Transformers II

CS 4391 Introduction Computer Vision

Professor Yu Xiang

The University of Texas at Dallas

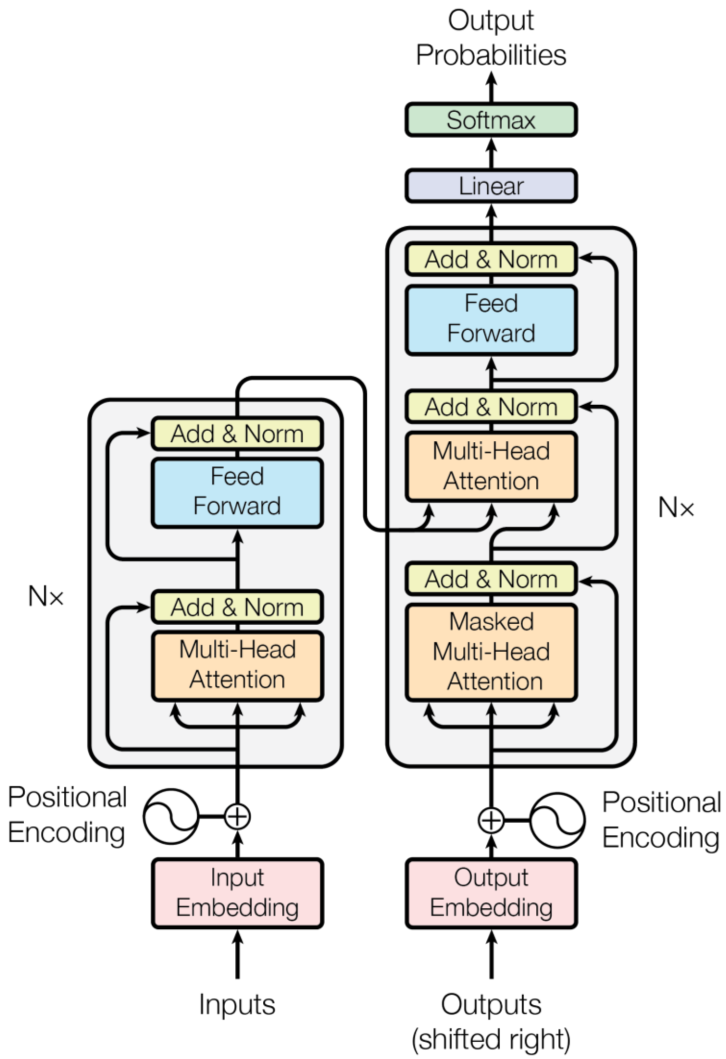
Transformer

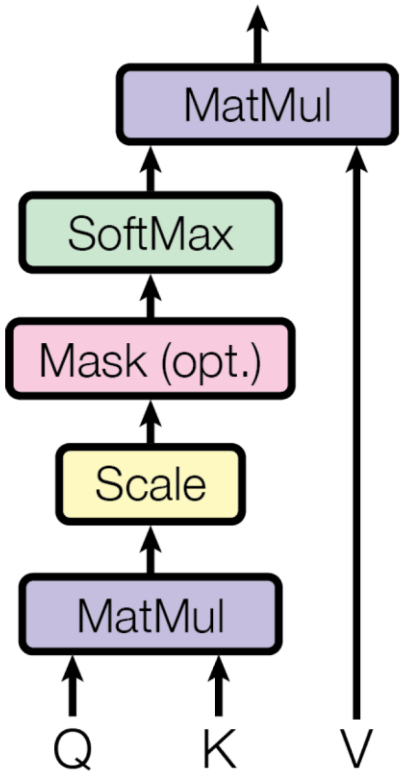
- No recurrence

- Attention only
  - Global dependencies between input and output

  - More parallelization compared to RNNs

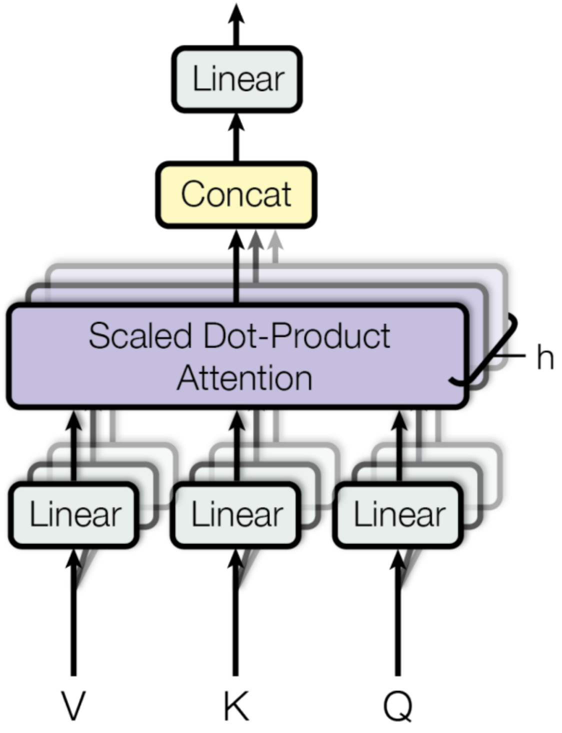
Transformer





