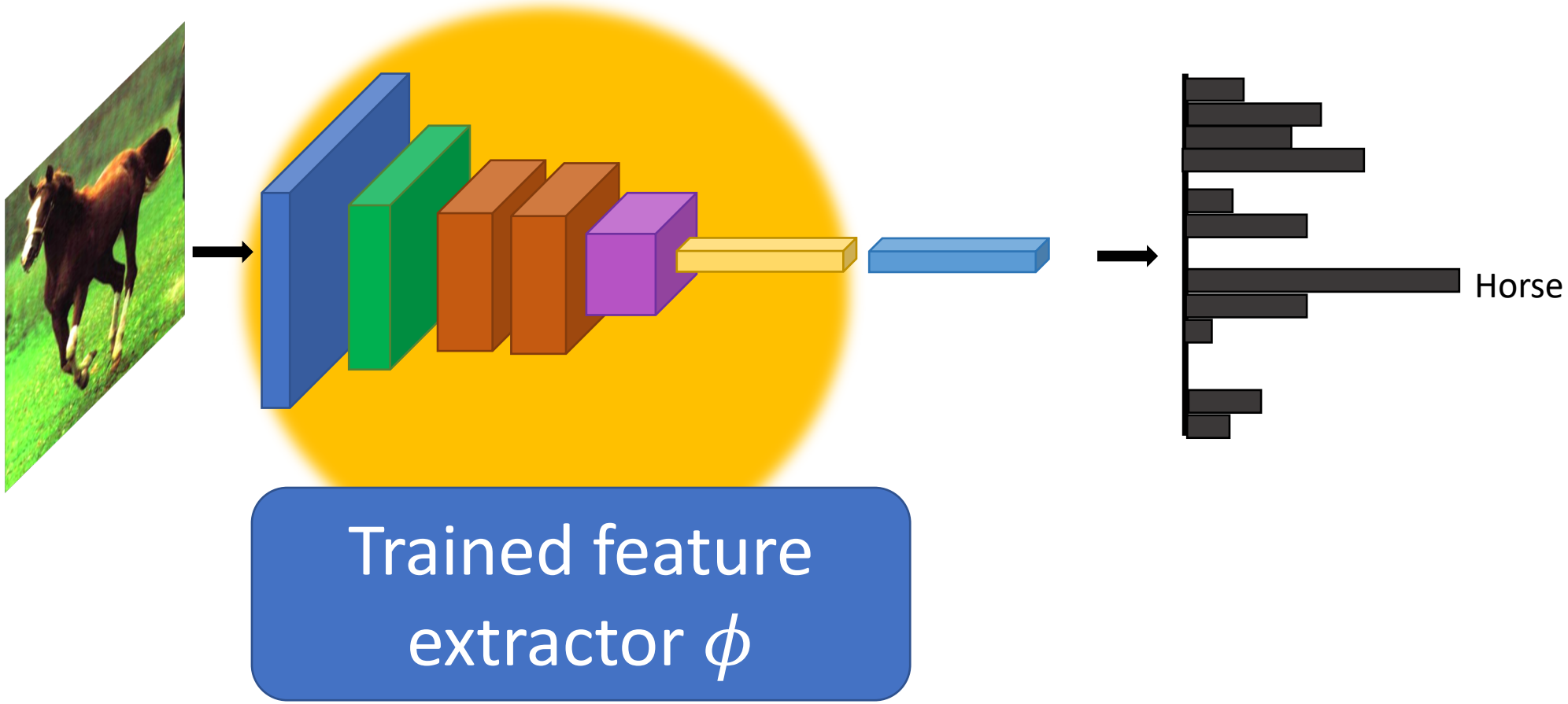
Transfer Learning

Transfer learning with convolutional networks



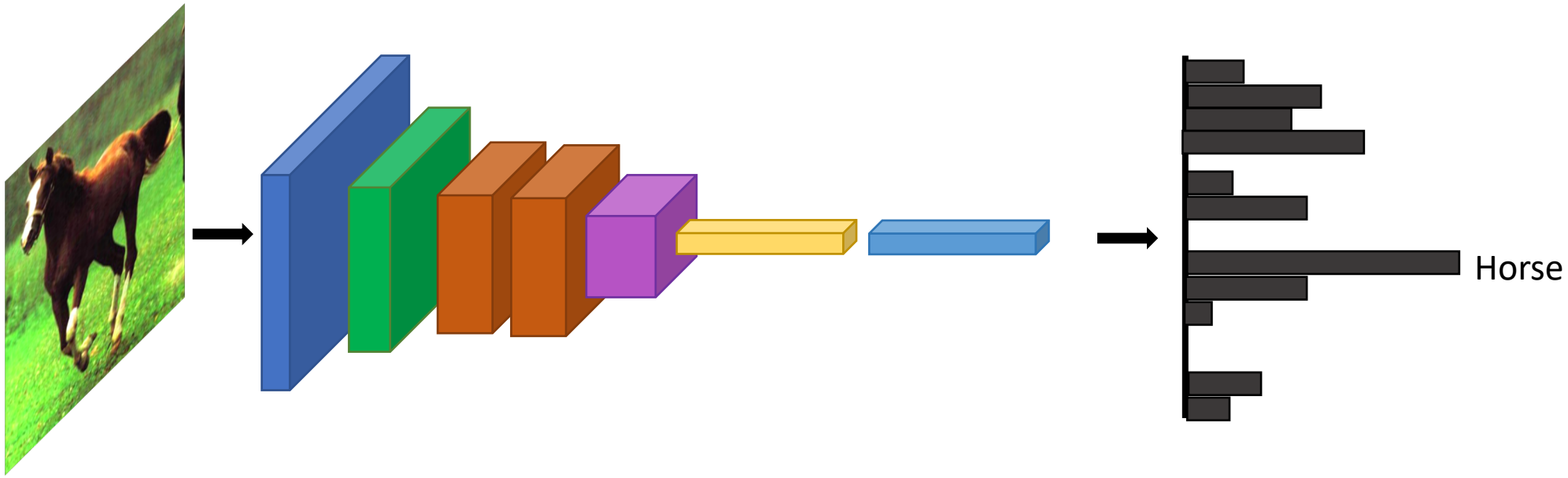
Transfer learning with convolutional networks

Dataset	Non-Convnet Method	Non-Convnet perf	Pretrained convnet + classifier	Improvement
Caltech 101	MKL	84.3	87.7	+3.4
VOC 2007	SIFT+FK	61.7	79.7	+18
CUB 200	SIFT+FK	18.8	61.0	+42.2
Aircraft	SIFT+FK	61.0	45.0	-16
Cars	SIFT+FK	59.2	36.5	-22.7

