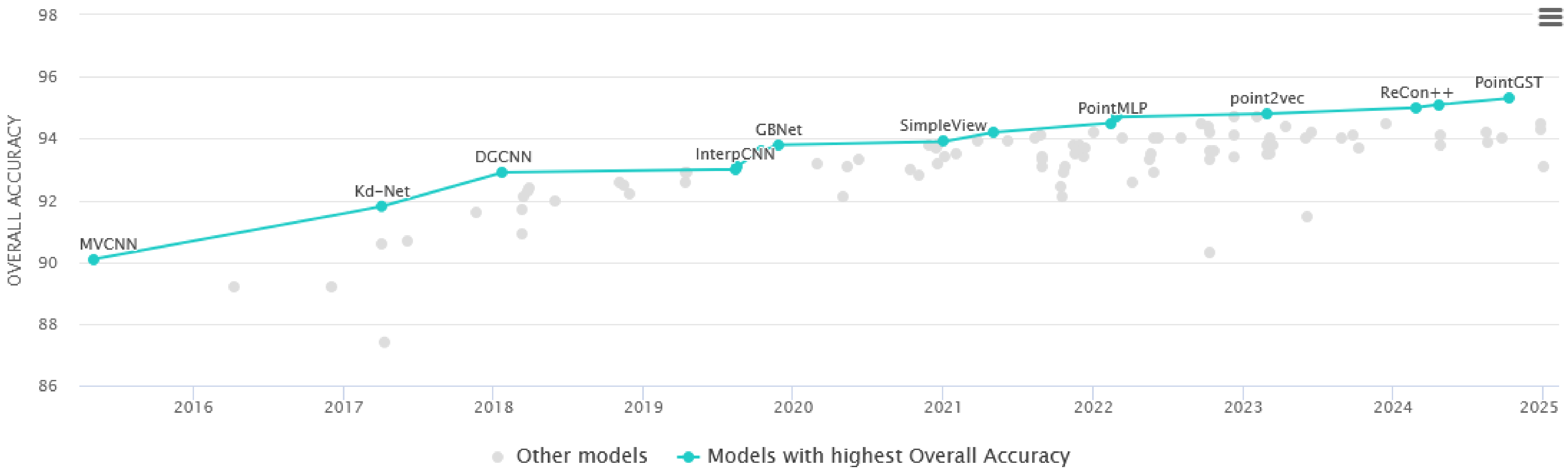
Table 3. Shape classification results on the ModelNet40 dataset.

https://paperswithcode.com/sota/3d-point-cloud-classification-on-modelnet40

3D Point Cloud Classification on ModelNet40

Leaderboard Dataset

View Overall Accuracy by Date for All models



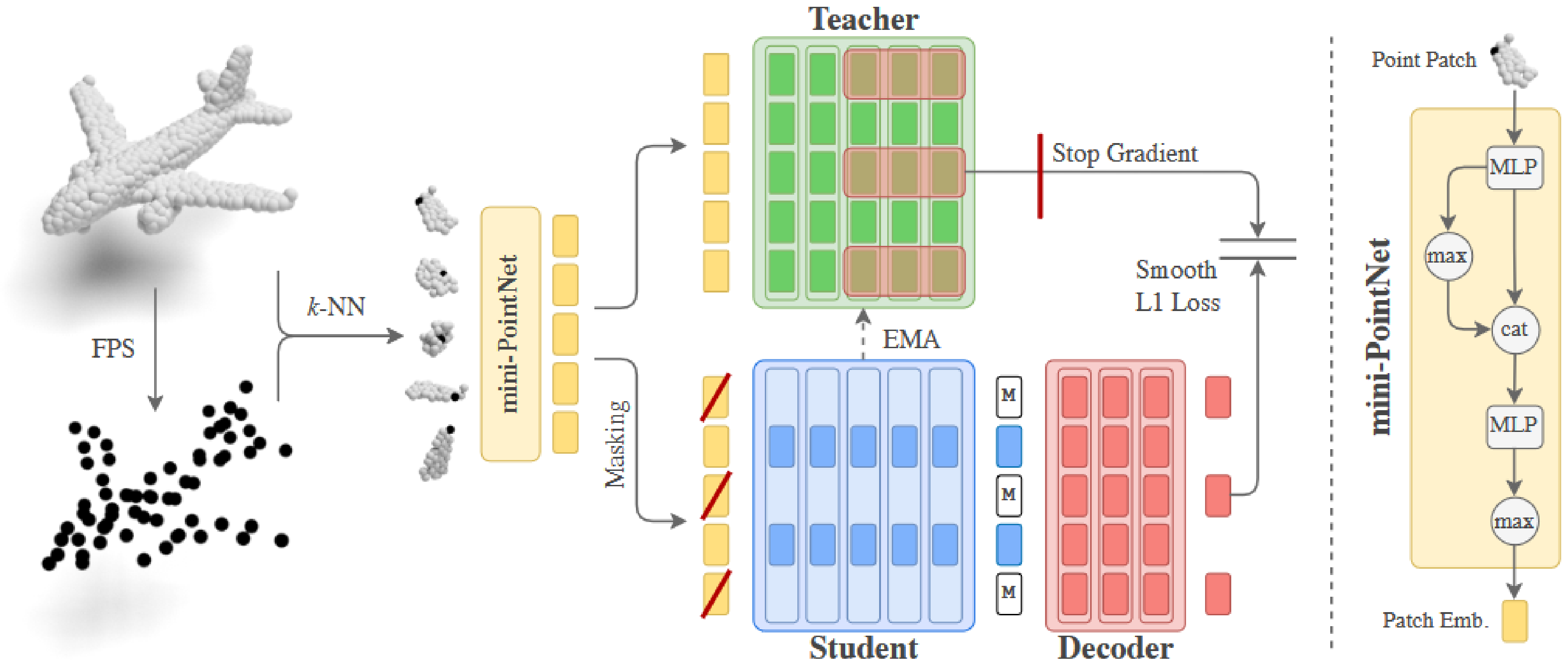


Fig. 2: Point2Vec pre-training. Our model divides the input point cloud into point patches using farthest point sampling (FPS) and k-NN aggregation. We obtain patch embeddings by applying a mini-PointNet ☐ to each point patch (*right*). The teacher Transformer encoder ☐ infers a contextualized representation 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.

