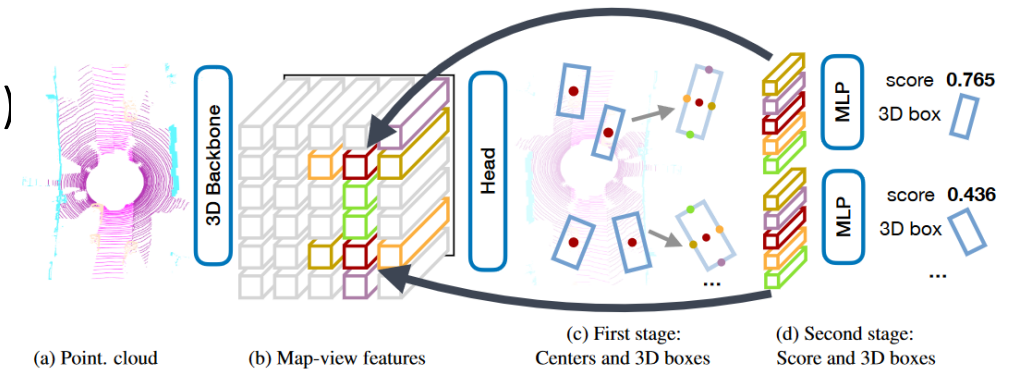


# “Attention” and “Transformer” Architectures

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 (LaserNet)
  - Project 3D data into fixed 2D views (MVCNN)
- With lidar, multi-modal approaches (adding images, radar) help surprisingly little compared to lidar-only approaches (~3 mAP).
- These alternate representations might be applicable more broadly, e.g. reasoning about depth estimates might be easier in bird's eye view (PseudoLidar)



CenterPoint, which is near state of the art for 3D object detection when using VoxelNet backbone.

Center-based 3D Object Detection and Tracking. Tianwei Yin, Xingyi Zhou, Philipp Krähenbühl. CVPR 2021

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

# Outline

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



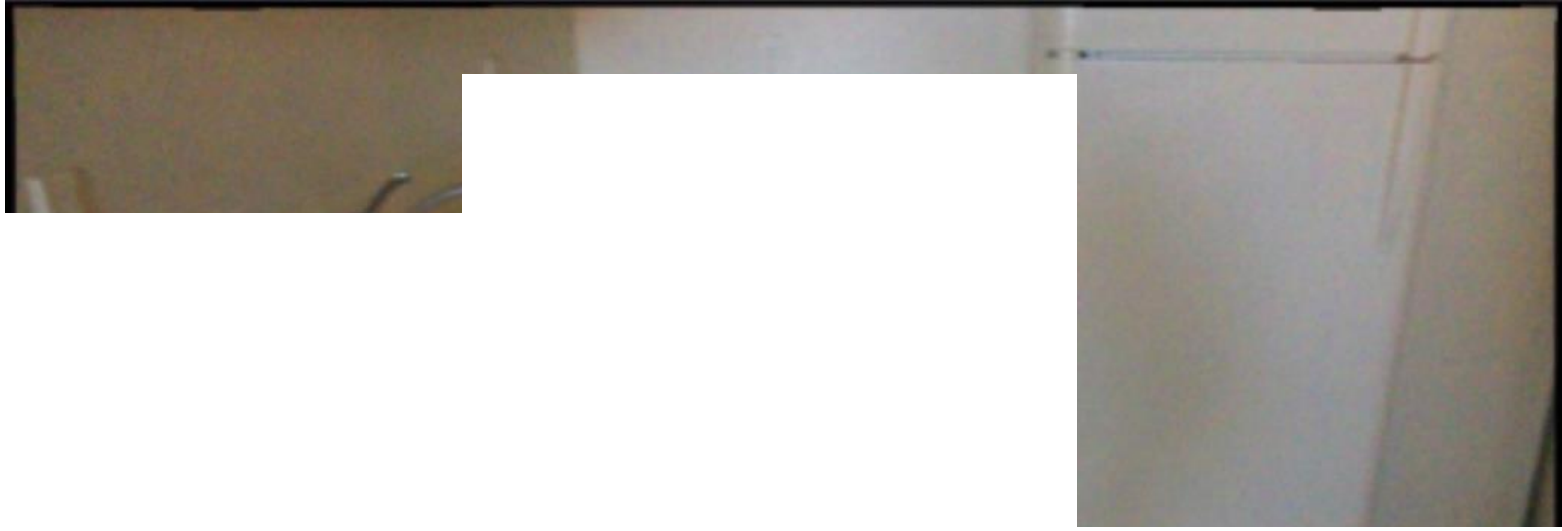


Ground truth



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**

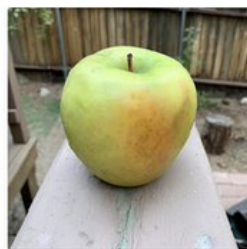
@moyix

The latest generation of adversarial image attacks is, uh, somewhat simpler to carry out [openai.com/blog/multimoda...](https://openai.com/blog/multimodal-attacks)

### 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,<sup>22</sup> 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.



**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

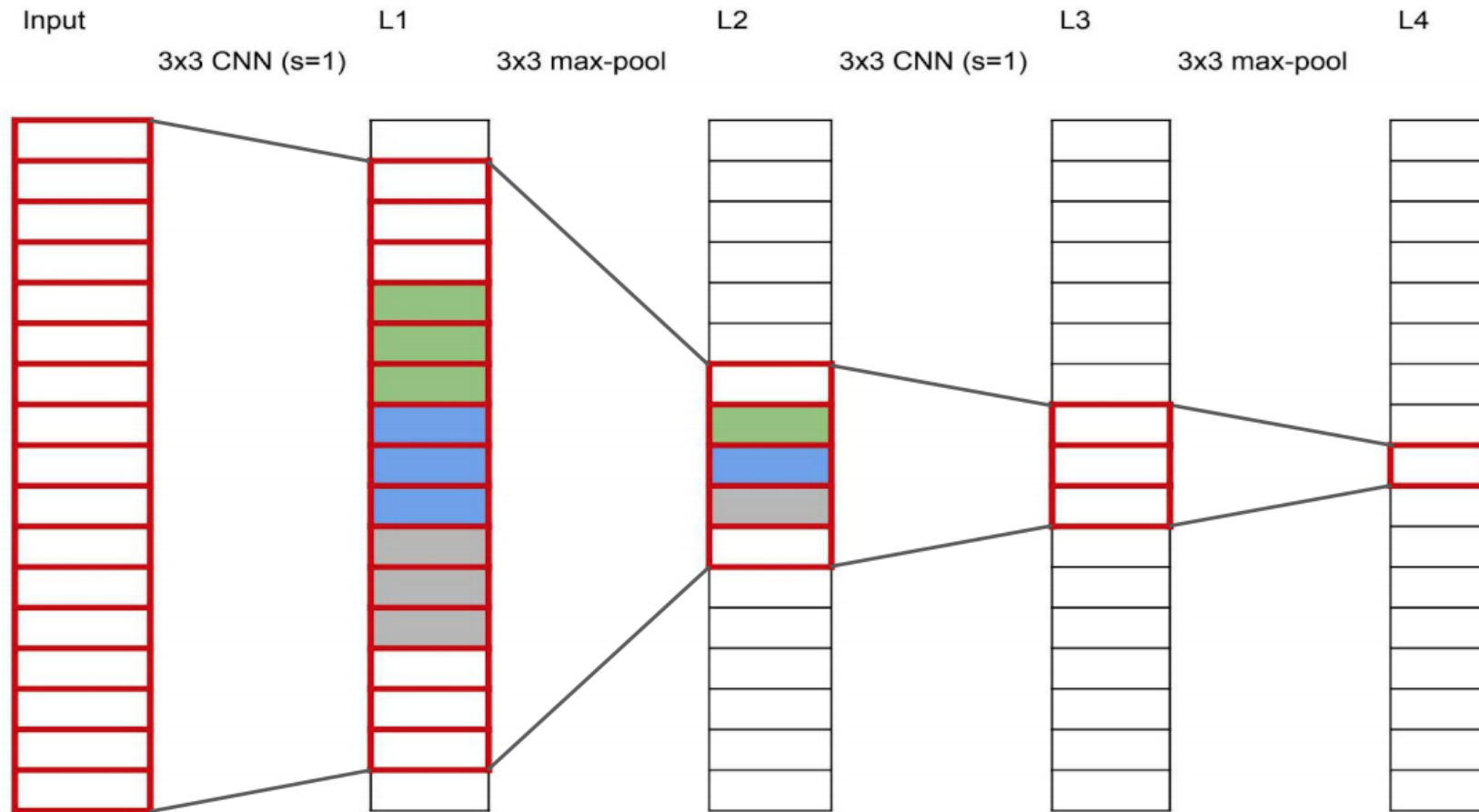


9:28 PM · Mar 4, 2021 · Twitter for iPad

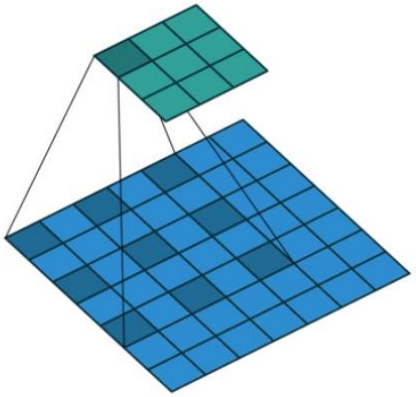
So how do we fix these problems?