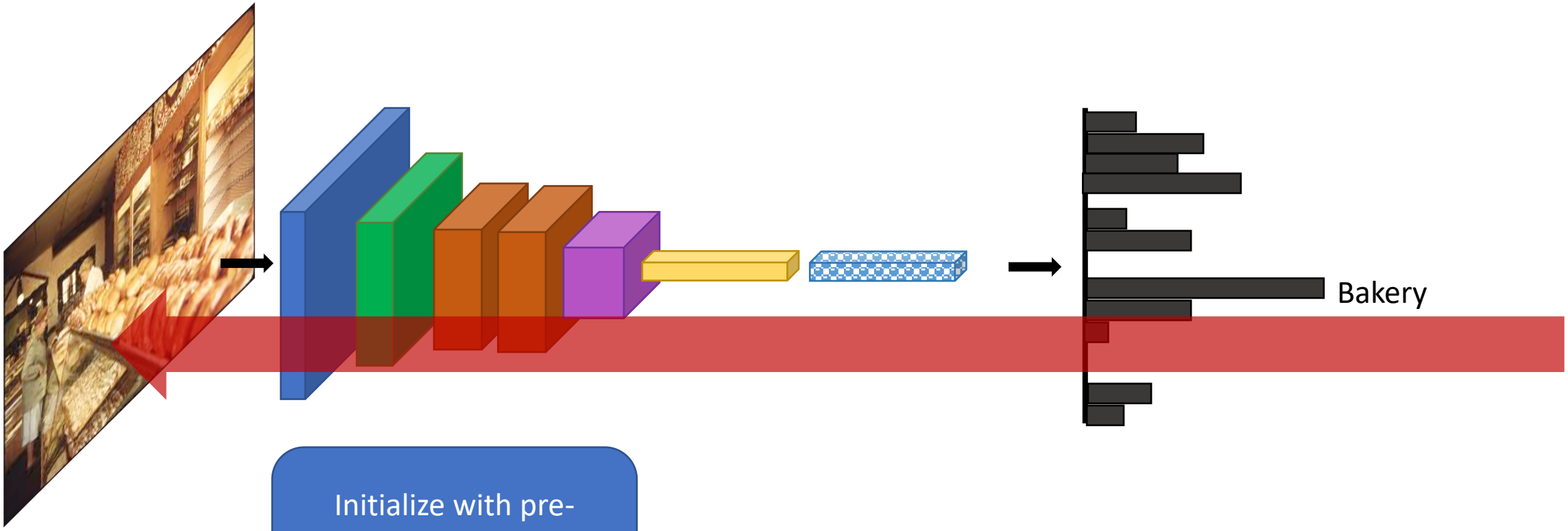
Why transfer learning?

- Availability of training data
- Computational cost
- Ability to pre-compute feature vectors and use for multiple tasks
- *Con: NO end-to-end learning*

Finetuning



Finetuning



Initialize with pre-trained, then train with low learning rate

Finetuning

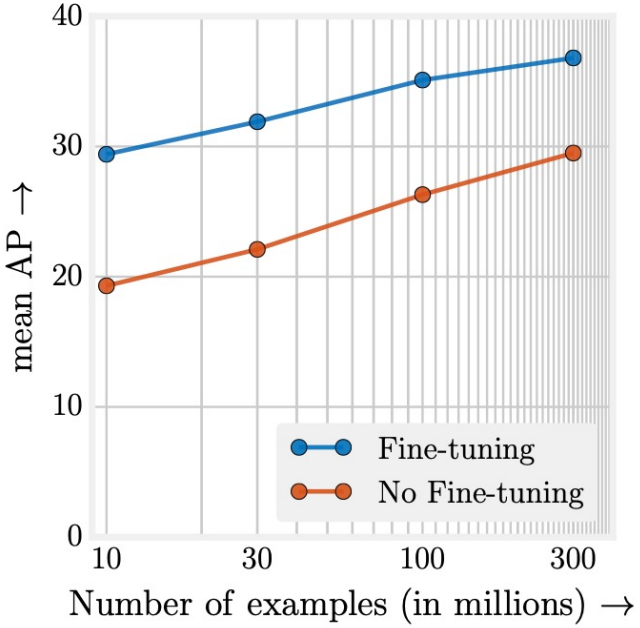
Dataset	Non-Convnet Method	Non-Convnet perf	Pretrained convnet + classifier	Finetuned convnet	Improvement
Caltech 101	MKL	84.3	87.7	88.4	+4.1
VOC 2007	SIFT+FK	61.7	79.7	82.4	+20.7
CUB 200	SIFT+FK	18.8	61.0	70.4	+51.6
Aircraft	SIFT+FK	61.0	45.0	74.1	+13.1
Cars	SIFT+FK	59.2	36.5	79.8	+20.6

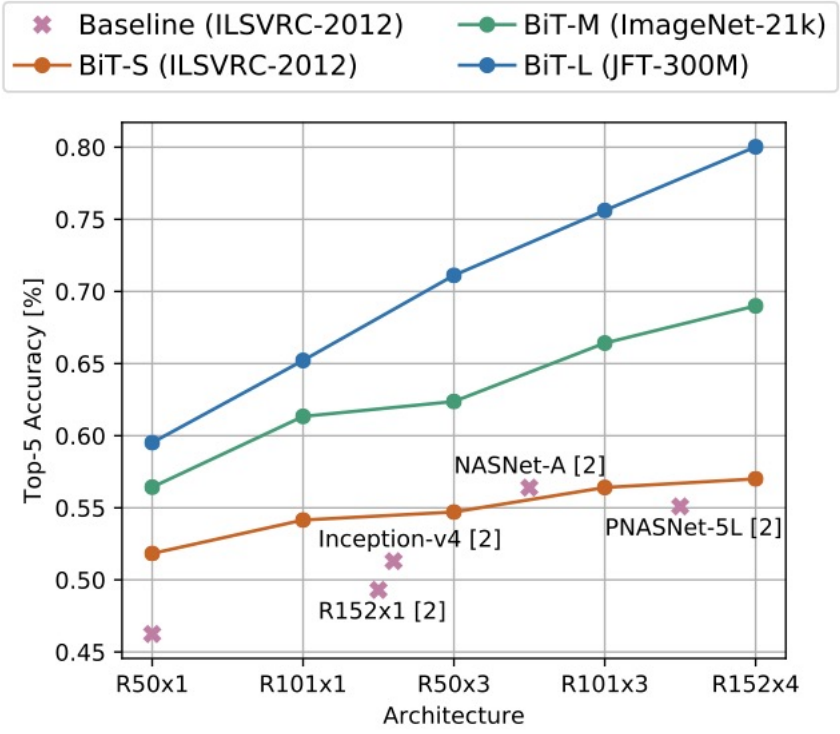
What impacts transfer accuracies?

- Relationship between pre-training and target task?
- Unclear: sometimes transfer works across very different domains
  - E.g., ImageNet -> Satellite images
- Very limited work on understanding this

What impacts transfer accuracy?

- Size of the pre-training dataset
- Size of the model
- Bigger is better





1. Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., & Houlsby, N. (2020). Big transfer (bit): General visual representation learning. In *Computer Vision–ECCV 2020: 16th European Conferenc Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16* (pp. 491-507). Springer International Publishing.
2. Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision* (pp. 843-852).

Concerns about big transfer

- Opaque datasets?
- Uncurated datasets?
- Bias in the datasets?
  - Do biased datasets affect transfer? Turns out yes.