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

Multi-Head Attention



$$\mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

$$\mathrm{MultiHead}(Q,K,V)$$

$$= \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$

Attention is all you need. Vaswani et al., NeurIPS’17

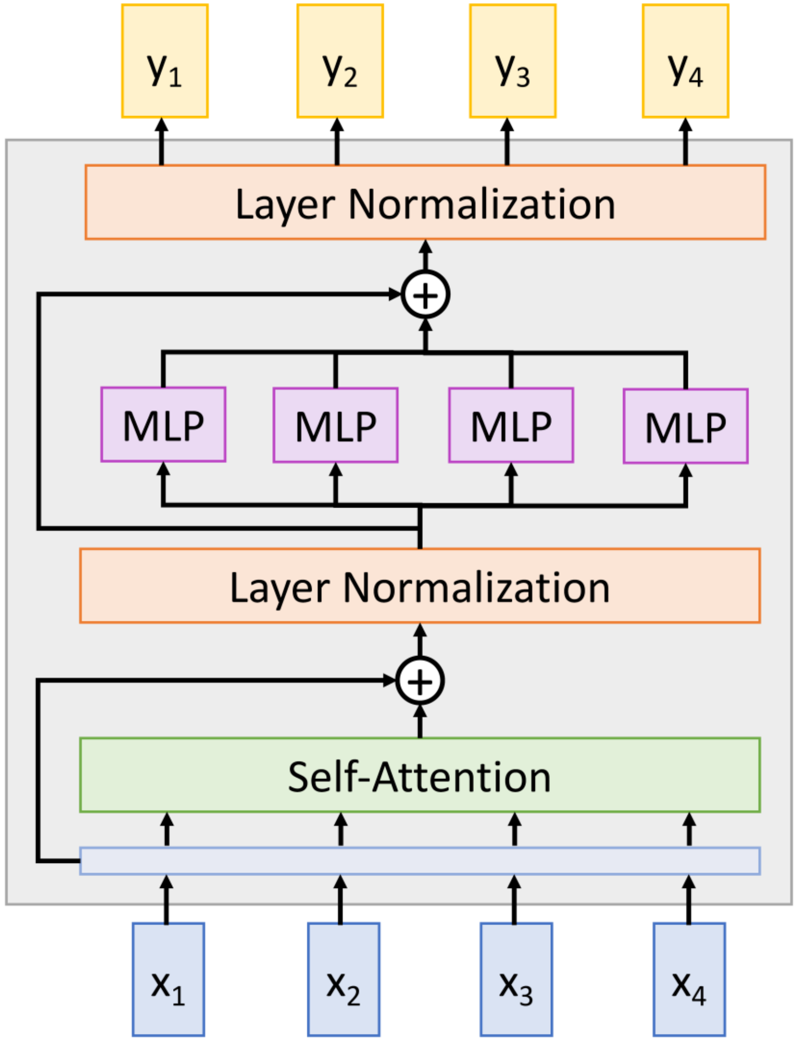
Transformer

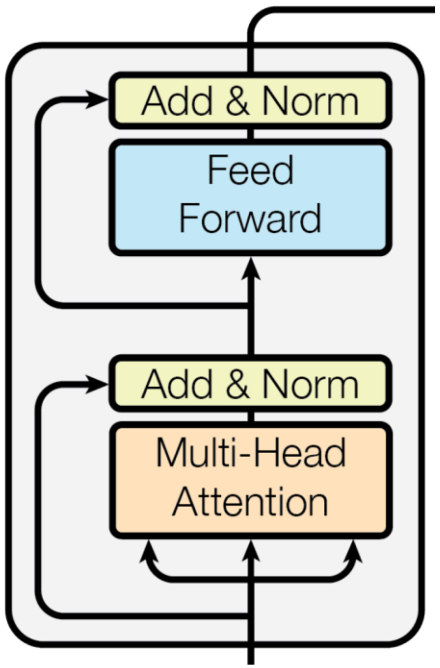
- Transformer block
  - Input: a set of vectors