Outline

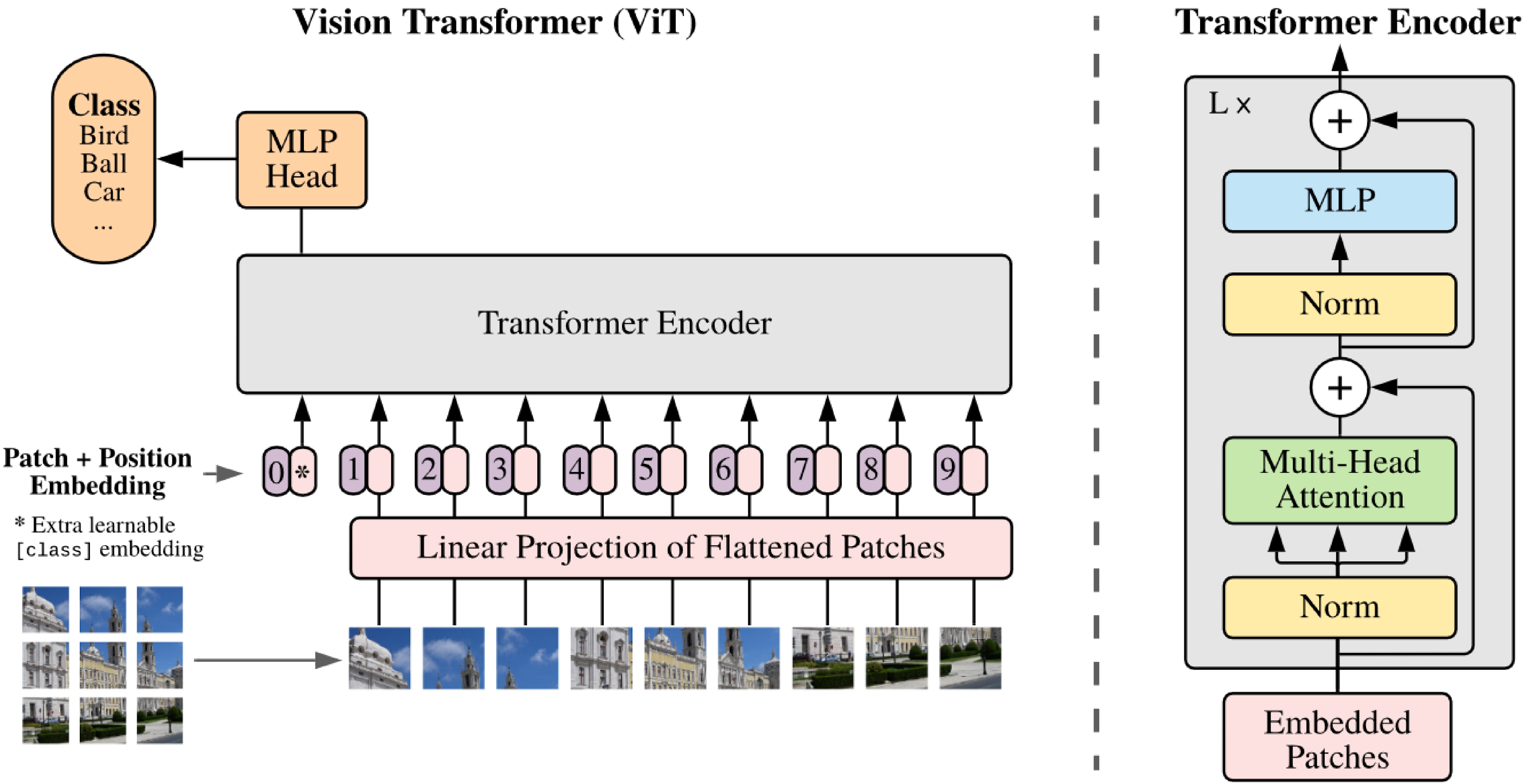
- Context and Receptive Field
- Going Beyond Convolutions in…
  - Text
  - Point Clouds
  - Images

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

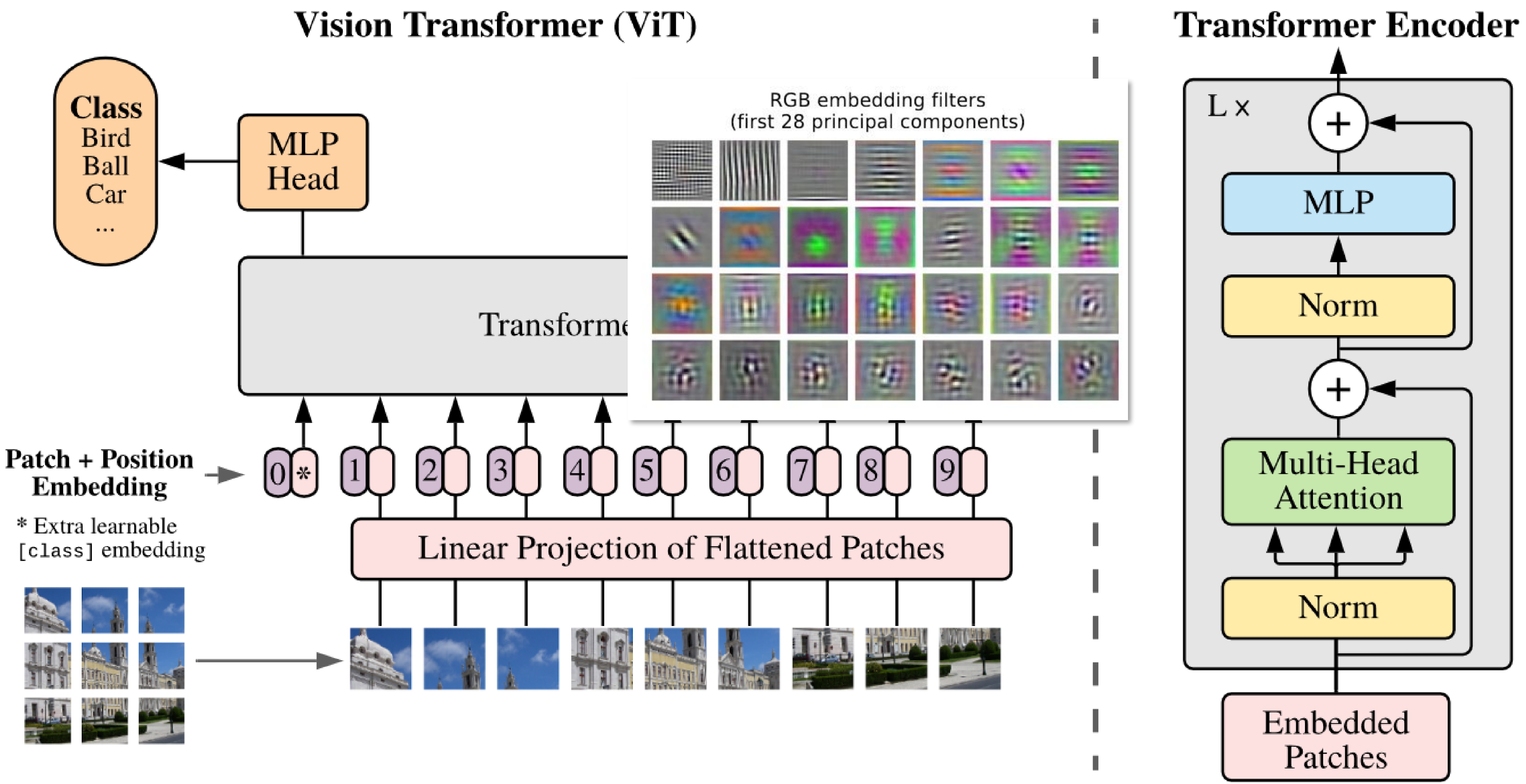
Alexey Dosovitskiy[*,†], **Lucas Beyer**[*], **Alexander Kolesnikov**[*], **Dirk Weissenborn**[*], **Xiaohua Zhai**[*], **Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby**[*,†]
[*]equal technical contribution, [†]equal advising
Google 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)
Transformer Encoder
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
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