Computational complexity

Analyzing computational complexity

- What is the computational complexity of a single convolutional layer?
  - $h \times w \times c$ input and output
  - $k \times k$ kernel
- Space:
  - Input/output: hwc
  - Filters: k^2c^2
- Time (Flops): hwk^2c^2

Reducing computational complexity

- …while maintaining accuracy?
- Multiple ways:
  - Make architecture *a priori* cheaper
  - Make *weights* and *operations* cheaper
  - Make inference adaptive

Cheaper convolutional blocks

- Standard convolution:
  - Each filter operates on all channels
  - Single $k \times k$ filter operating on c channels producing one output channel: k^2c parameters, cost
  - c such filters: k^2c^2 parameters, cost
- *Depthwise separable* convolution
  - Each filter operates on a single channel
  - c filters operating on c channels: k^2c parameters, cost
  - But each channel is independently processed
  - Add a 1x1 convolution at the end with cost c^2 : $k^2c + c^2$ parameters

Cheaper convolutional blocks

- Depthwise separable convolutions are specific instance of more general idea: *grouped convolutions*
- Grouped convolutions in original AlexNet network
- Grouped convolution:
  - Divide input channels into g groups
  - Apply convolutional layers on each group independently
  - Concatenate

Grouped and depth-wise convolutions

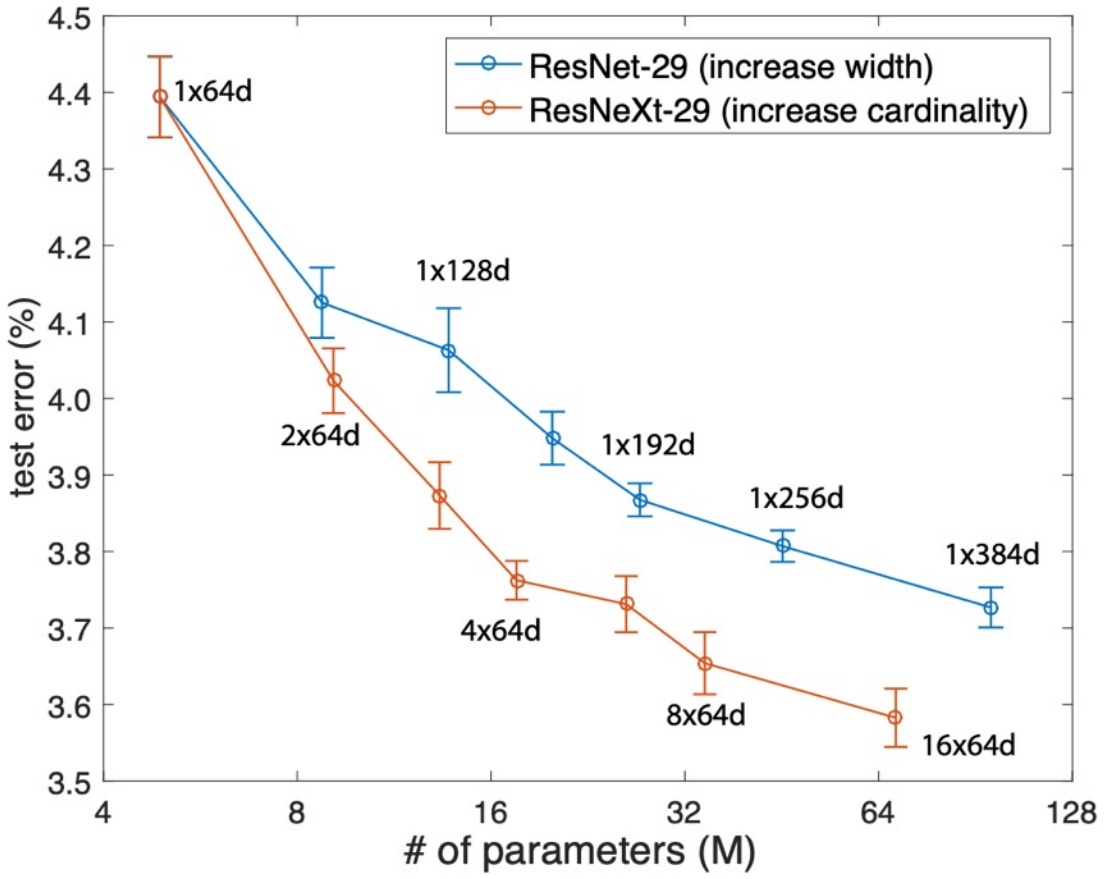


Table 4. Depthwise Separable vs Full Convolution MobileNet

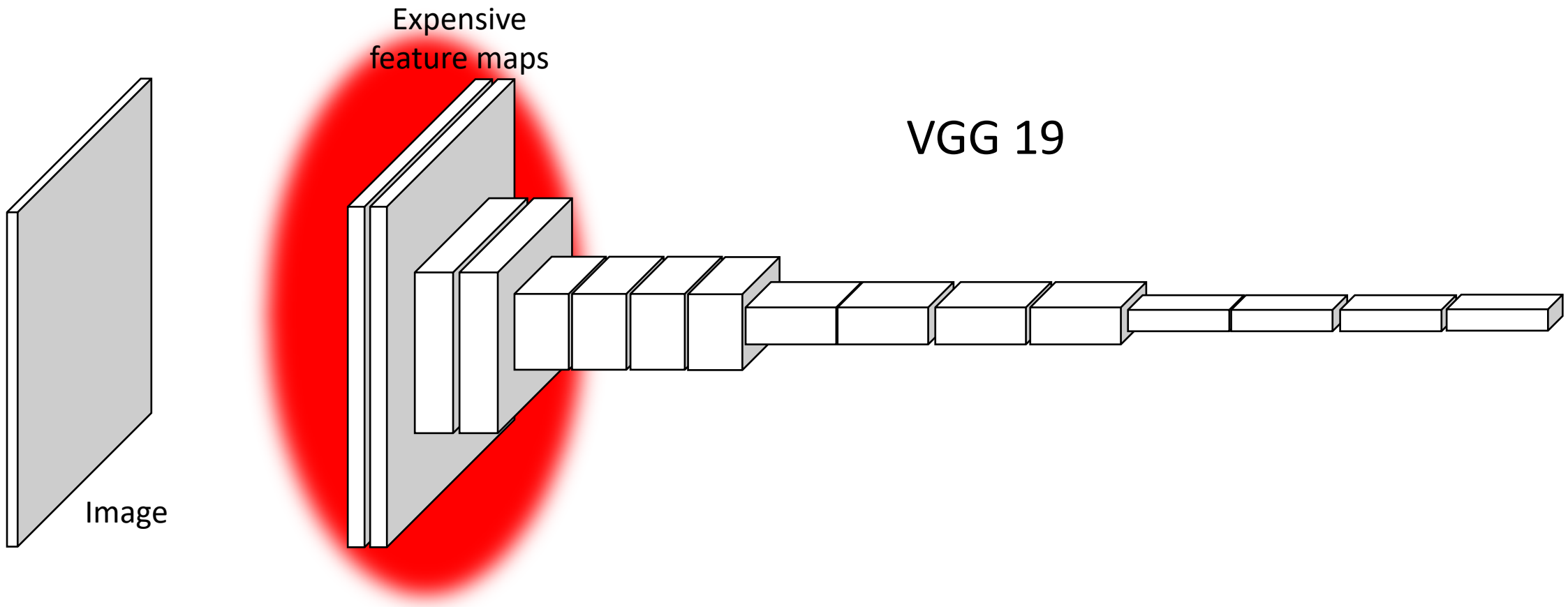
Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
Conv MobileNet	71.7%	4866	29.3
MobileNet	70.6%	569	4.2

Xie, Saining, et al. "Aggregated residual transformations for deep neural networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).