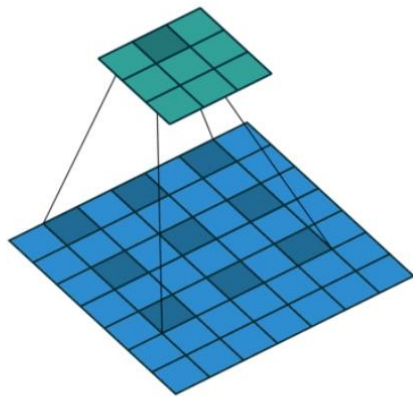
# Receptive field



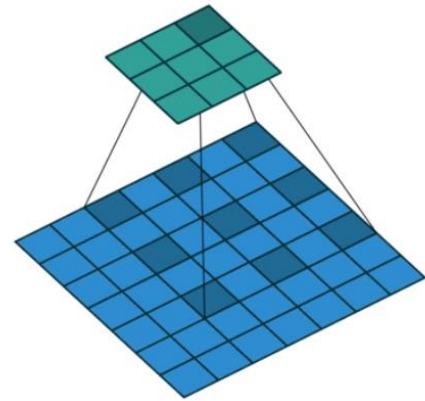
# Dilated Convolution



No padding, no stride, dilation



No padding, no stride, dilation



No padding, no stride, dilation



# Receptive field could also be an issue in 3D

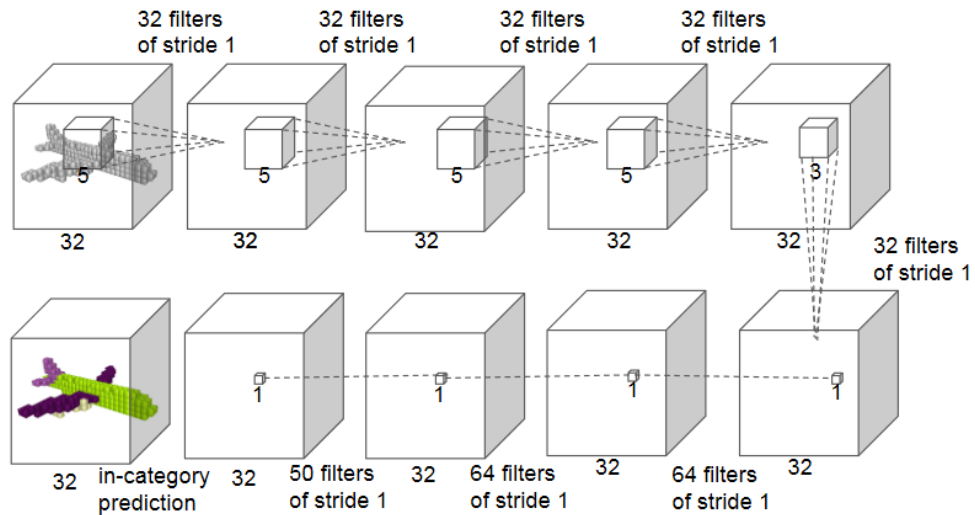


Figure 10. **Baseline 3D CNN segmentation network.** The network is fully convolutional and predicts part scores for each voxel.

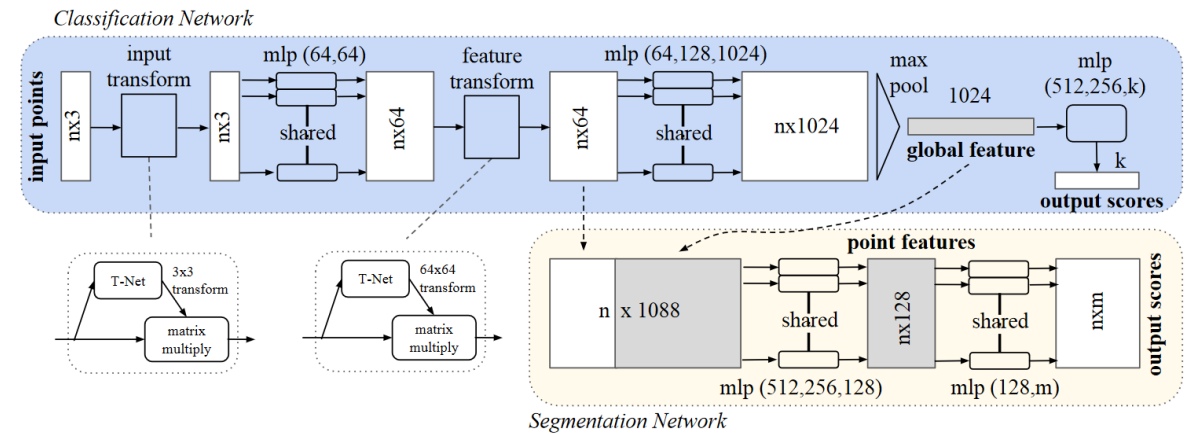


Figure 2. **PointNet Architecture.** The classification network takes  $n$  points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for  $k$  classes. The segmentation network is an extension to the classification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

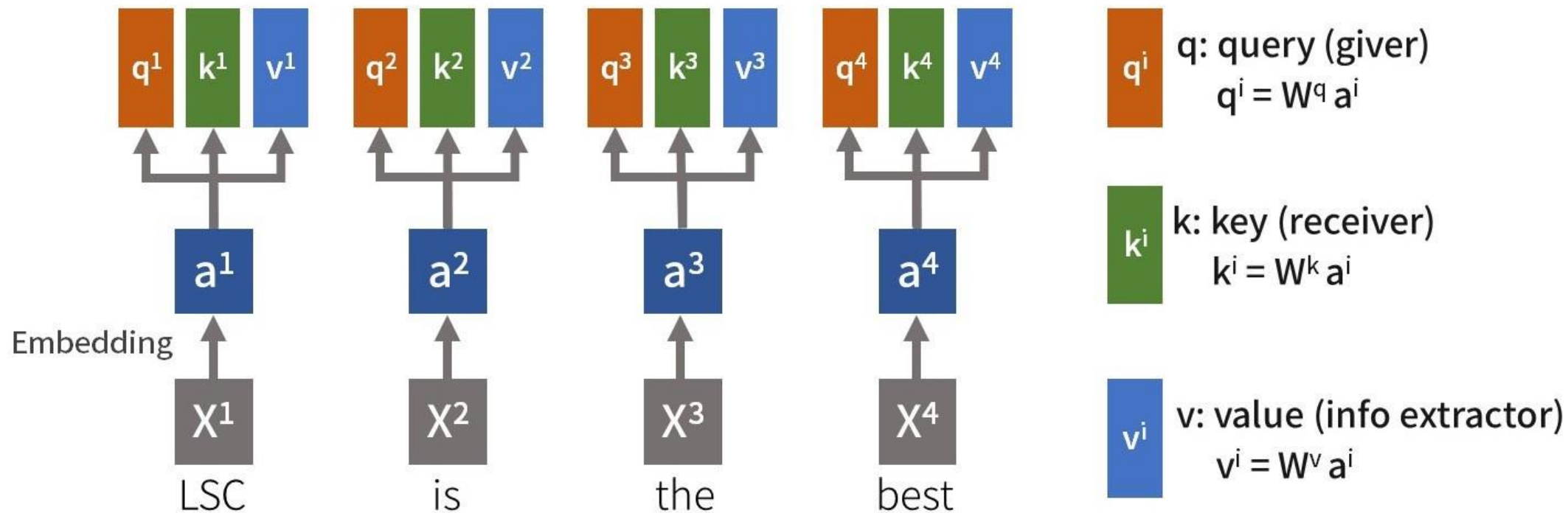
**Łukasz Kaiser\***  
Google Brain  
lukaszkaizer@google.com

**Illia Polosukhin\* †**  
illia.polosukhin@gmail.com

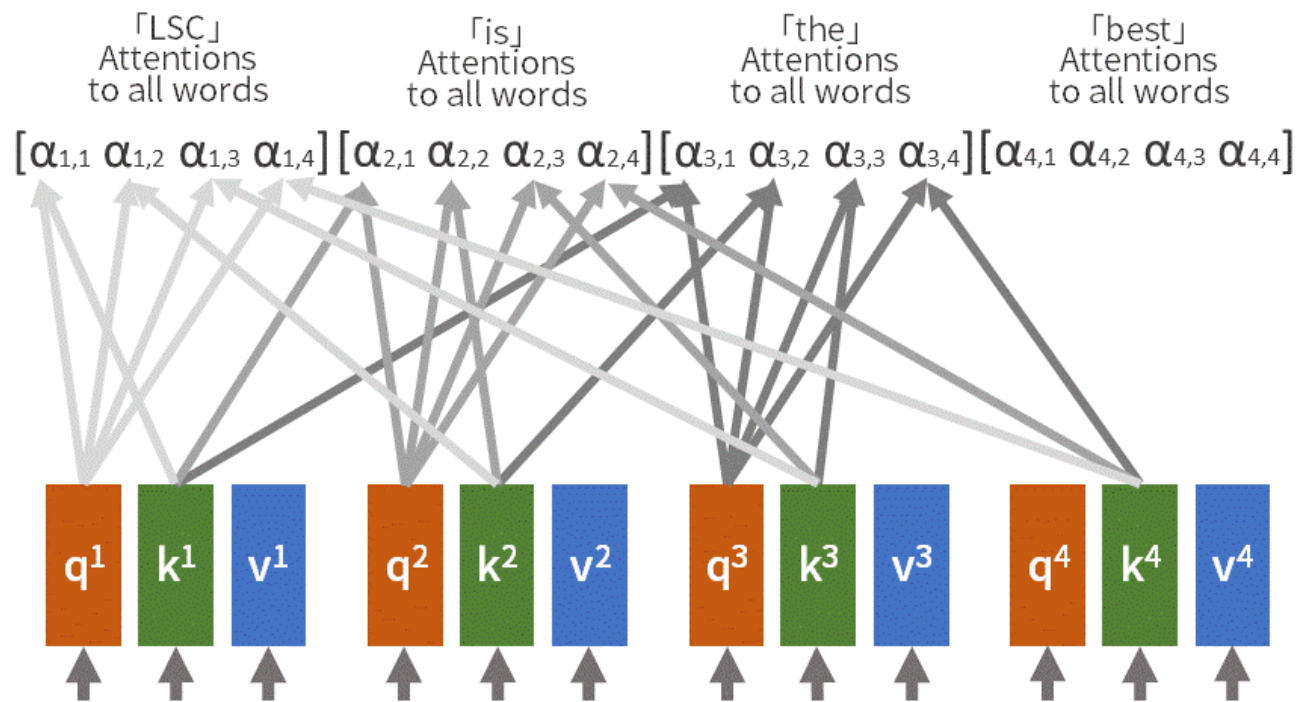
## Abstract

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, based on the self-attention mechanism.