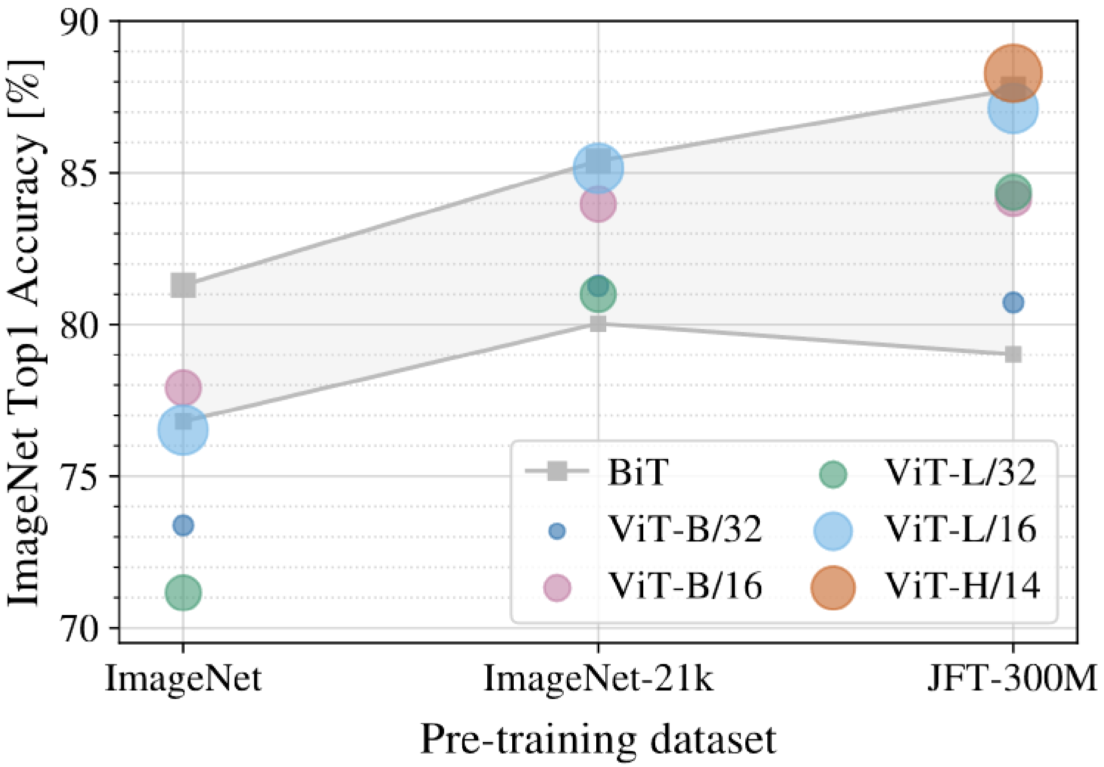


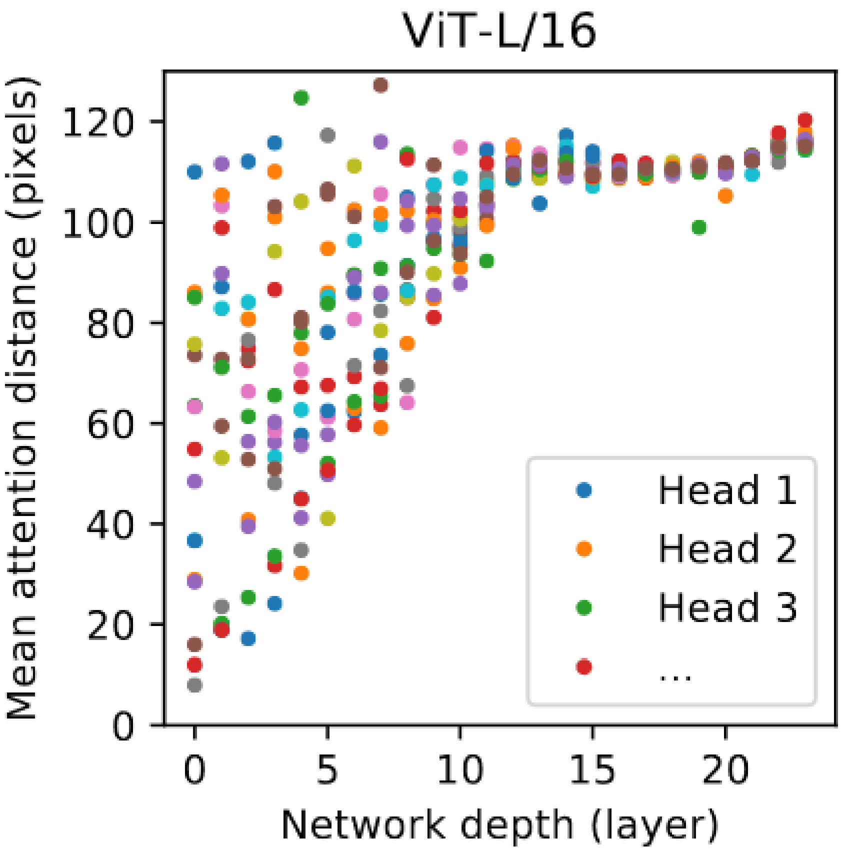
Figure 3: Transfer to ImageNet. While large ViT models perform worse than BiT ResNets (shaded area) when pre-trained on small datasets, they shine when pre-trained on larger datasets. Similarly, larger ViT variants overtake smaller ones as the dataset grows.

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.

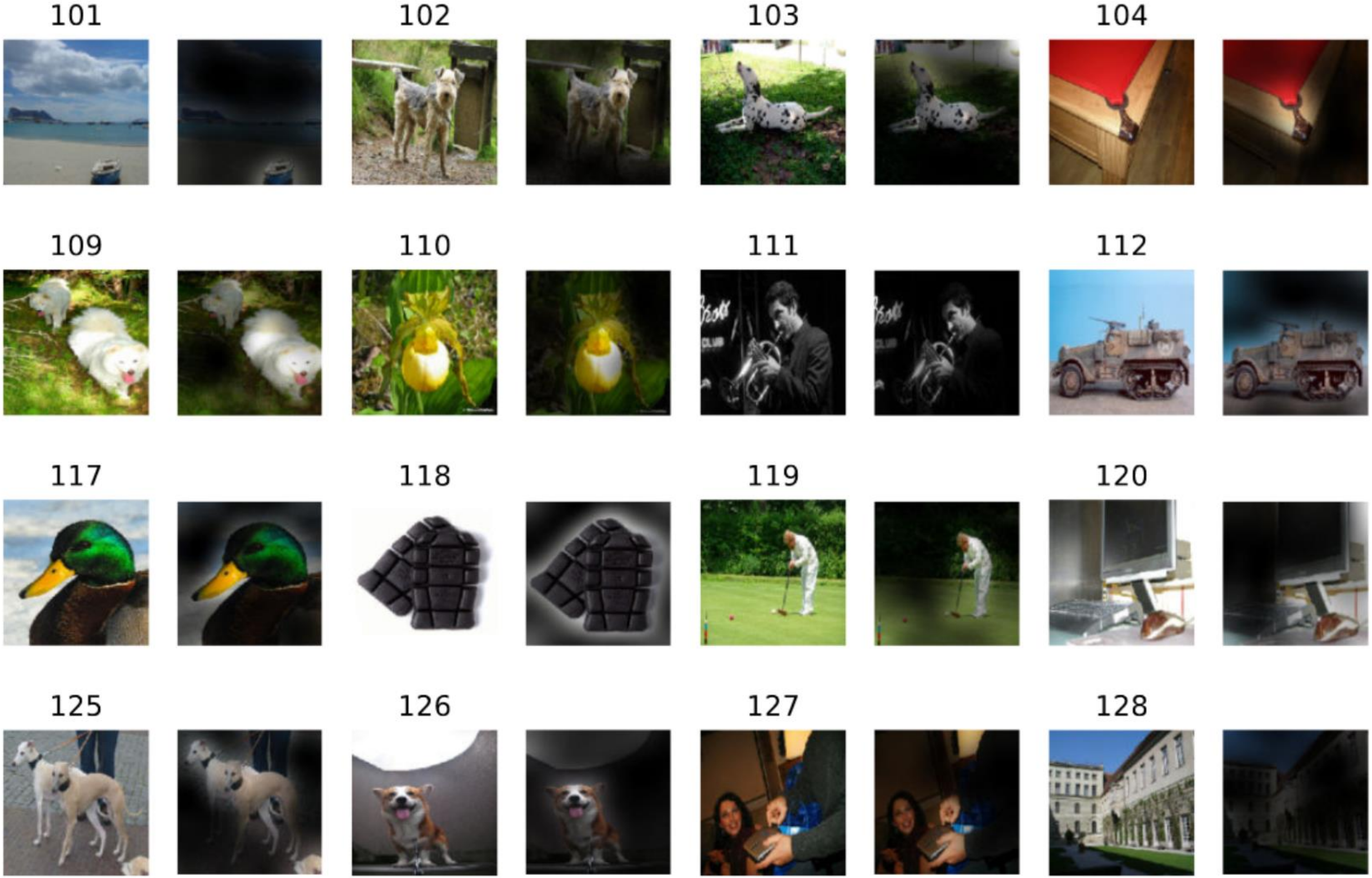
However, the picture changes if the models are trained on larger datasets (14M-300M images). We find that large scale training trumps inductive bias.