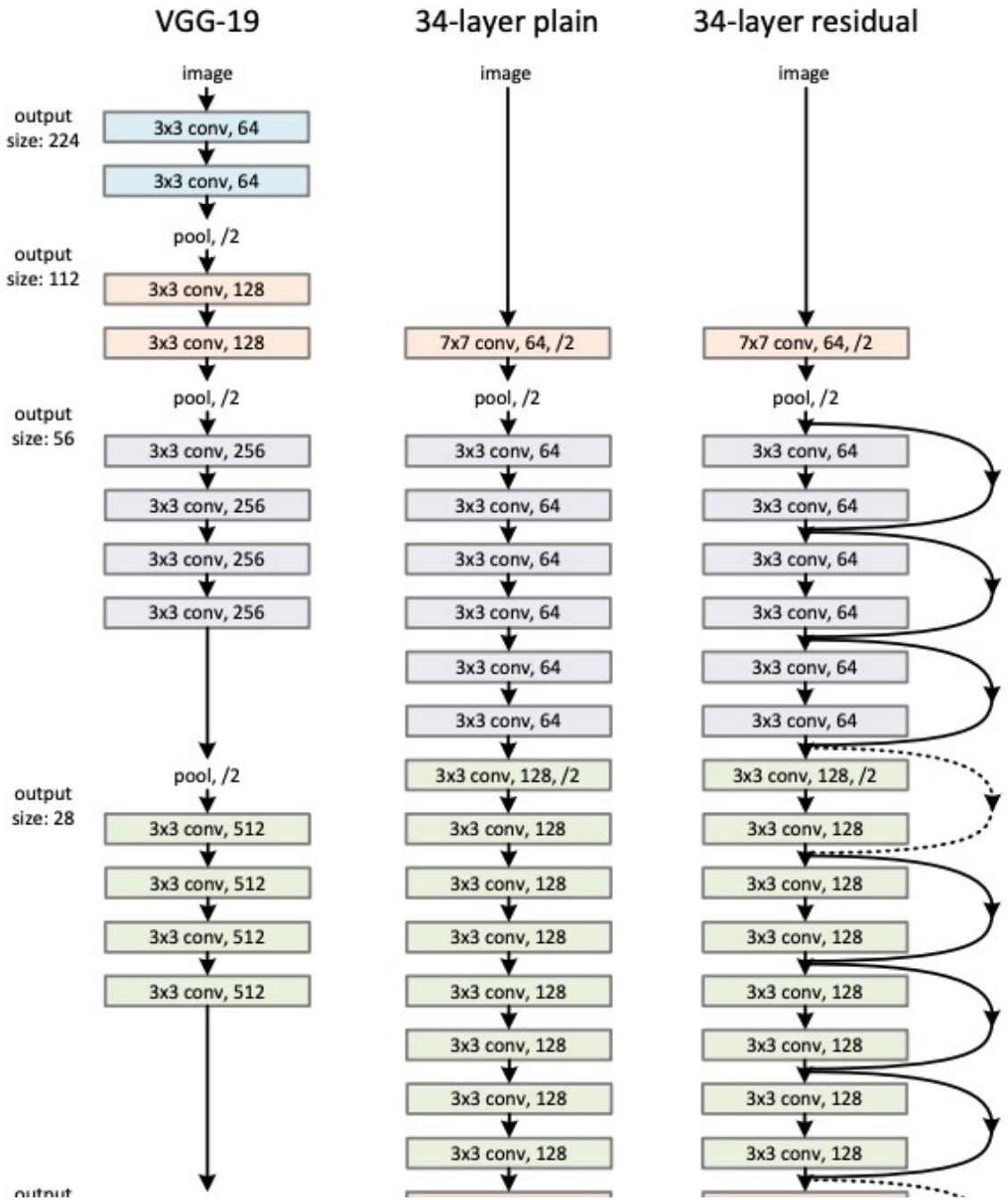
Other architectural changes

- Biggest memory consumption: large feature maps



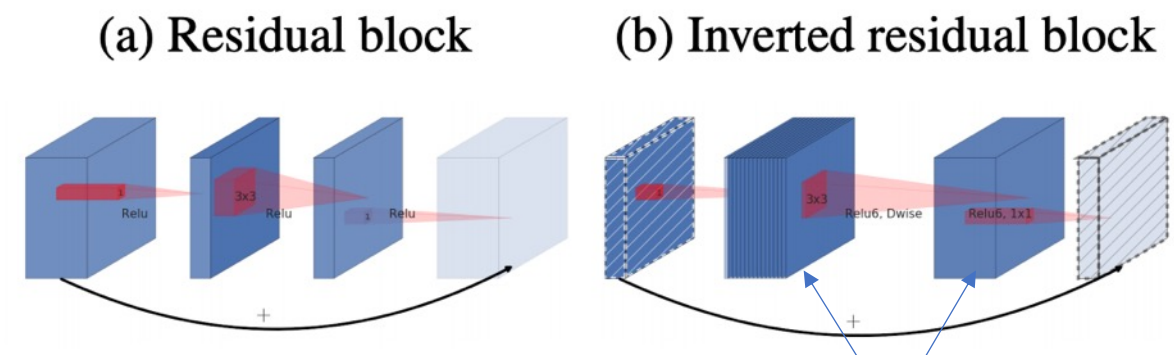
Other architectural changes

- Biggest memory consumption: large feature maps
- Simple solution: reduce resolution early



Other architectural changes

- Biggest memory consumption: large feature maps
- Simple solution (ResNet):
  - Reduce resolution drastically (/4) early
- More sophisticated changes: Inverted residuals (MobileNet v2)



Dispose of these

Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

Other kinds of connections

- DenseNets
  - Replace addition of residuals with concatenation
  - Alternative to solving vanishing gradient problem
  - Should *increase* number of parameters, but *decreases* them
  - Better re-use of features