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



**Input: LSC is the best!**

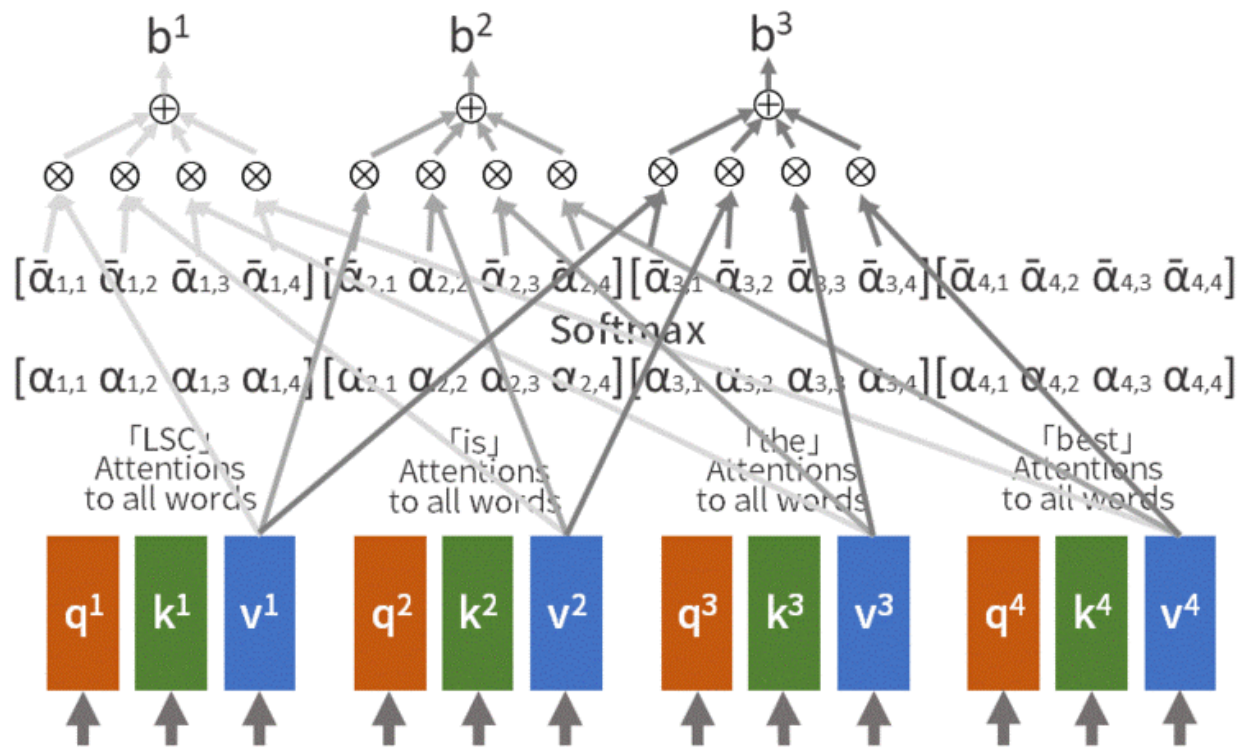


$$\alpha_{i,j} = \frac{q^i \cdot k^j}{\sqrt{d}}$$

d: dimension of q, k

Attention Matrix

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



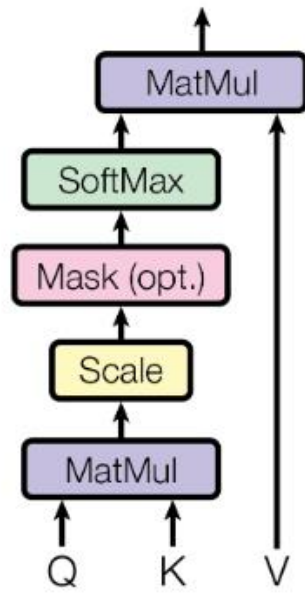
$$b^i = \sum_j \bar{\alpha}_{i,j} v^j$$

# Complexity Comparison

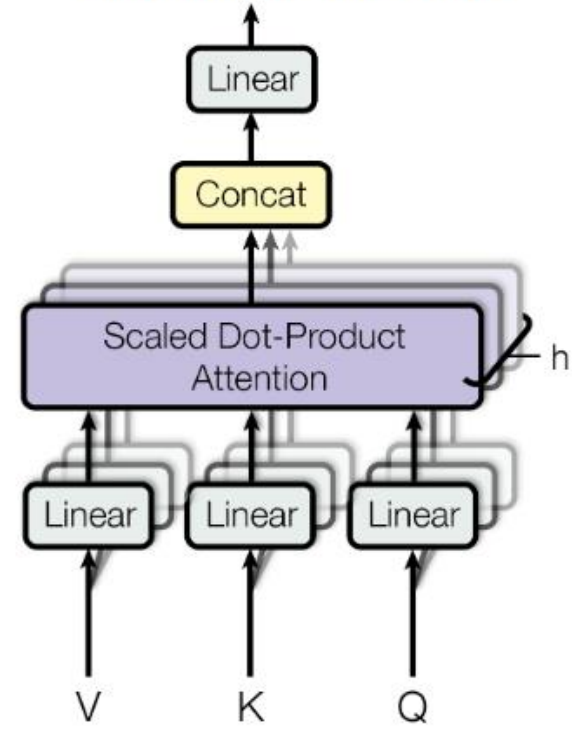
Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(\log_k(n))$



Scaled Dot-Product Attention



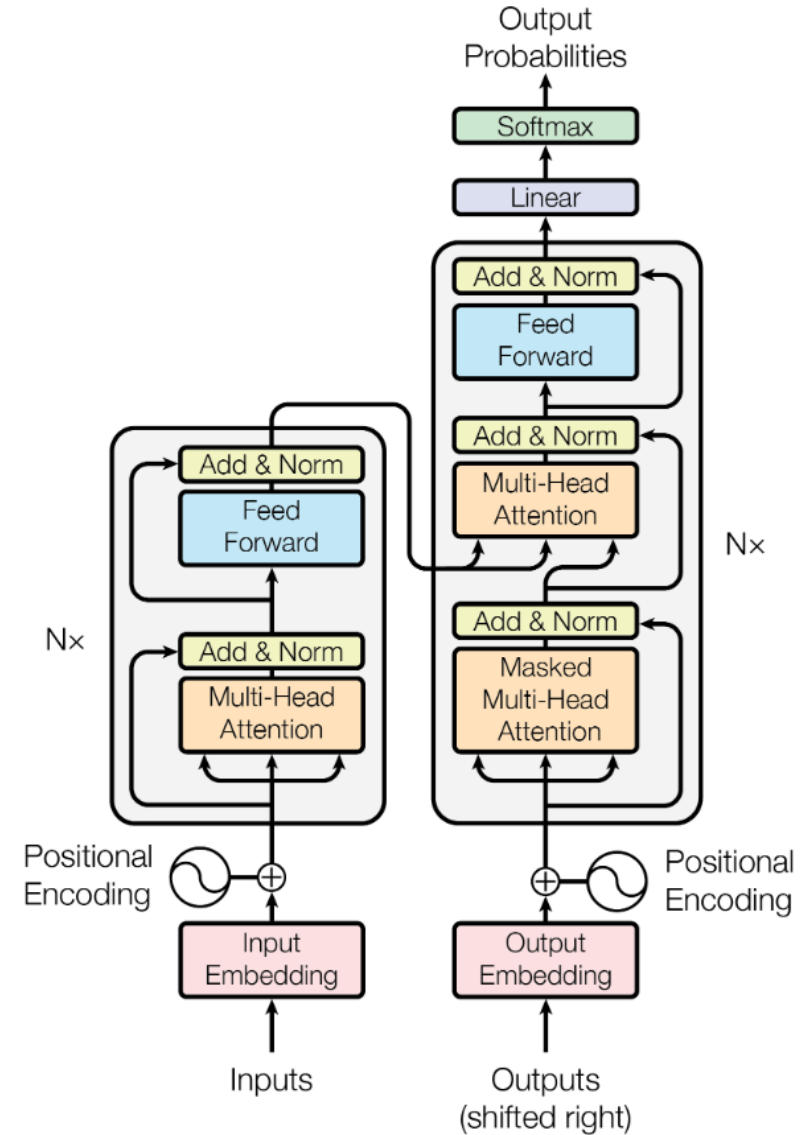
Multi-Head Attention



# 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	<b>41.29</b>	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	<b><math>3.3 \cdot 10^{18}</math></b>	
Transformer (big)	<b>28.4</b>	<b>41.8</b>	$2.3 \cdot 10^{19}$	