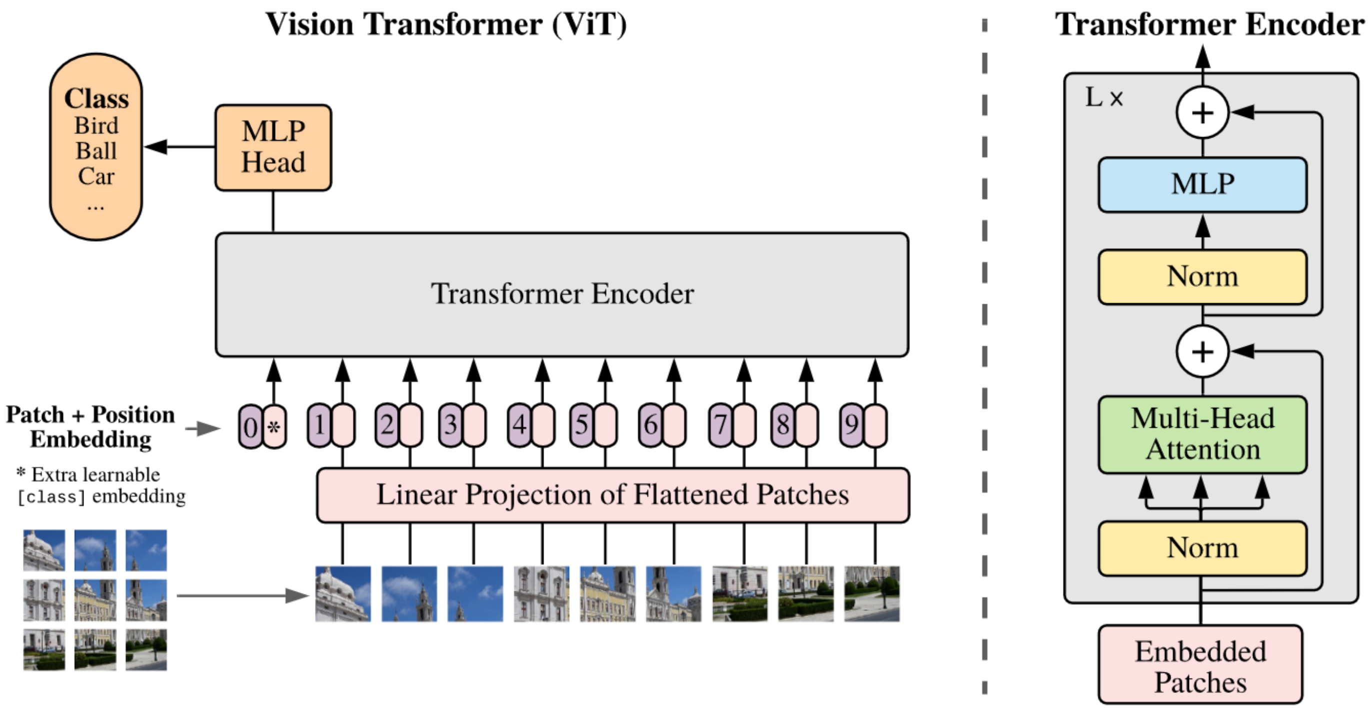
Dosovitskiy et al.

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

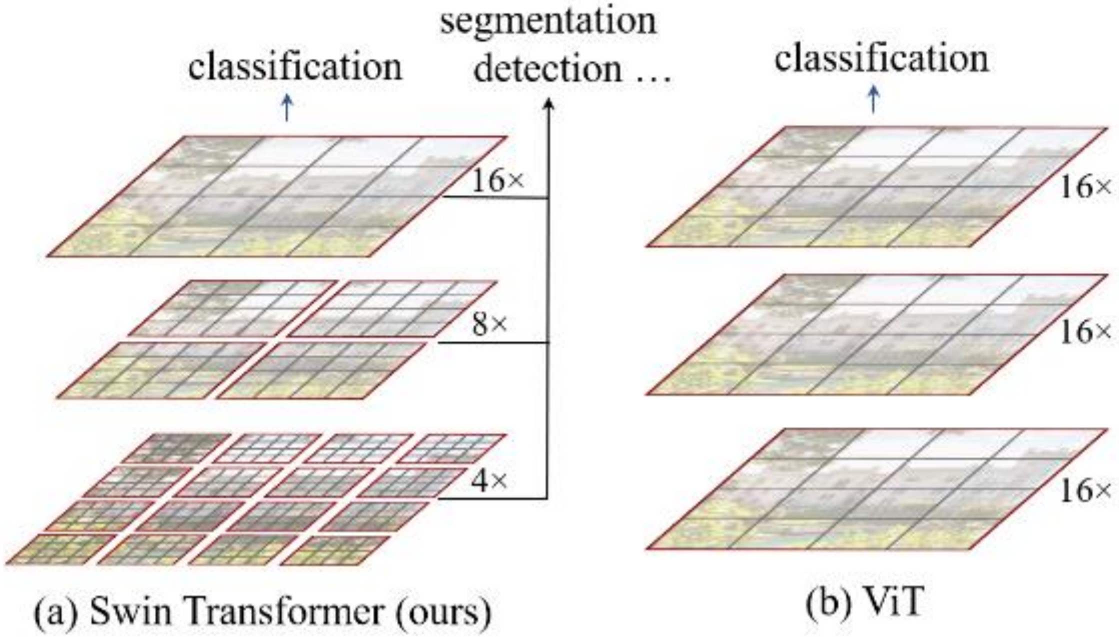


ViT-L/16
Mean attention distance (pixels)
Network depth (layer)
Head 1
Head 2
Head 3
...
0
20
40
60
80
100
120
0
5
10
15
20


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



- This can’t be ideal, right?



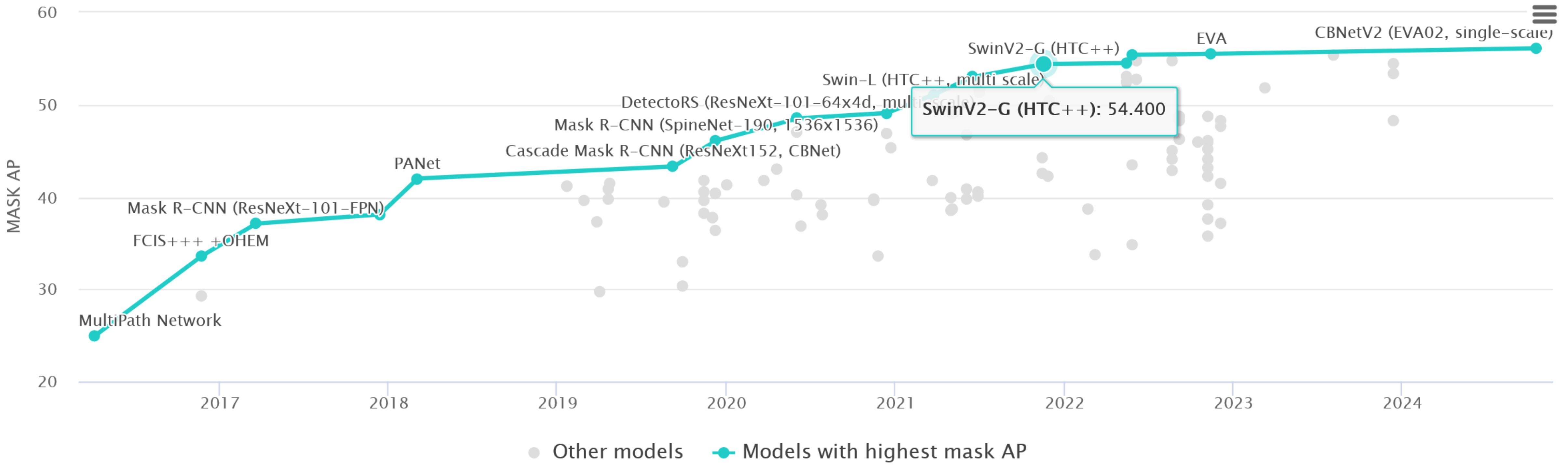
Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

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

Instance Segmentation on COCO test-dev

Leaderboard Dataset

View mask AP by Date for All models



https://paperswithcode.com/sota/instance-segmentation-on-coco

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.