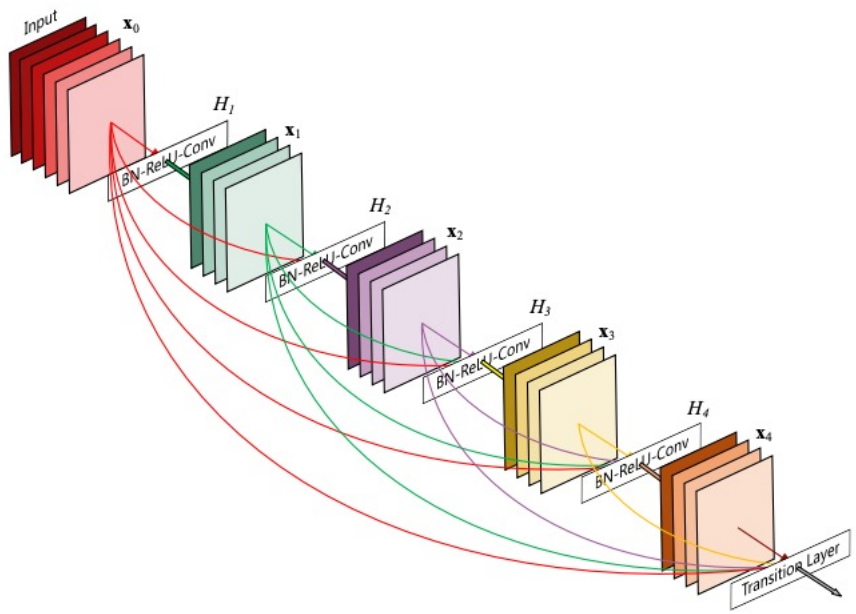
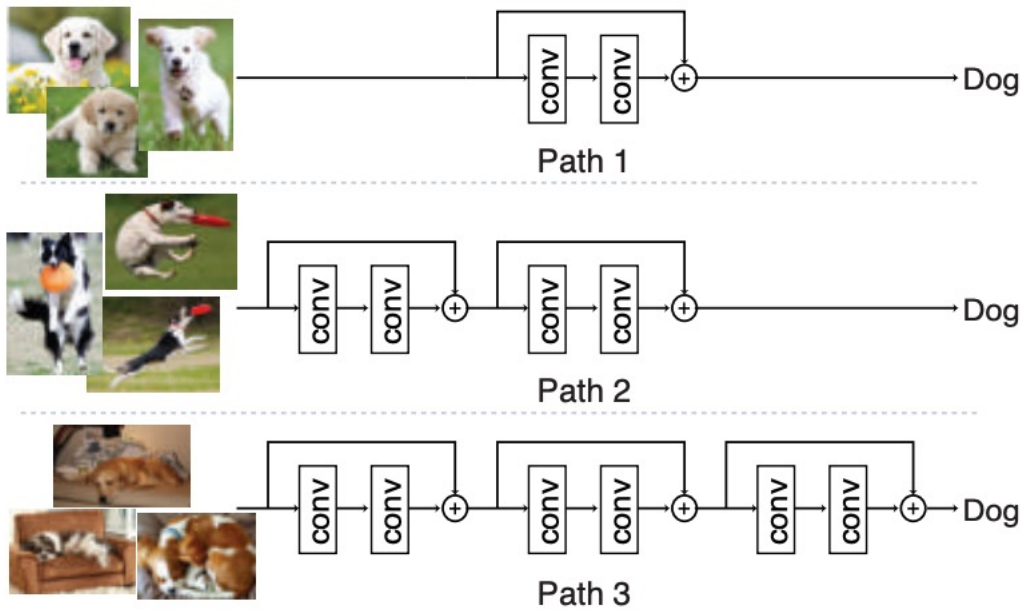


Figure 1: A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input.

Adaptive inference

- Some examples are harder than others
- Should be able to use different amounts of computation for different examples
- Version 1: skip some residuals

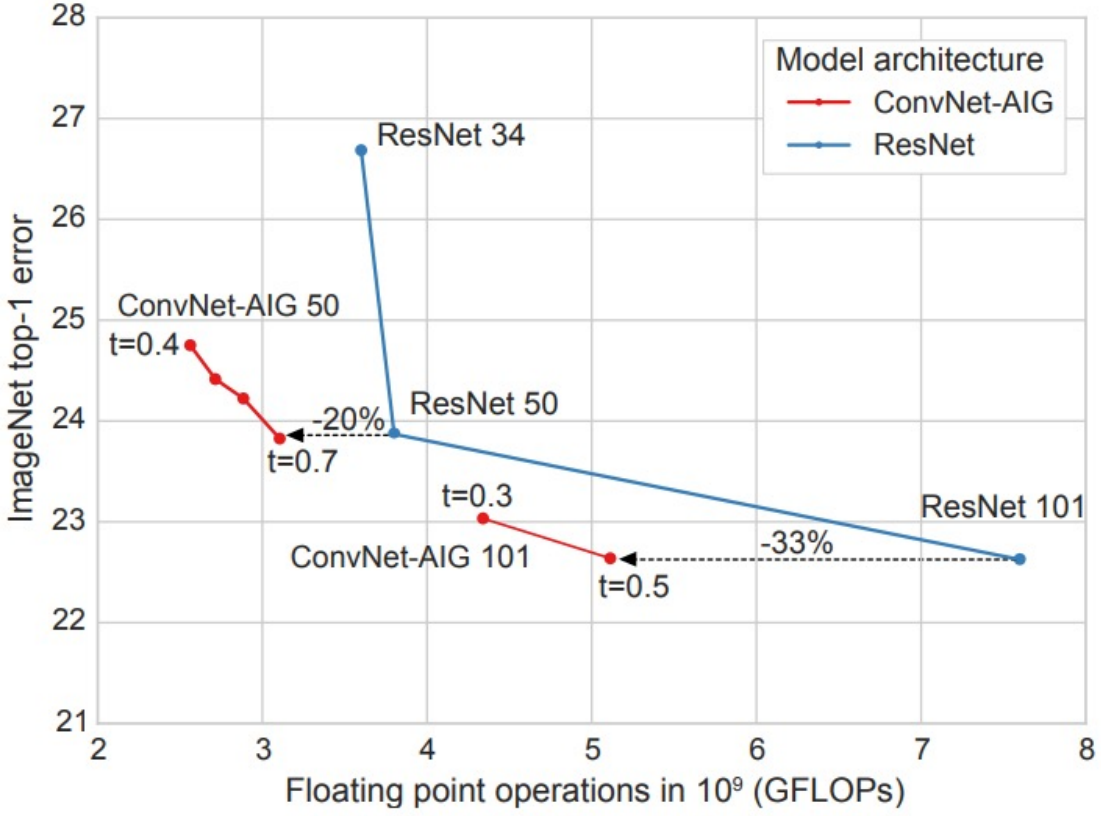


Veit, Andreas, and Serge Belongie. "Convolutional networks with adaptive inference graphs." *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

Wu, Zuxuan, et al. "Blockdrop: Dynamic inference paths in residual networks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

Adaptive inference

- Some examples are harder than others
- Should be able to use different amounts of computation for different examples
- Version 1: skip some residuals