$$n \times d_{\text{model}}$$

  - Output: a set of vectors

$$n \times d_{\text{model}}$$



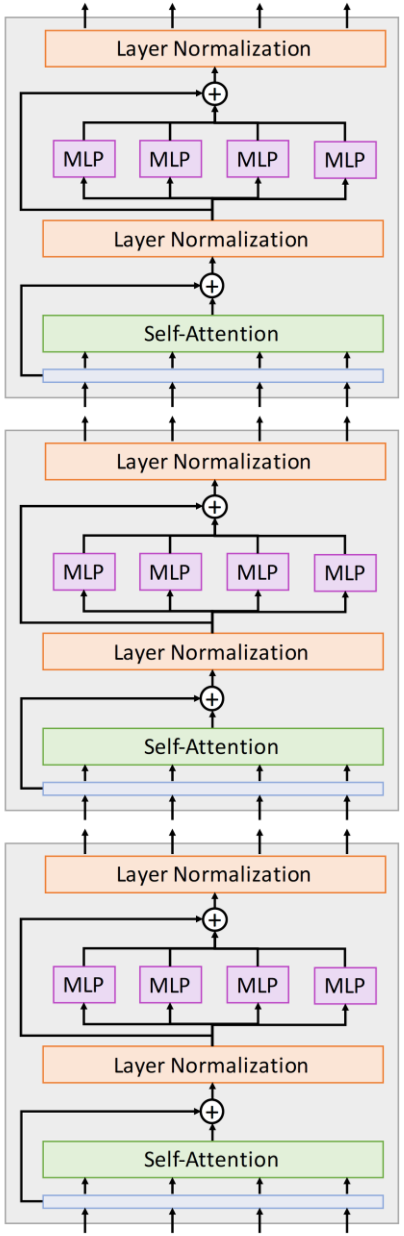


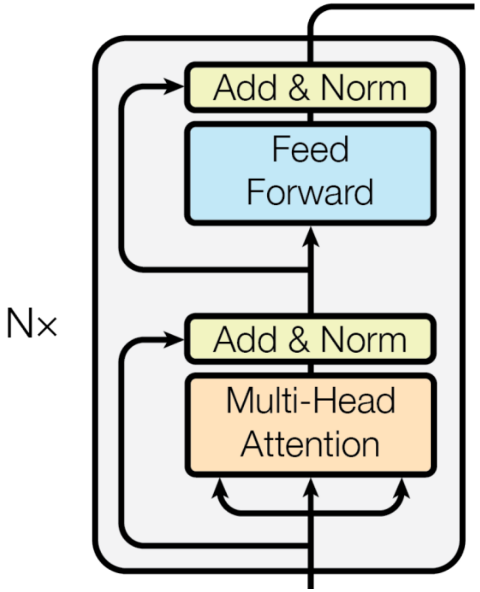
Transformer

- Hyper-parameters
  - Number of blocks
  - Number of heads per block

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_{\mathrm{h}})W^O$$

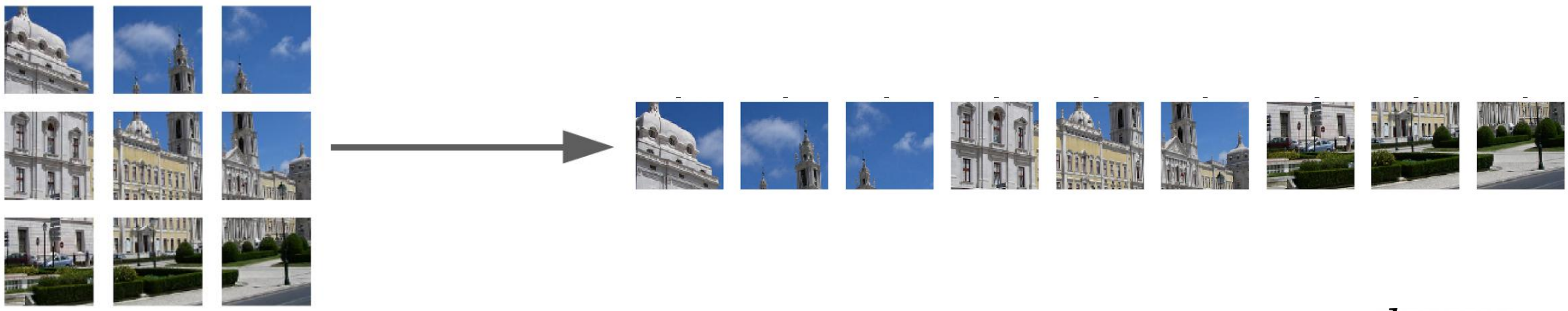
  - Width (channels per head, FFN width)





Vision Transformer (ViT)