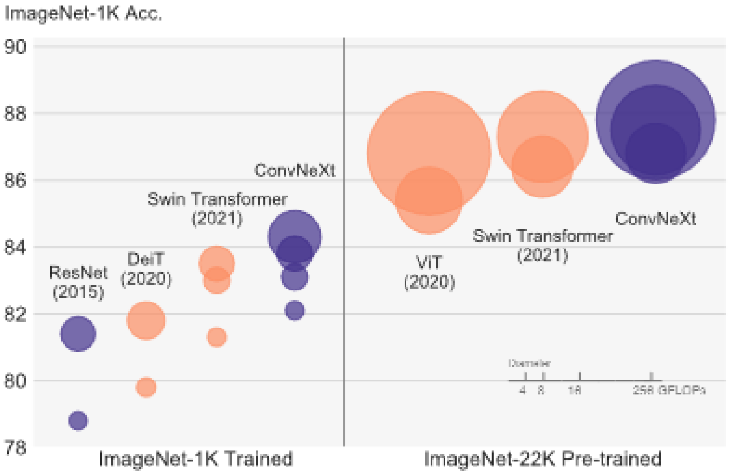


Figure 1. **ImageNet-1K classification** results for • ConvNets and ○ vision Transformers. Each bubble's area is proportional to FLOPs of a variant in a model family. ImageNet-1K/22K models here take 224^2/384^2 images respectively. ResNet and ViT results were obtained with improved training procedures 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.

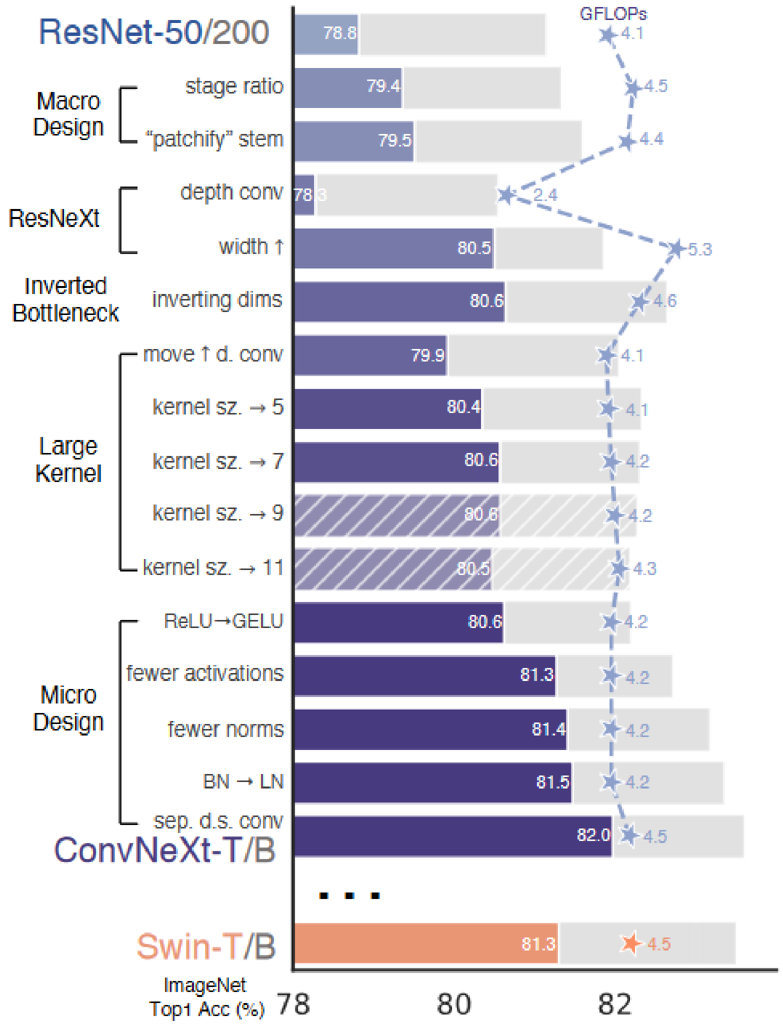


Figure 2. We modernize a standard ConvNet (ResNet) towards the design of a hierarchical vision Transformer (Swin), without introducing any attention-based modules. The foreground bars are model accuracies in the ResNet-50/Swin-T FLOP regime; results for the ResNet-200/Swin-B regime are shown with the 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

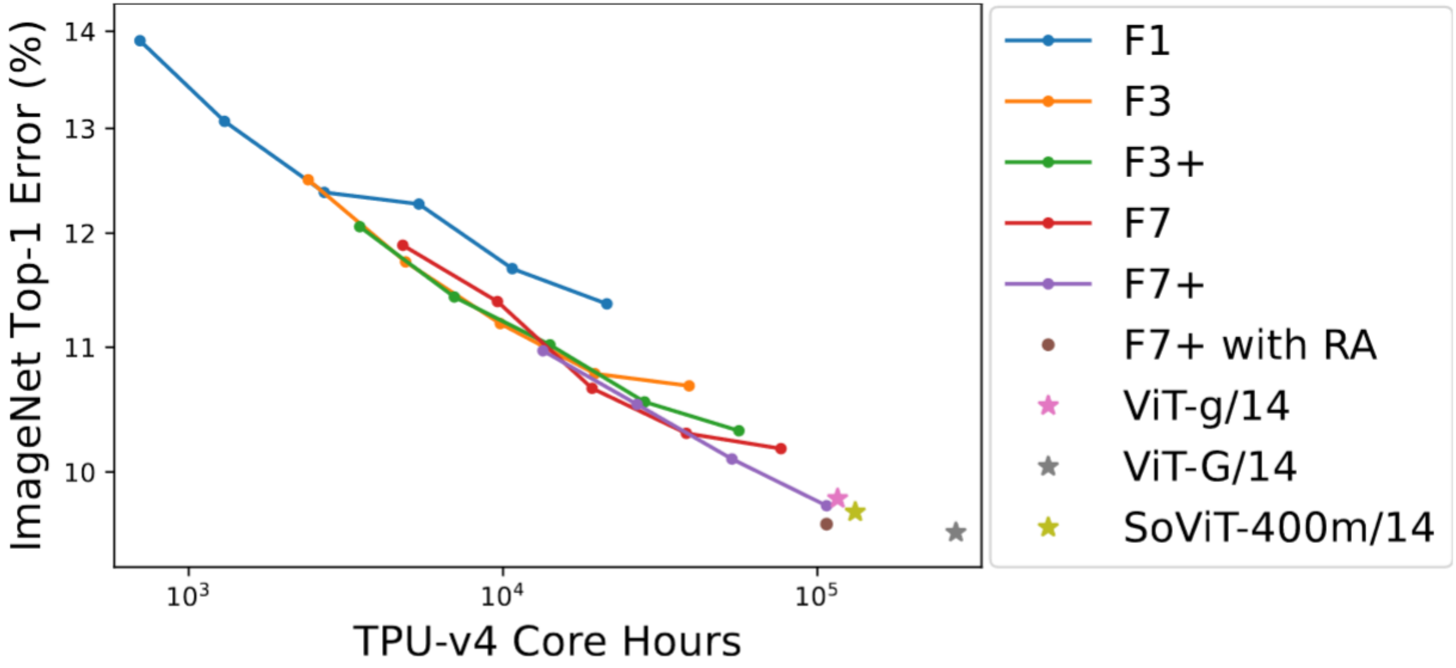
Table 3. **COCO object detection and segmentation results** using Mask-RCNN and Cascade Mask-RCNN. ‡ indicates that the model is pre-trained on ImageNet-22K. ImageNet-1K pre-trained Swin results are from their Github repository [3]. AP numbers of the ResNet-50 and X101 models are from [45]. We measure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