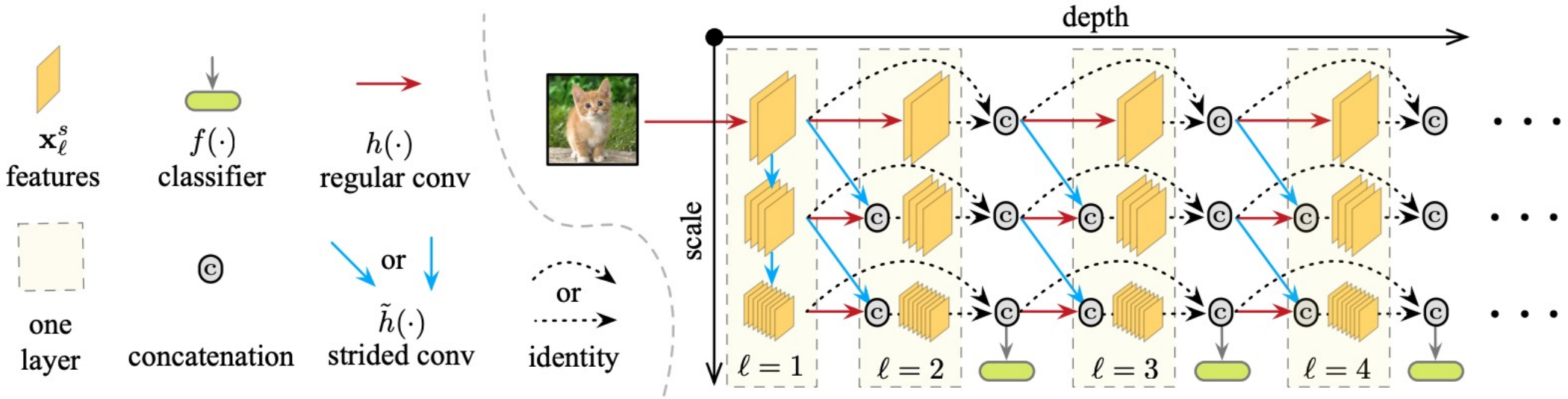


Veit, Andreas, and Serge Belongie. "Convolutional networks with adaptive inference graphs." *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

Wu, Zuxuan, et al. "Blockdrop: Dynamic inference paths in residual networks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

Adaptive inference

- Some examples are harder than others
- Should be able to use different amounts of computation for different examples
- Version 2: reduce resolution at different rates



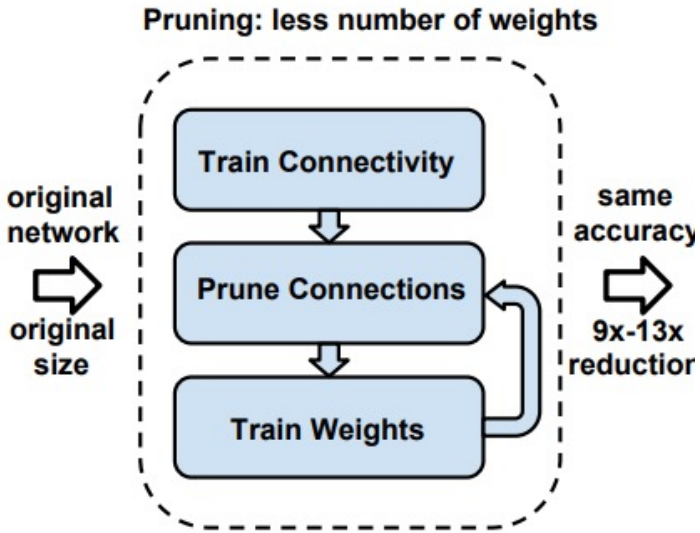
Huang, Gao, et al. "Multi-scale dense networks for resource efficient image classification." *ICLR 2018*

Compressing model weights

- All of model storage: filters
- Flops also scale with non-zero entries in filters (in principle)
- Compress filters
  - Sparsify them
  - Represent them with fewer bits

Pruning network connections

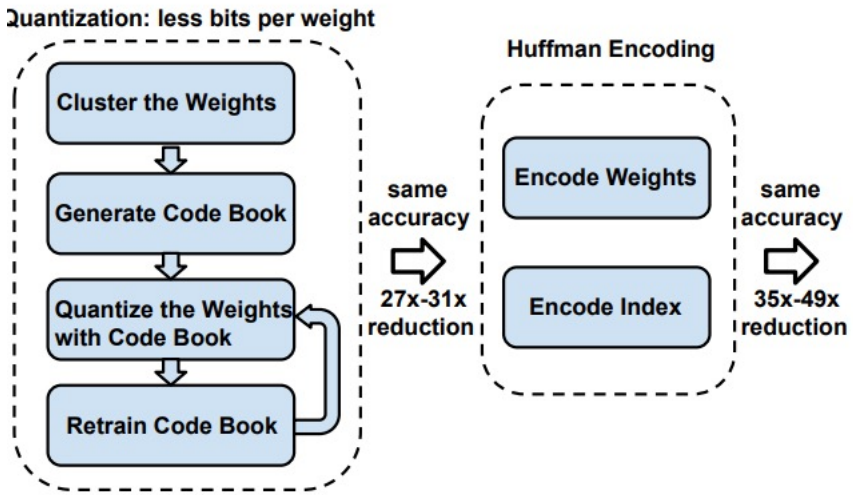
- Simple approach: prune weights that are below a threshold
- Retrain rest of the weights
- Repeat



Han, Song, Huizi Mao, and William J. Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." *ICLR, 2016*

Filter quantization

- Two questions:
  - How do we quantize?
  - Quantization → discrete values. How do we optimize?
- Example 1: *cluster*
  - Weights → indices into dictionary
  - Update dictionary elements as parameters.



Han, Song, Huizi Mao, and William J. Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." *ICLR, 2016*

Filter quantization

- Two questions:
  - How do we quantize?
  - Quantization → discrete values. How do we optimize?
- Example 2: *binarize/ternarize*
  - Weights → binary/ternary, + real-valued scale
  - Parameter updates happen in real space

Zhu, Chenzhuo, et al. "Trained ternary quantization." *ICLR, 2017*.

Rastegari, Mohammad, et al. "Xnor-net: Imagenet classification using binary convolutional neural networks." *European Conference on Computer Vision*. Springer, Cham, 2016.