# Outline

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

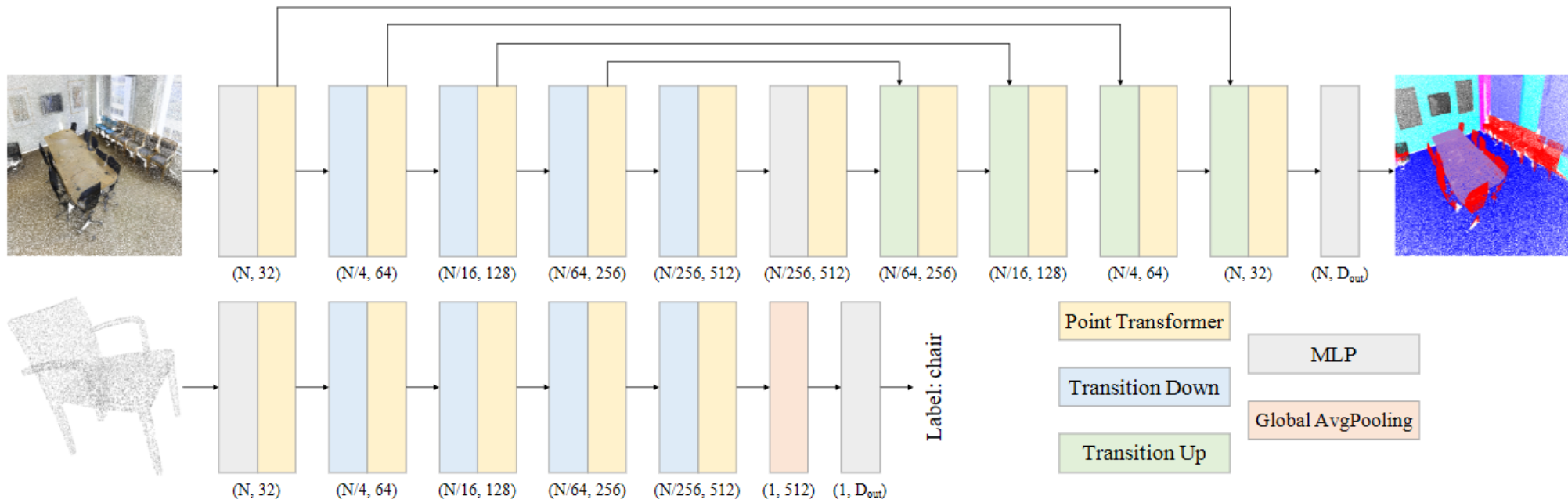
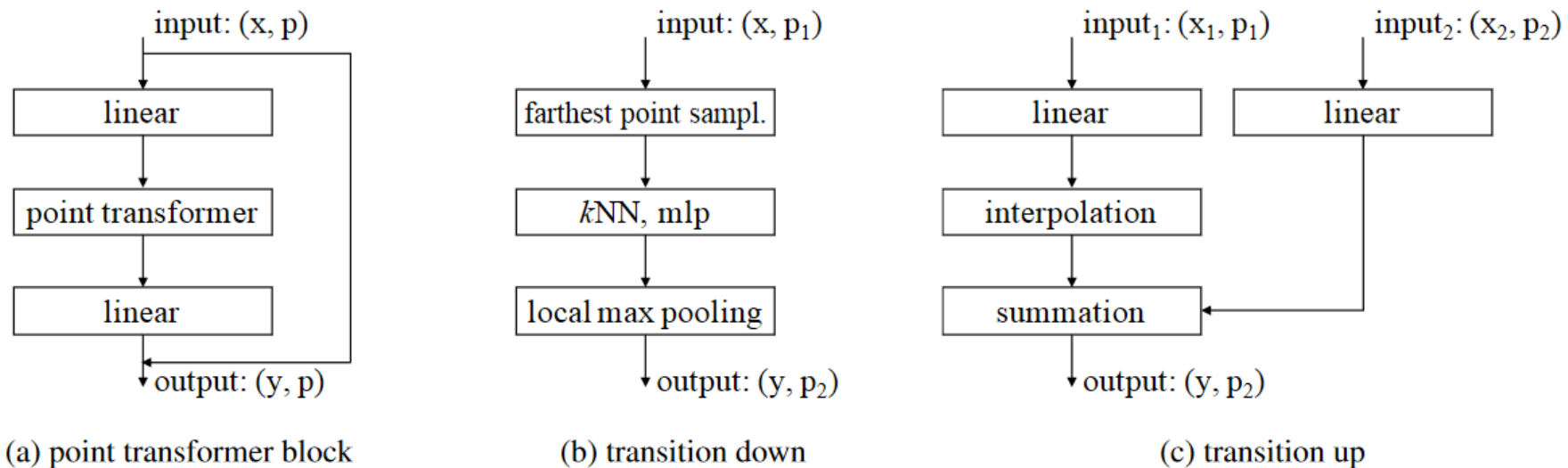


Figure 3. Point transformer networks for semantic segmentation (top) and classification (bottom).



Input

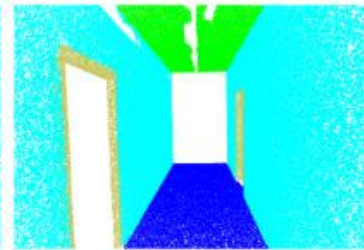
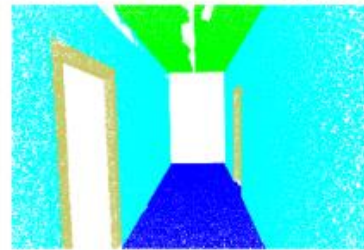
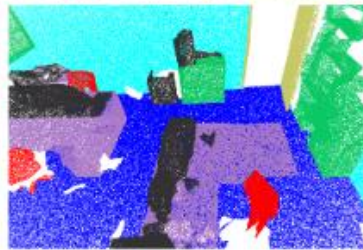
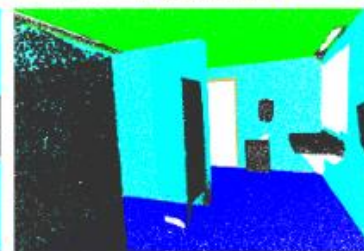
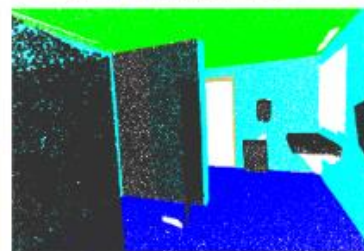
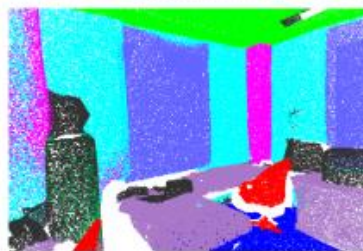
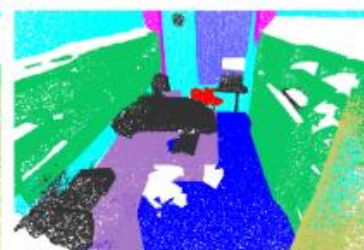
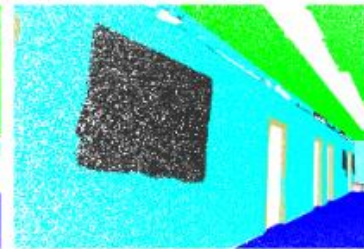
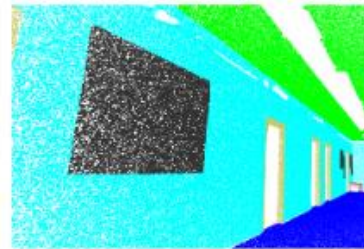
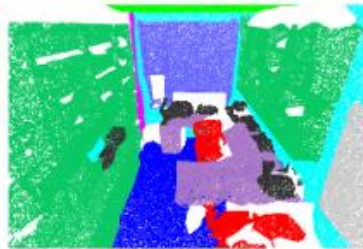
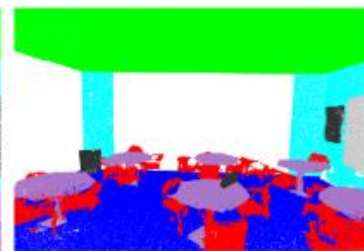
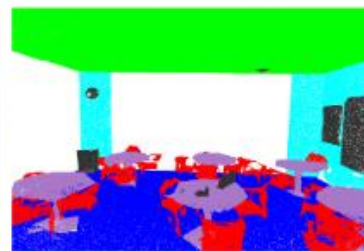
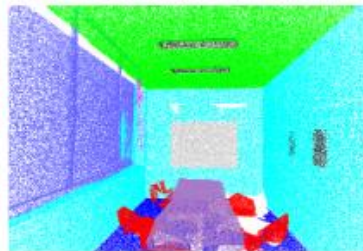
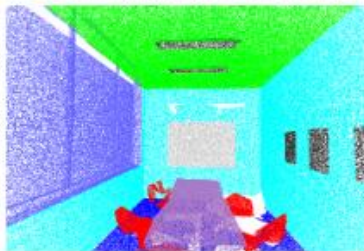
Ground Truth

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	<b>90.8</b>	<b>76.5</b>	<b>70.4</b>	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	<b>90.6</b>	<b>93.7</b>

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

Table 3. Shape classification results on the ModelNet40 dataset.