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



CLIP

Learning Transferable Visual Models From Natural Language Supervision

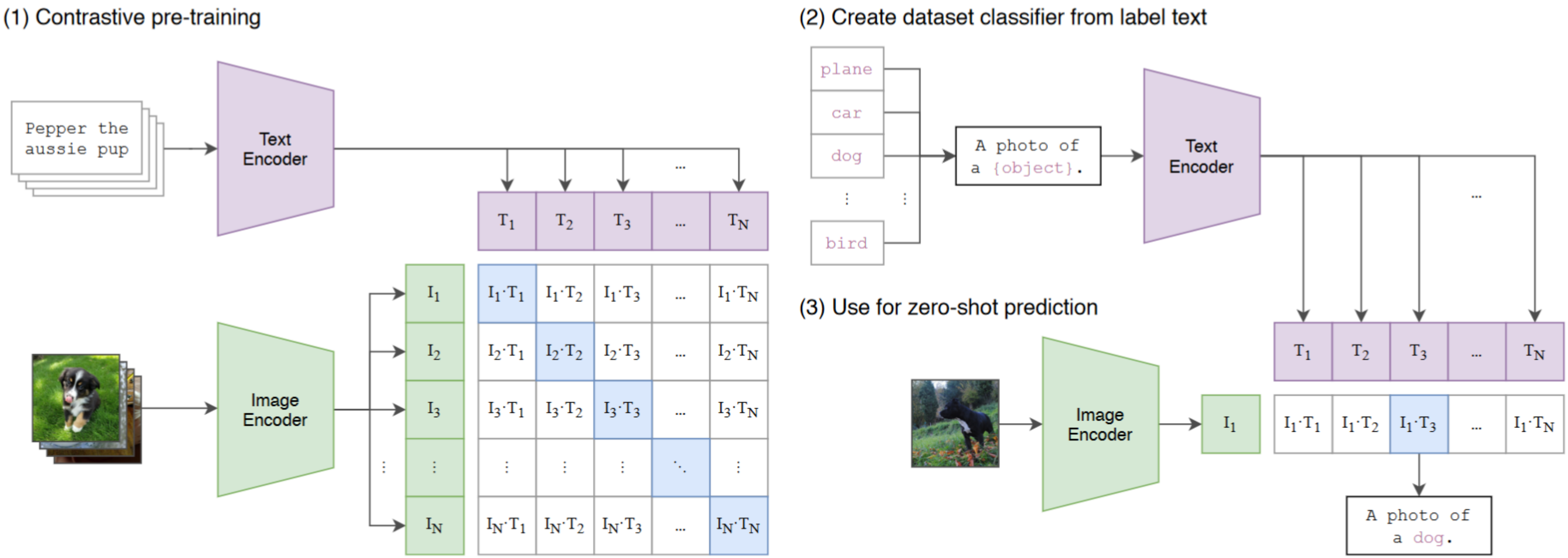
Alec Radford [*1] Jong Wook Kim [*1] Chris Hallacy [1] Aditya Ramesh [1] Gabriel Goh [1] Sandhini Agarwal [1]
Girish Sastry [1] Amanda Askell [1] Pamela Mishkin [1] Jack Clark [1] Gretchen Krueger [1] Ilya Sutskever [1]

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-

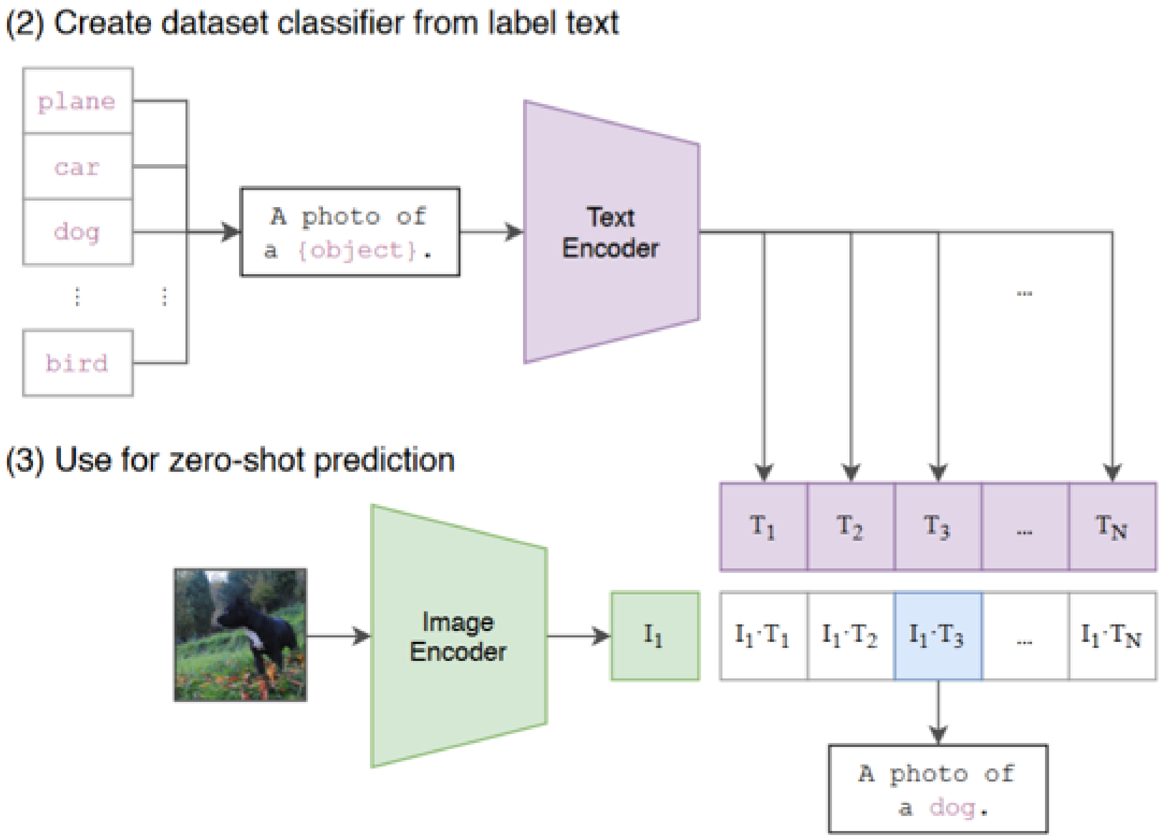
Task-agnostic objectives such as autoregressive and masked language modeling have scaled across many orders of magnitude in compute, model capacity, and data, steadily improving capabilities. The development of "text-to-text" as a standardized input-output interface (McCann et al., 2018; Radford et al., 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.