Beyond ConvNets

Attention (Transformers)

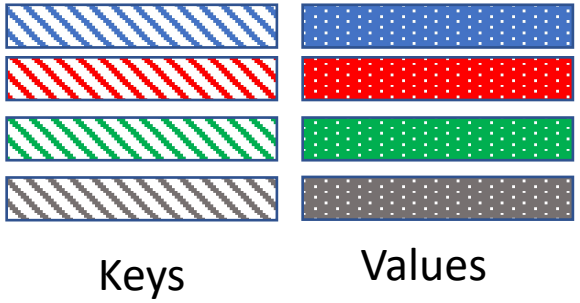
- Comes from the NLP community
- Originally an approach for language generation
- Motivation: need to attend to different parts of the sentence for, e.g., translation

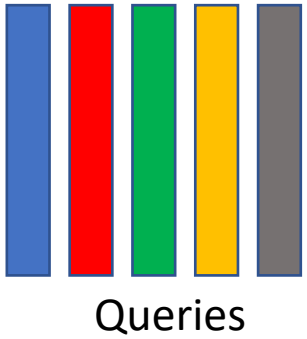
He went to the river

वह नदी के पास गया

Attention (Transformers)

- Comes from the NLP community
- Is an approach for processing sets

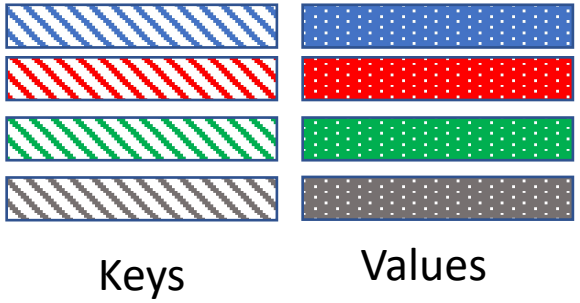


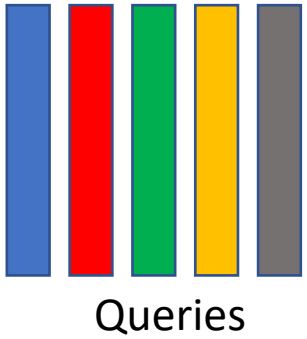


Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*(pp. 5998-6008).

Attention (Transformers)

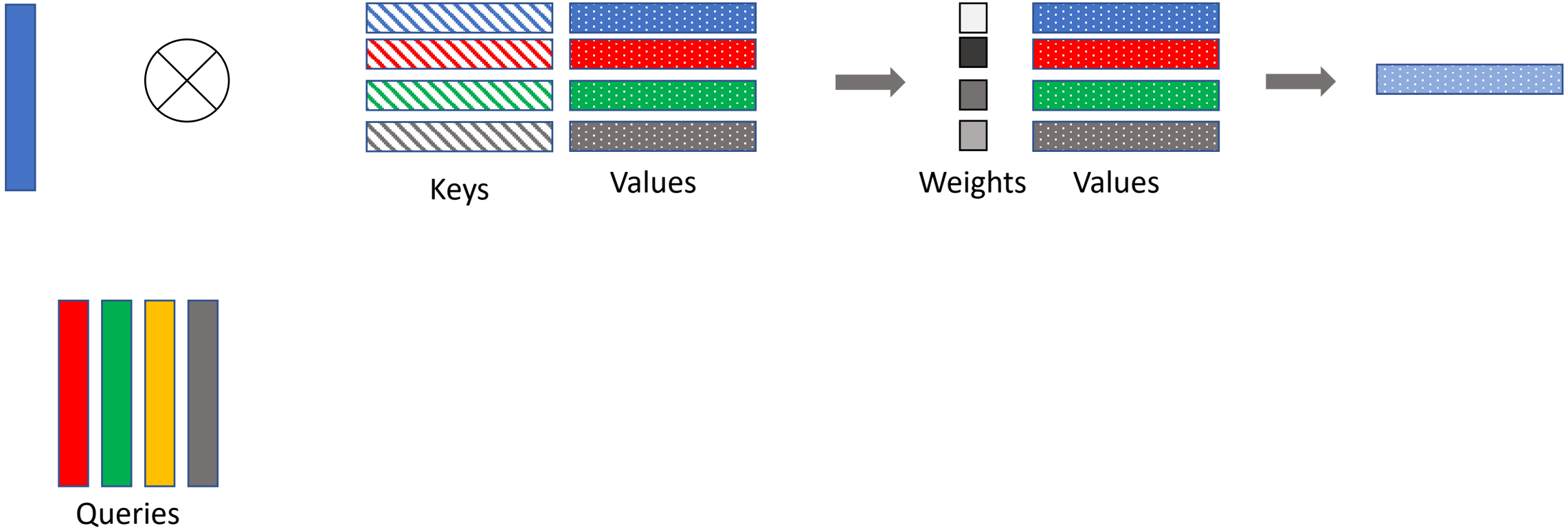
- Comes from the NLP community
- Is an approach for processing sets



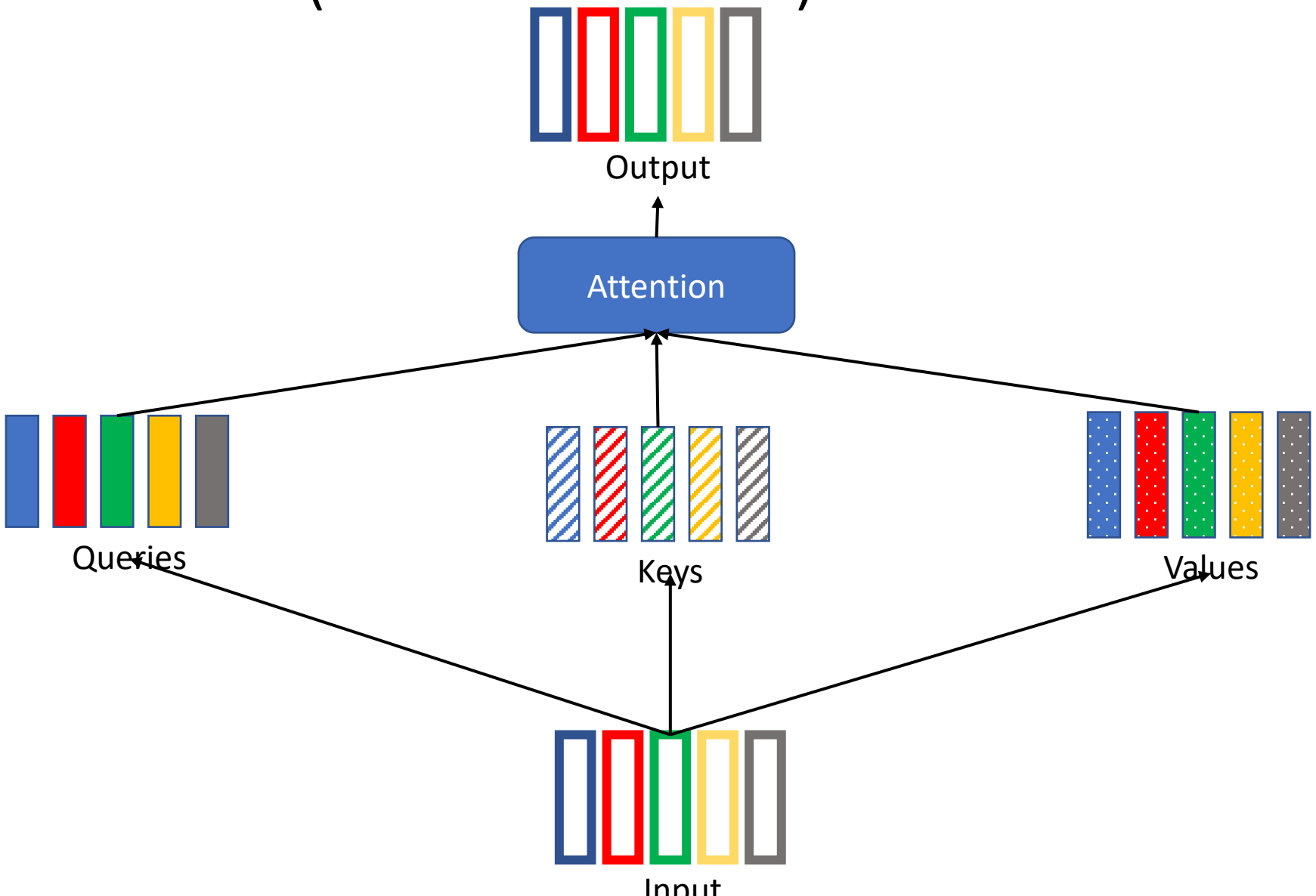


Attention (Transformers)