# 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<sup>\*,†</sup>, Lucas Beyer<sup>\*</sup>, Alexander Kolesnikov<sup>\*</sup>, Dirk Weissenborn<sup>\*</sup>,  
Xiaohua Zhai<sup>\*</sup>, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,  
Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby<sup>\*,†</sup>

<sup>\*</sup>equal technical contribution, <sup>†</sup>equal advising

Google Research, Brain Team

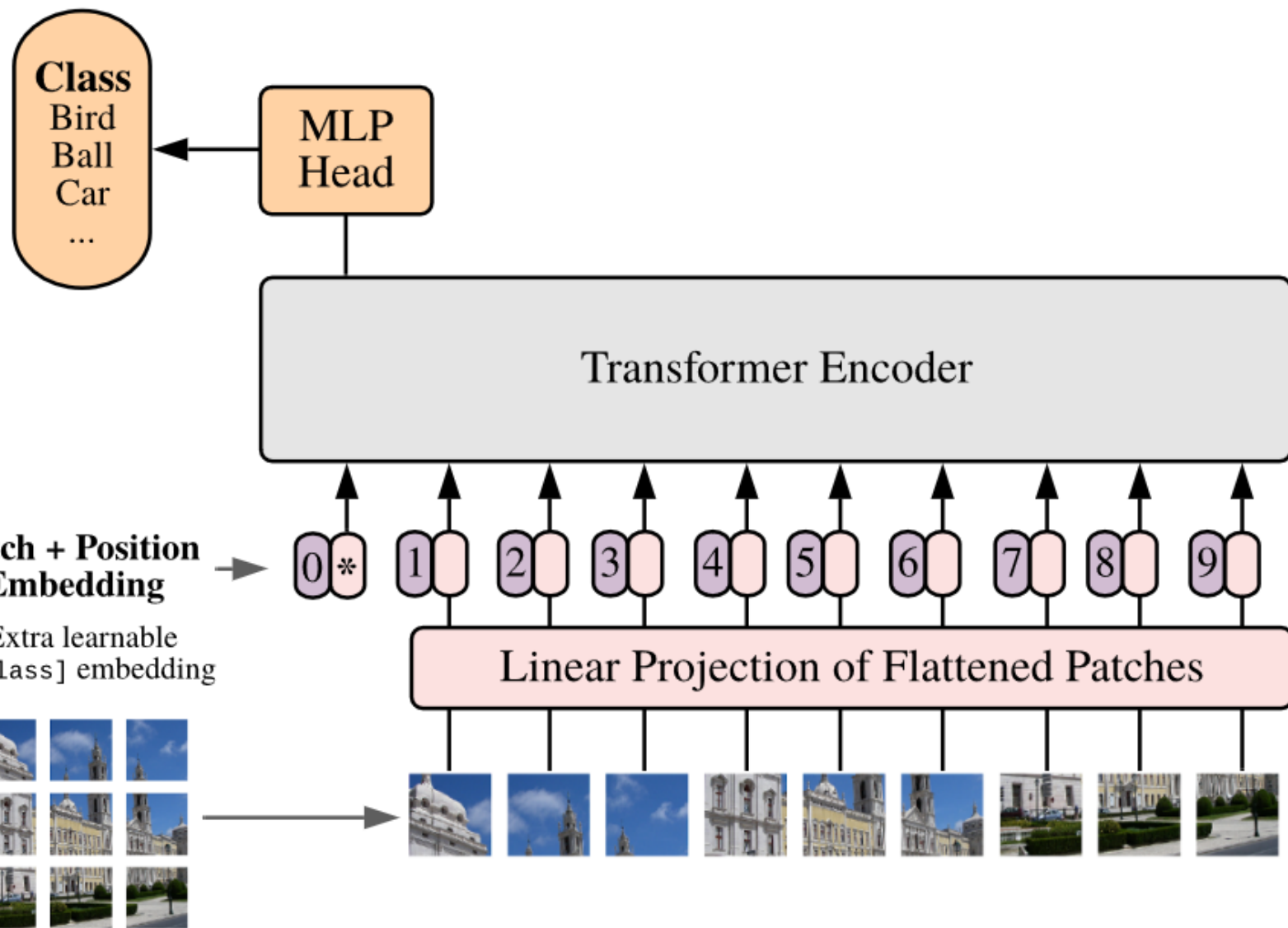
{adosovitskiy, neilhoulby}@google.com

## ABSTRACT

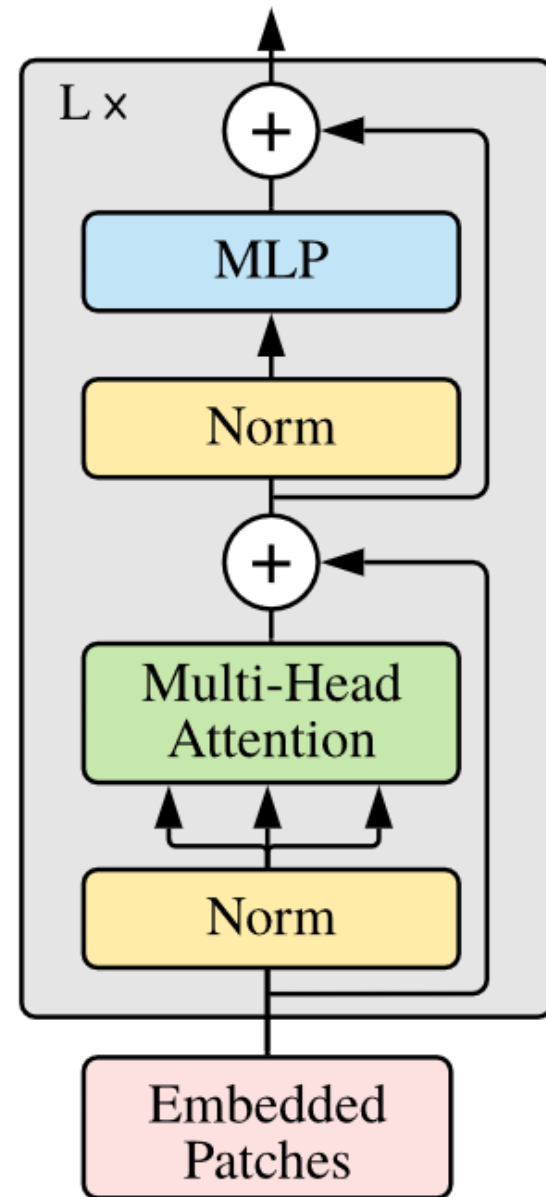
While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks 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<sup>1</sup>



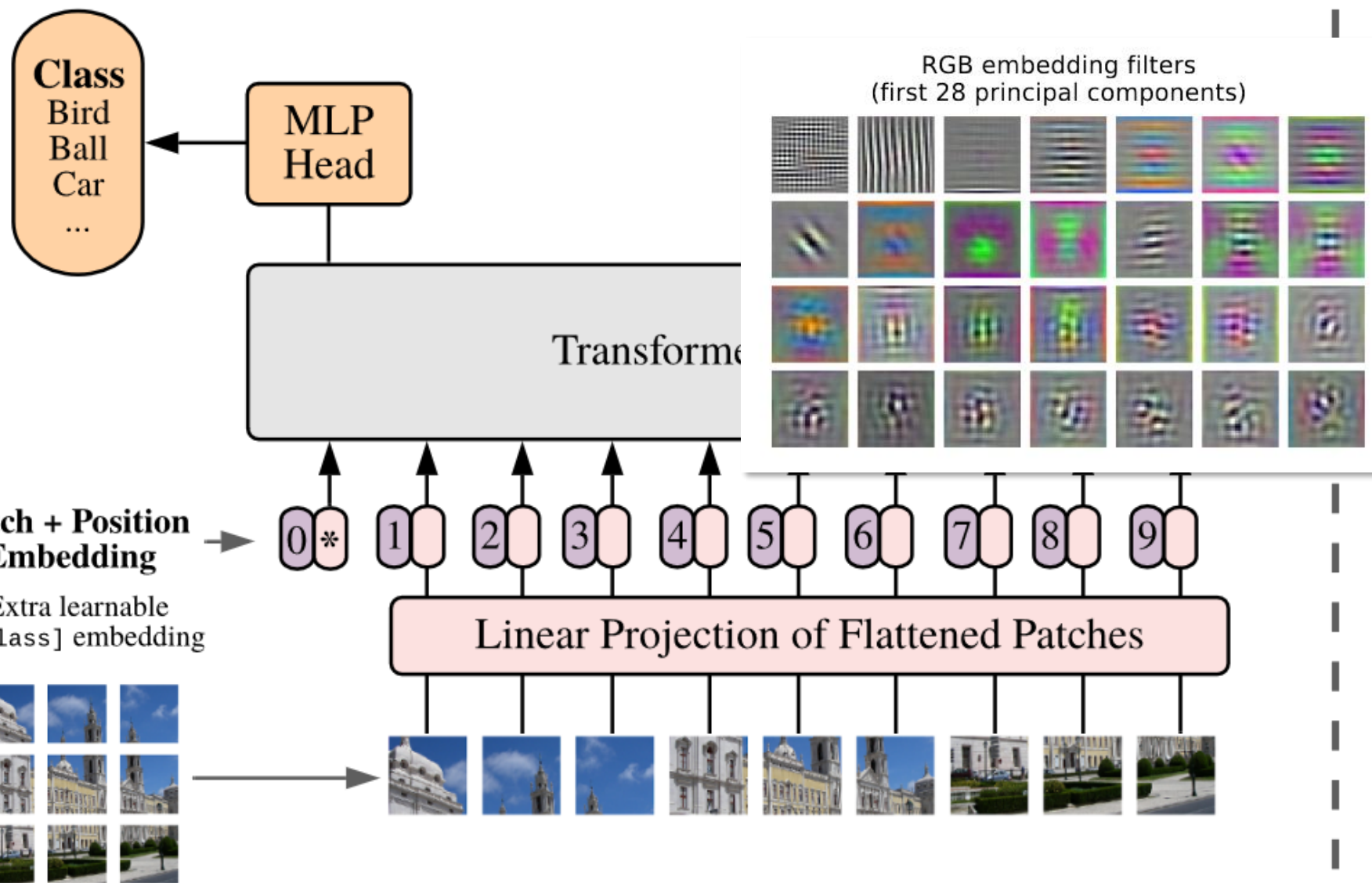
# Vision Transformer (ViT)