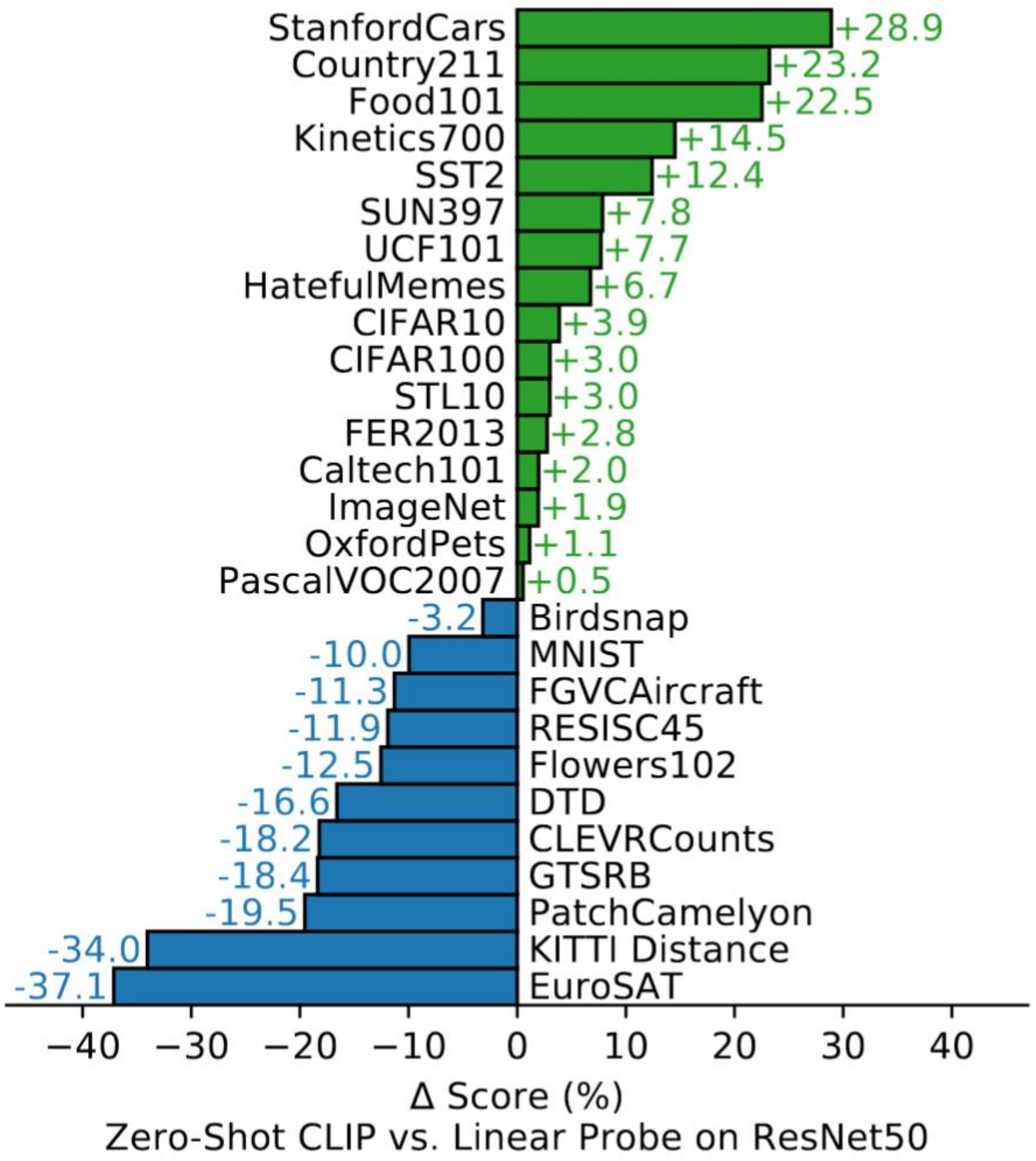
These results suggest that the aggregate supervision accessible to modern pre-training methods within web-scale collections of text surpasses that of high-quality crowd-labeled NLP datasets. However, in other fields such as computer vision it is still standard practice to pre-train models on crowd-labeled 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.

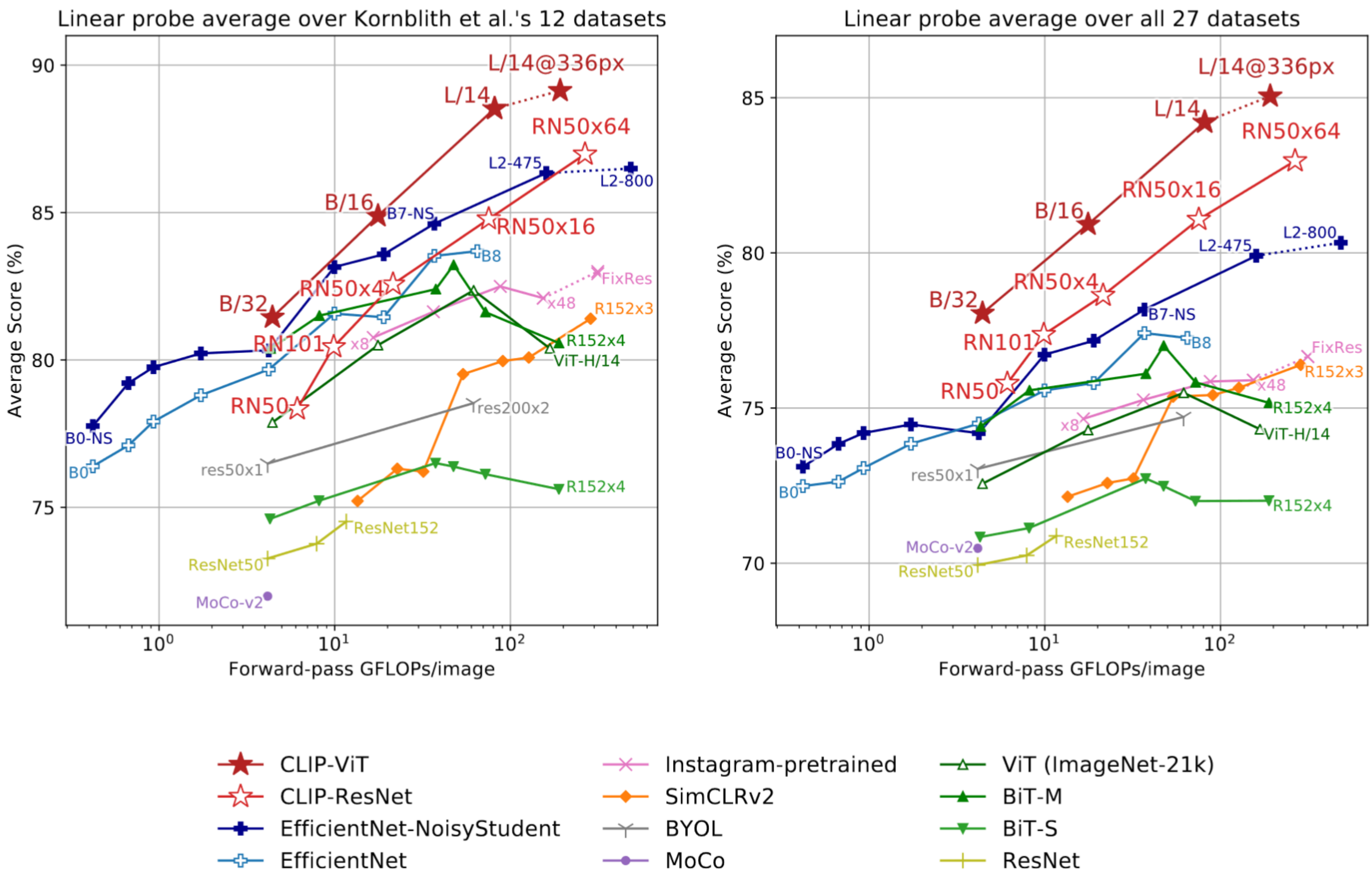


- “weakly supervised” contrastive learning on 400M text / image pairs
- ViT architecture trained from scratch