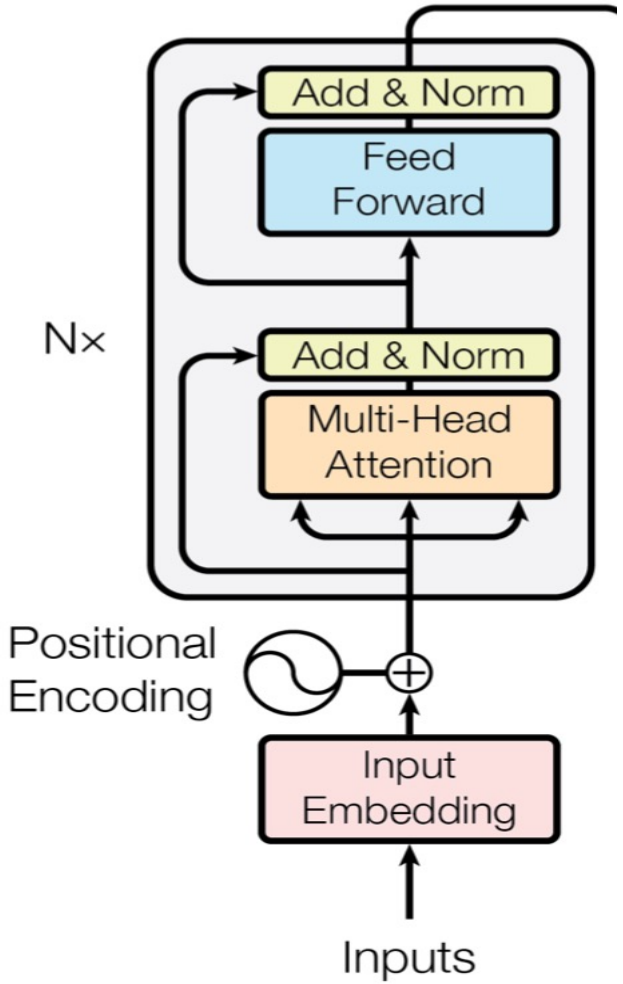
- Comes from the NLP community
- Is an approach for processing sets



Attention (Transformers)

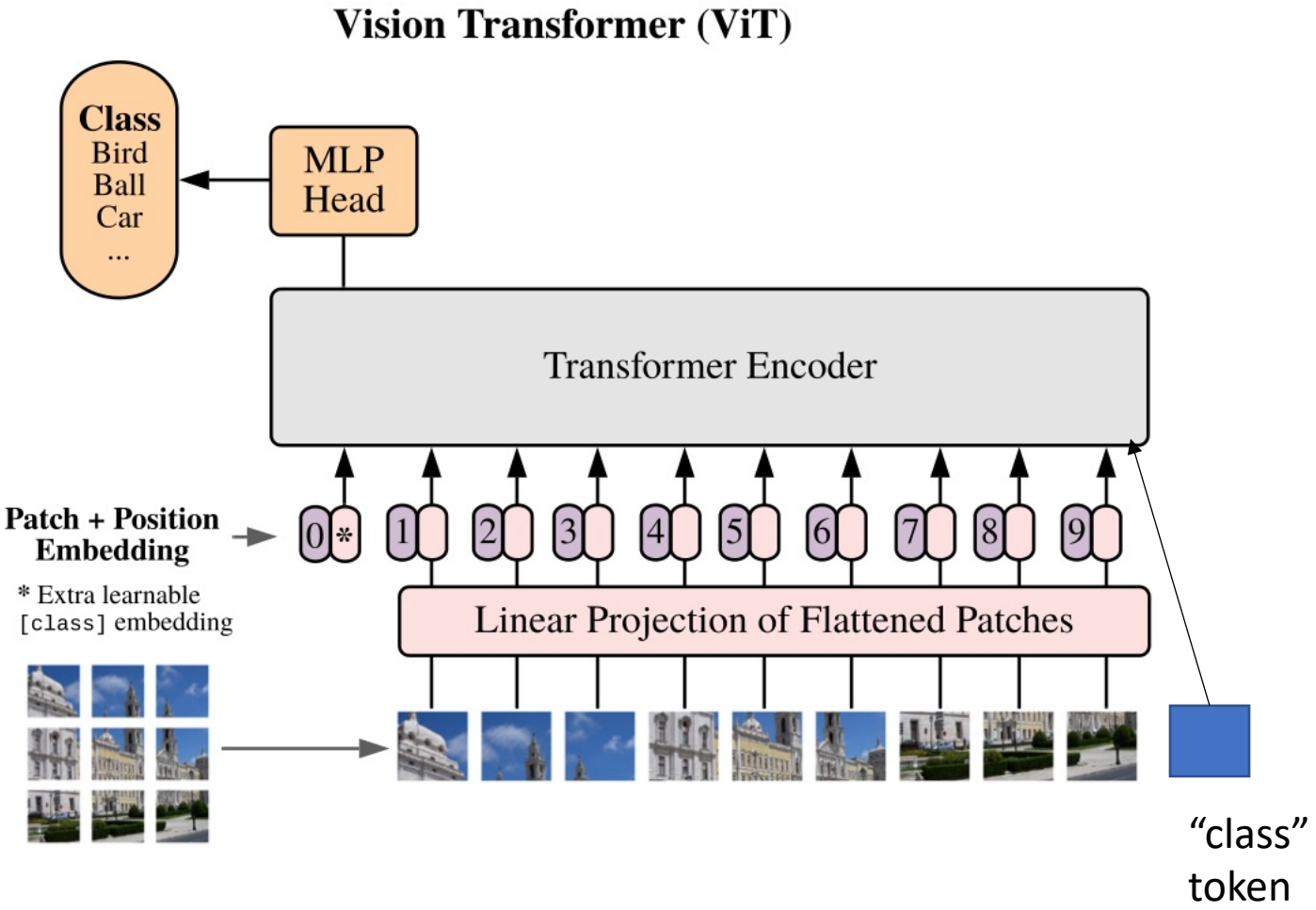


Attention (Transformers)



Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems*. 2017.

Attention (Transformers) in computer vision



1. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
2. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021, July). Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning* (pp. 10347-10357). PMLR.

Transformers in computer vision

- An evolving frontier
  - First key results only late last year
- Preliminary conclusions
  - Transformers can outperform convnets, but require much stronger data augmentation
  - Transformers seem to transfer better.