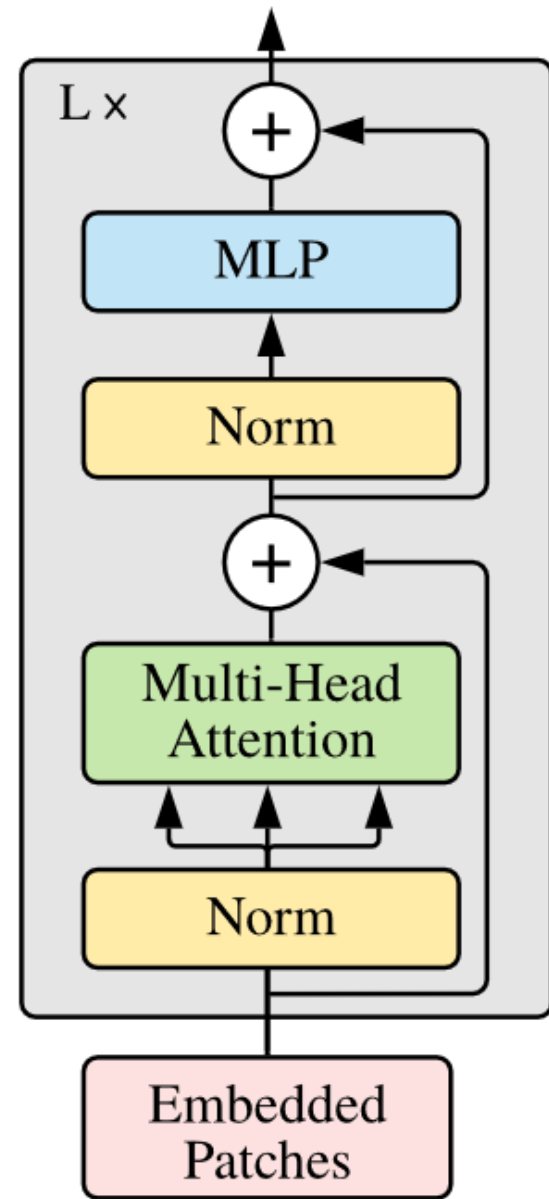
# Transformer Encoder



# Vision Transformer (ViT)



# Transformer Encoder



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	<b>88.55</b> $\pm 0.04$	87.76 $\pm 0.03$	85.30 $\pm 0.02$	87.54 $\pm 0.02$	88.4/88.5*
ImageNet ReaL	<b>90.72</b> $\pm 0.05$	90.54 $\pm 0.03$	88.62 $\pm 0.05$	90.54	90.55
CIFAR-10	<b>99.50</b> $\pm 0.06$	99.42 $\pm 0.03$	99.15 $\pm 0.03$	99.37 $\pm 0.06$	—
CIFAR-100	<b>94.55</b> $\pm 0.04$	93.90 $\pm 0.05$	93.25 $\pm 0.05$	93.51 $\pm 0.08$	—
Oxford-IIIT Pets	<b>97.56</b> $\pm 0.03$	97.32 $\pm 0.11$	94.67 $\pm 0.15$	96.62 $\pm 0.23$	—
Oxford Flowers-102	99.68 $\pm 0.02$	<b>99.74</b> $\pm 0.00$	99.61 $\pm 0.02$	99.63 $\pm 0.03$	—
VTAB (19 tasks)	<b>77.63</b> $\pm 0.23$	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

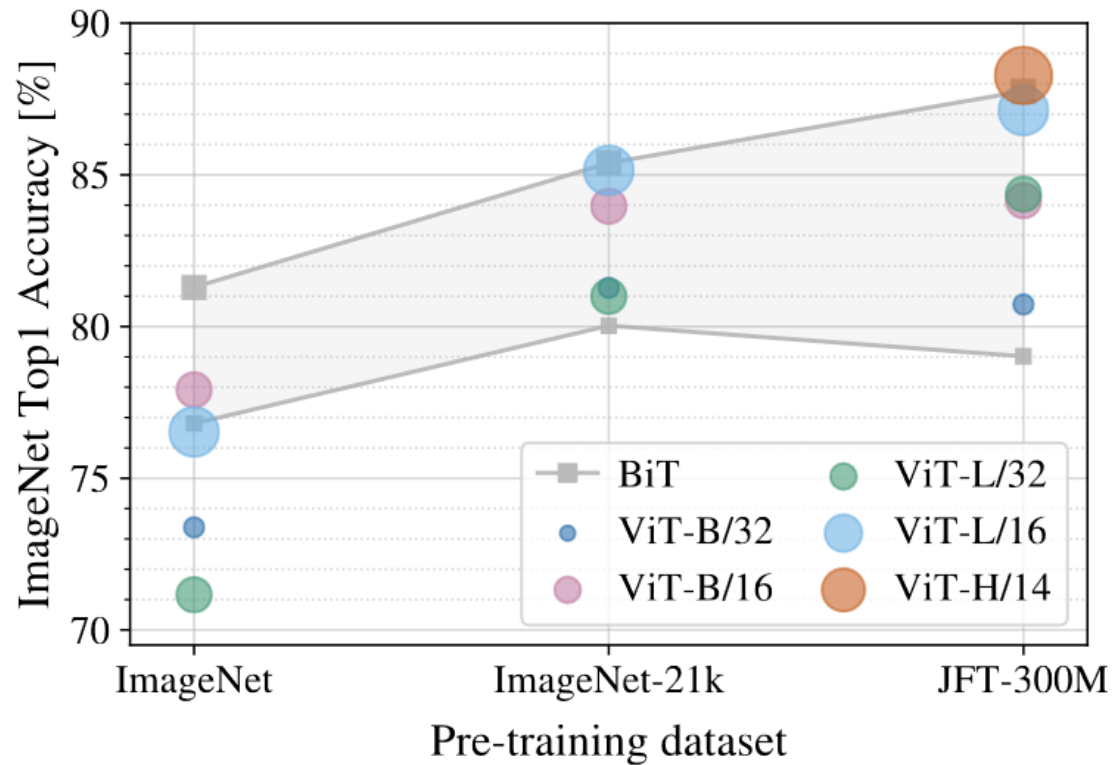


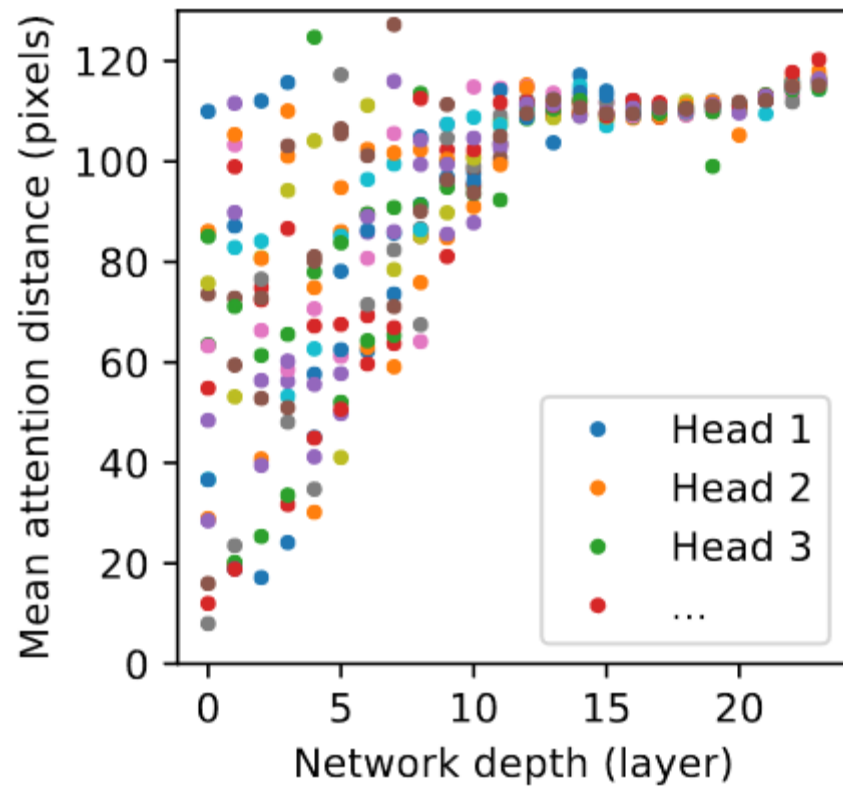
Figure 3: Transfer to ImageNet. While large ViT models perform worse than BiT ResNets (shaded area) when pre-trained on small datasets, they shine when pre-trained on larger datasets. Similarly, larger ViT variants overtake smaller ones as the dataset grows.