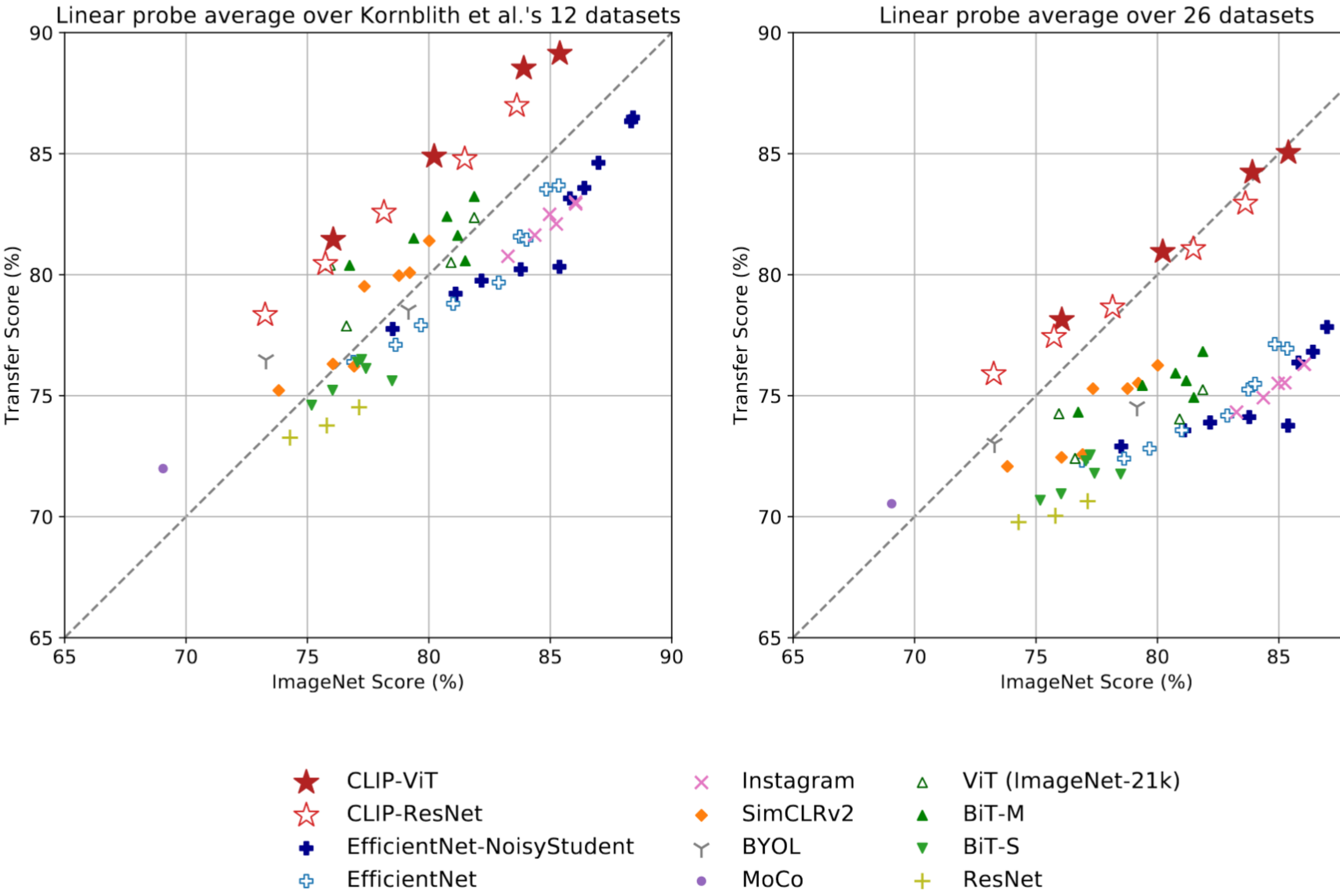
Dataset	Classes	Train size	Test size	Evaluation metric
Food-101	102	75,750	25,250	accuracy
CIFAR-10	10	50,000	10,000	accuracy
CIFAR-100	100	50,000	10,000	accuracy
Birdsnap	500	42,283	2,149	accuracy
SUN397	397	19,850	19,850	accuracy
Stanford Cars	196	8,144	8,041	accuracy
FGVC Aircraft	100	6,667	3,333	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
(3) Use for zero-shot prediction
Image Encoder
I_1
T_1 T_2 T_3 ... T_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
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
Birdsnap -3.2
MNIST -10.0
FGVCAircraft -11.3
RESISC45 -11.9
Flowers102 -12.5
DTD -16.6
CLEVRCounts -18.2
GTSRB -18.4
PatchCamelyon -19.5
KITTI Distance -34.0
EuroSAT -37.1
−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
RN101
RN50
B/16
B/32
L2-475
L2-800
B7-NS
B0-NS
B8
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


Linear probe average over Kornblith et al.'s 12 datasets
Linear probe average over 26 datasets
Transfer Score (%)
ImageNet Score (%)
65
70
75
80
85
90
CLIP-ViT
CLIP-ResNet
EfficientNet-NoisyStudent
EfficientNet
Instagram
SimCLRv2
BYOL
MoCo
ViT (ImageNet-21k)
BiT-M
BiT-S
ResNet

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

				kNN	linear		
Method	Arch.	Data	Text sup.	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

Method	Arch.	ADE20k (62.9) lin.	ADE20k (62.9) +ms	CityScapes (86.9) lin.	CityScapes (86.9) +ms	Pascal VOC (89.0) lin.	Pascal VOC (89.0) +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**

(a) (b) (c) (d)

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.

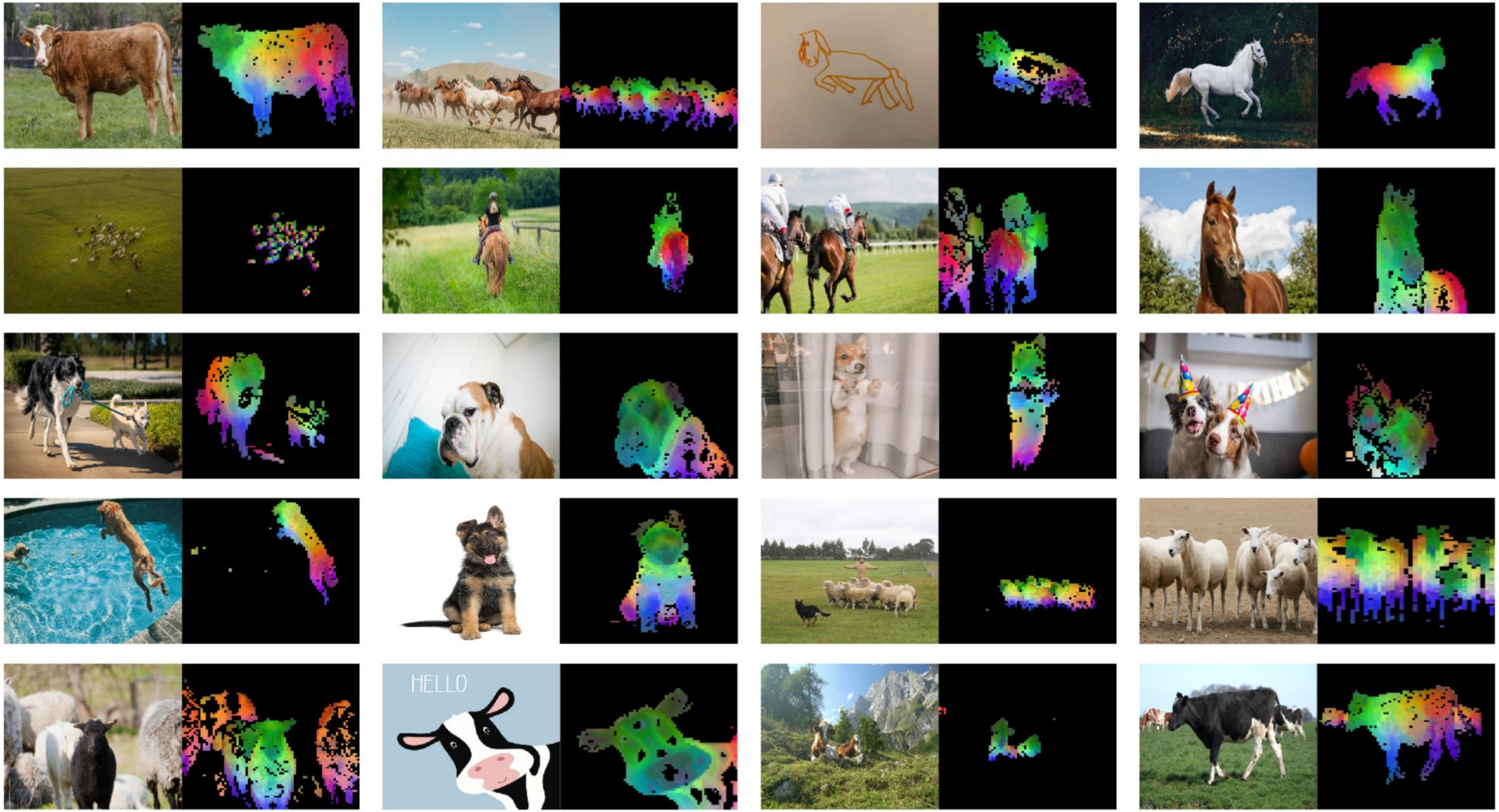


Figure 9: **More visualization of the first PCA components.** We compute the PCA between the patches from all of the images and show their first 3 components. Each component corresponds to a specific color channel. Same parts are matched between related images depsite changes of pose, style or even objects.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

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.
- 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.
- More than ever, you should start with one of these pre-trained models – CLIP if you want language support, DinoV2 if you want spatial reasoning