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.

ViT-L/16



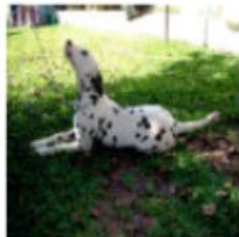
101



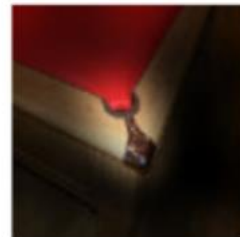
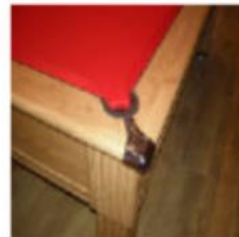
102



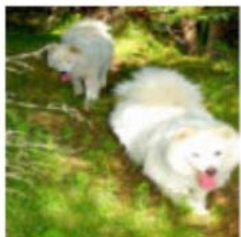
103



104



109



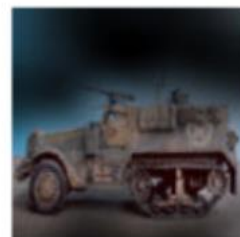
110



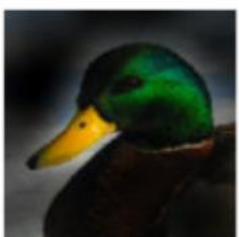
111



112



117



118



119



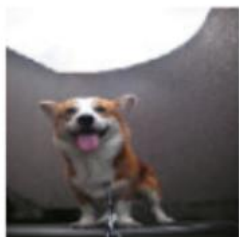
120



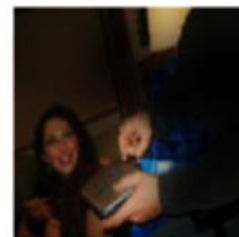
125



126

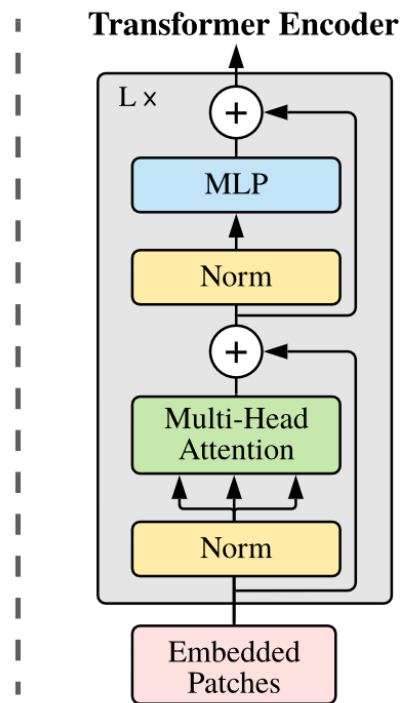
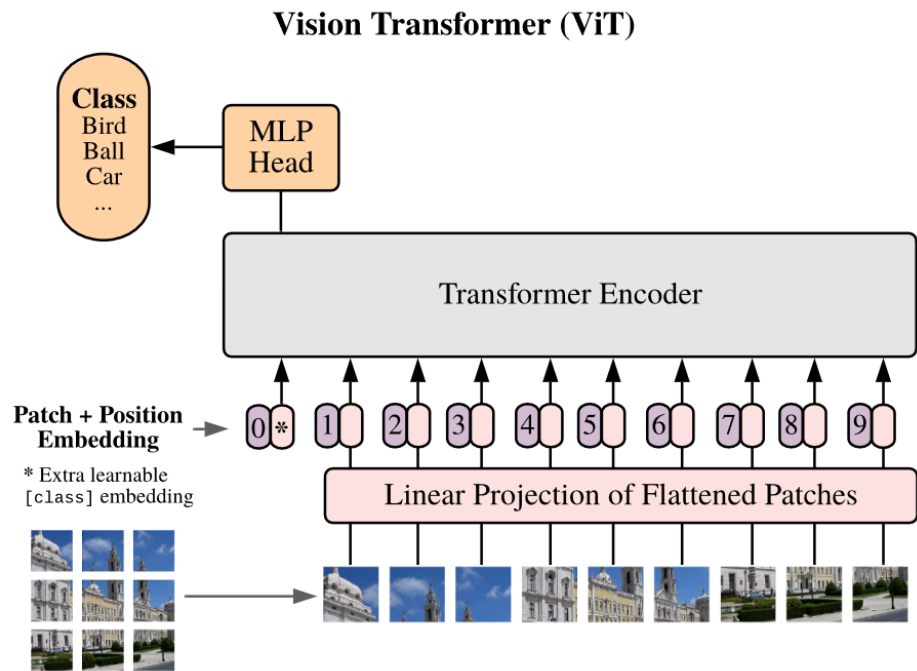


127

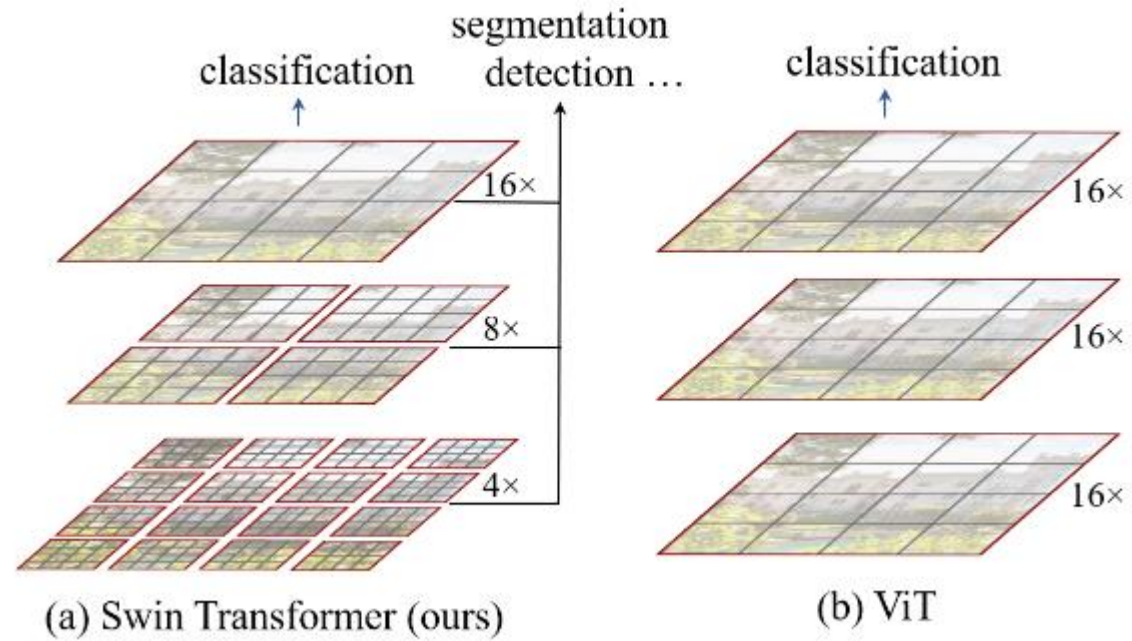


128





- This can't be ideal, right?



Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

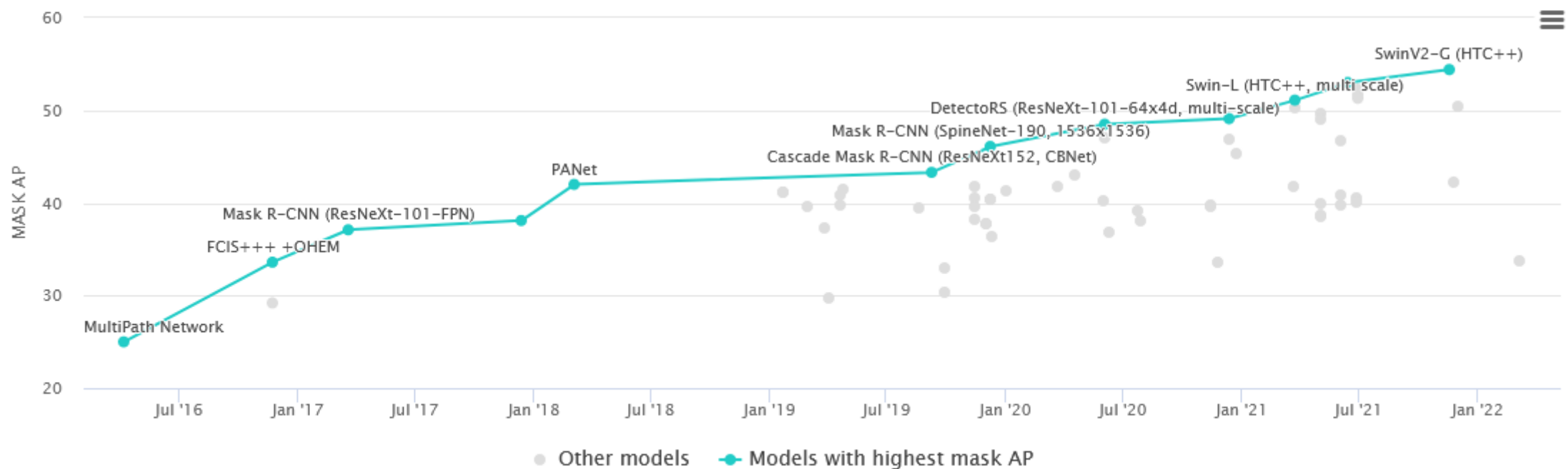
Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, Baining Guo



# Instance Segmentation on COCO test-dev

Leaderboard Dataset

View  by  for



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

# A ConvNet for the 2020s

Zhuang Liu<sup>1,2\*</sup> Hanzi Mao<sup>1</sup> Chao-Yuan Wu<sup>1</sup> Christoph Feichtenhofer<sup>1</sup> Trevor Darrell<sup>2</sup> Saining Xie<sup>1†</sup>

<sup>1</sup>Facebook AI Research (FAIR) <sup>2</sup>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.

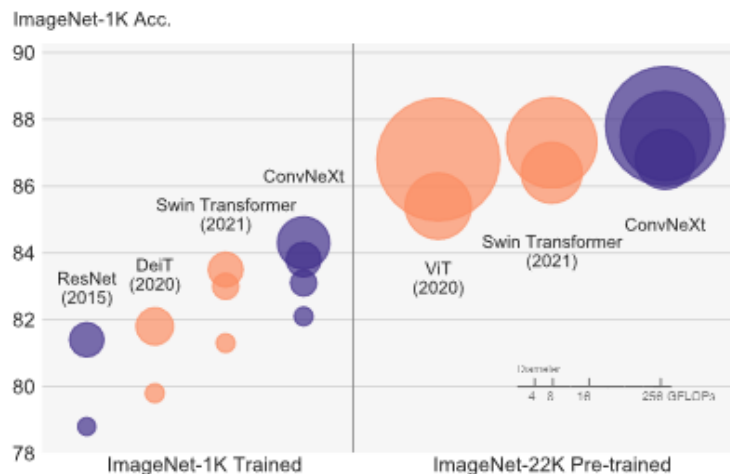


Figure 1. **ImageNet-1K classification** results for • ConvNets and ○ vision Transformers. Each bubble’s area is proportional to FLOPs of a variant in a model family. ImageNet-1K/22K models here take  $224^2/384^2$  images respectively. ResNet and ViT results were obtained with improved training procedures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

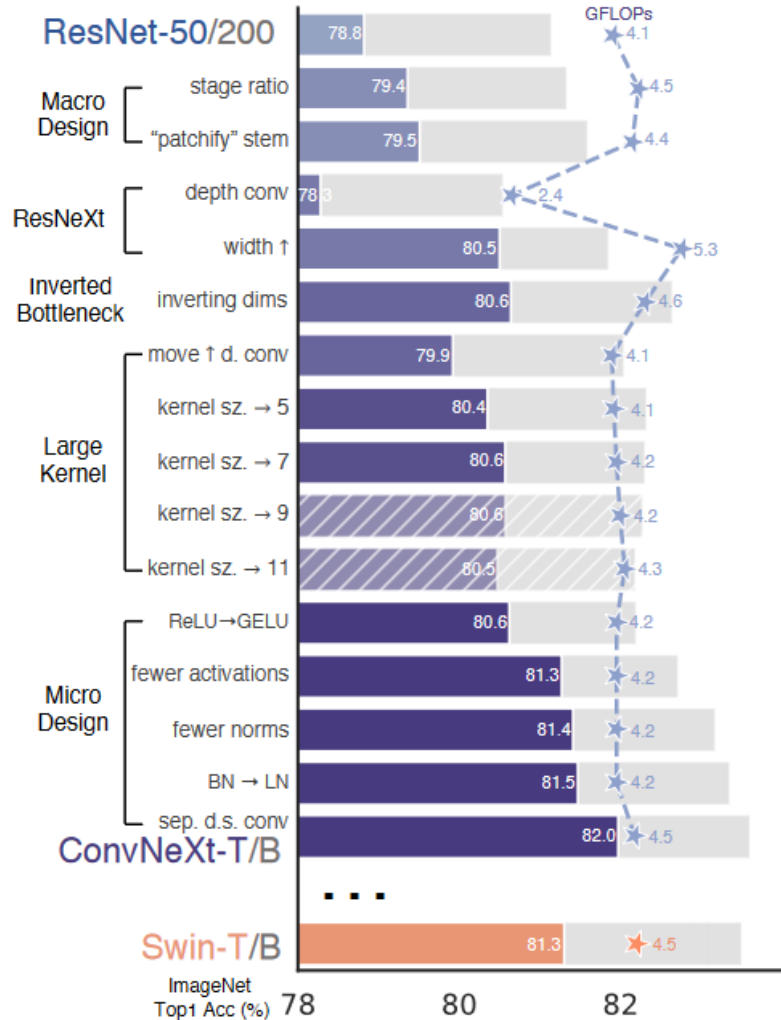


Figure 2. We modernize a standard ConvNet (ResNet) towards the design of a hierarchical vision Transformer (Swin), without introducing any attention-based modules. The foreground bars are model accuracies in the ResNet-50/Swin-T FLOP regime; results for the ResNet-200/Swin-B regime are shown with the 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 <sup>box</sup>	AP <sub>50</sub> <sup>box</sup>	AP <sub>75</sub> <sup>box</sup>	AP <sup>mask</sup>	AP <sub>50</sub> <sup>mask</sup>	AP <sub>75</sub> <sup>mask</sup>
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	<b>46.2</b>	67.9	50.8	<b>41.7</b>	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	<b>50.4</b>	69.1	54.8	<b>43.7</b>	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	<b>51.9</b>	70.8	56.5	<b>45.0</b>	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	<b>52.7</b>	71.3	57.2	<b>45.6</b>	68.9	49.5
○ Swin-B <sup>‡</sup>	982G	10.7	53.0	71.8	57.5	45.8	69.4	49.7
● ConvNeXt-B <sup>‡</sup>	964G	11.5	<b>54.0</b>	73.1	58.8	<b>46.9</b>	70.6	51.3
○ Swin-L <sup>‡</sup>	1382G	9.2	53.9	72.4	58.8	46.7	70.1	50.8
● ConvNeXt-L <sup>‡</sup>	1354G	10.0	<b>54.8</b>	73.8	59.8	<b>47.6</b>	71.3	51.7
● ConvNeXt-XL <sup>‡</sup>	1898G	8.6	<b>55.2</b>	74.2	59.9	<b>47.7</b>	71.6	52.2

Table 3. COCO object detection and segmentation results using Mask-RCNN and Cascade Mask-RCNN. <sup>‡</sup> 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).

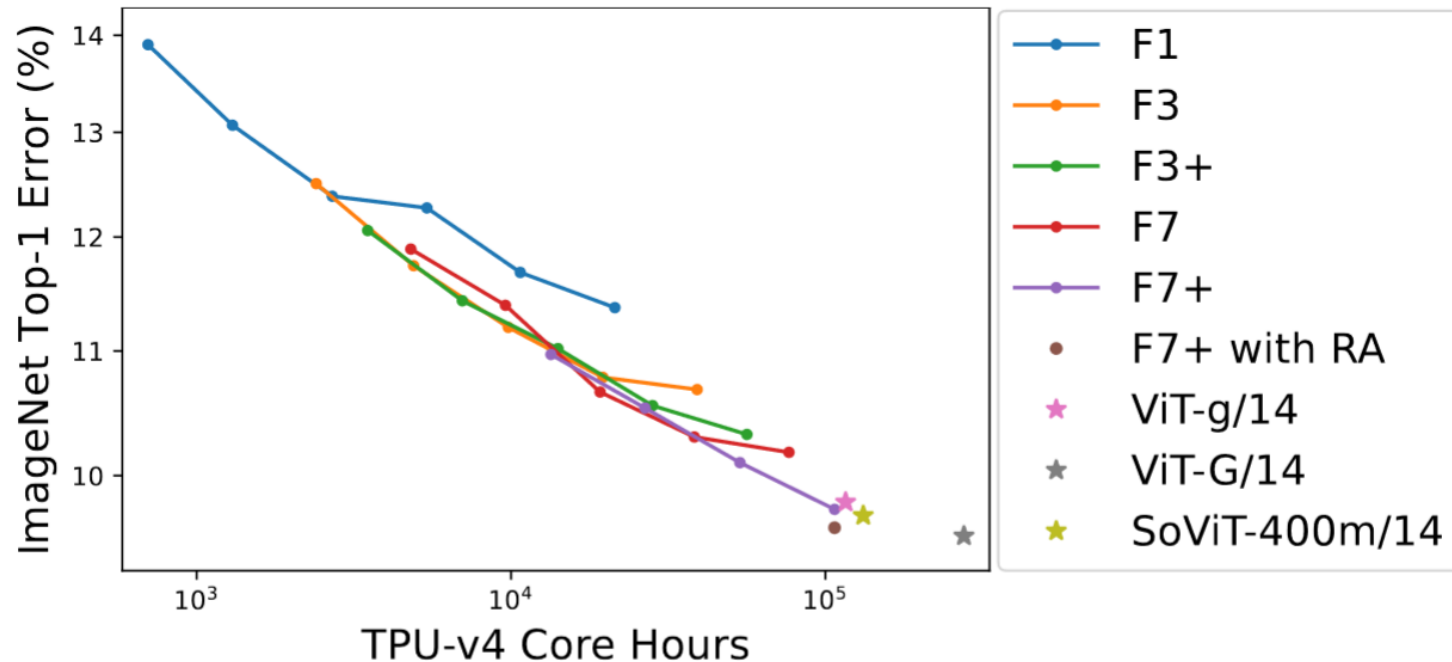
# ConvNets Match Vision Transformers at Scale

Samuel L Smith<sup>1</sup>, Andrew Brock<sup>1</sup>, Leonard Berrada<sup>1</sup> and Soham De<sup>1</sup>

<sup>1</sup>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



# 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.