- Convert an image into a sequence of “token”



- Input embedding by linear projection

d_{model}

$$\mathbf{x}_p^1\mathbf{E};\ \mathbf{x}_p^2\mathbf{E};\cdots;\ \mathbf{x}_p^N\mathbf{E} \qquad \mathbf{E} \in \mathbb{R}^{(P^2\cdot C)\times D}$$

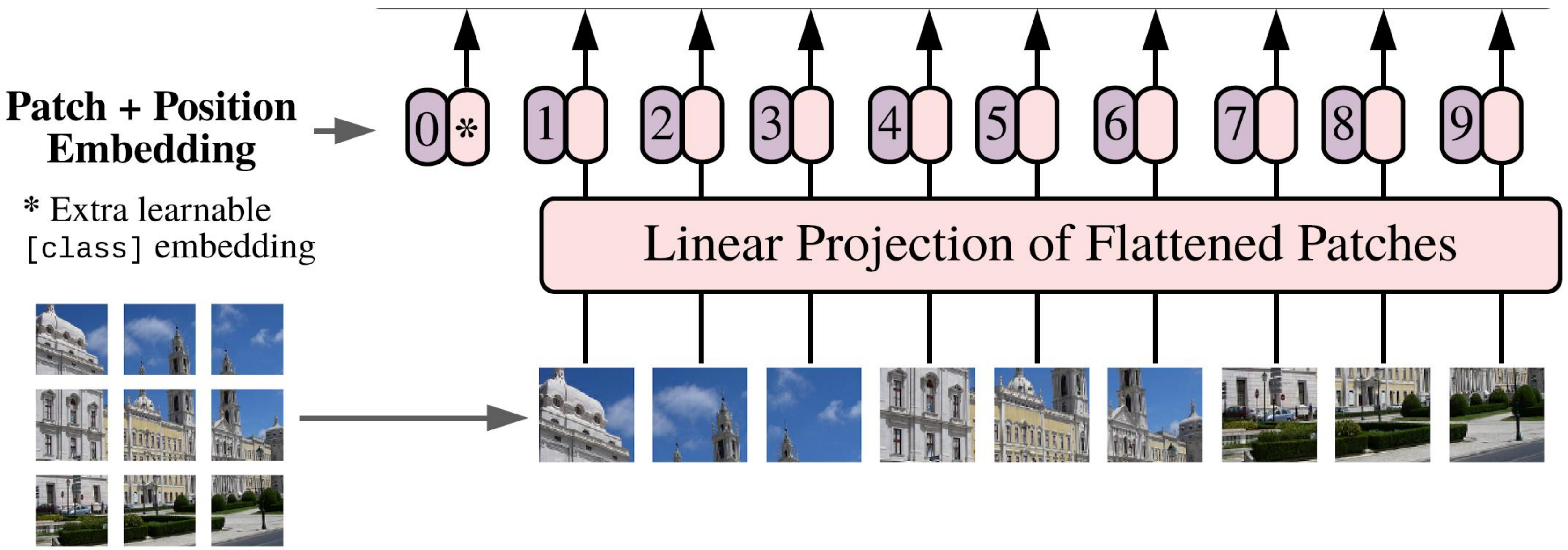
AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE. Dosovitskiy et al., ICLR’21

Vision Transformer (ViT)

- Adding positional embedding
- Prepend a learnable embedding $\mathbf{z}_0^0$

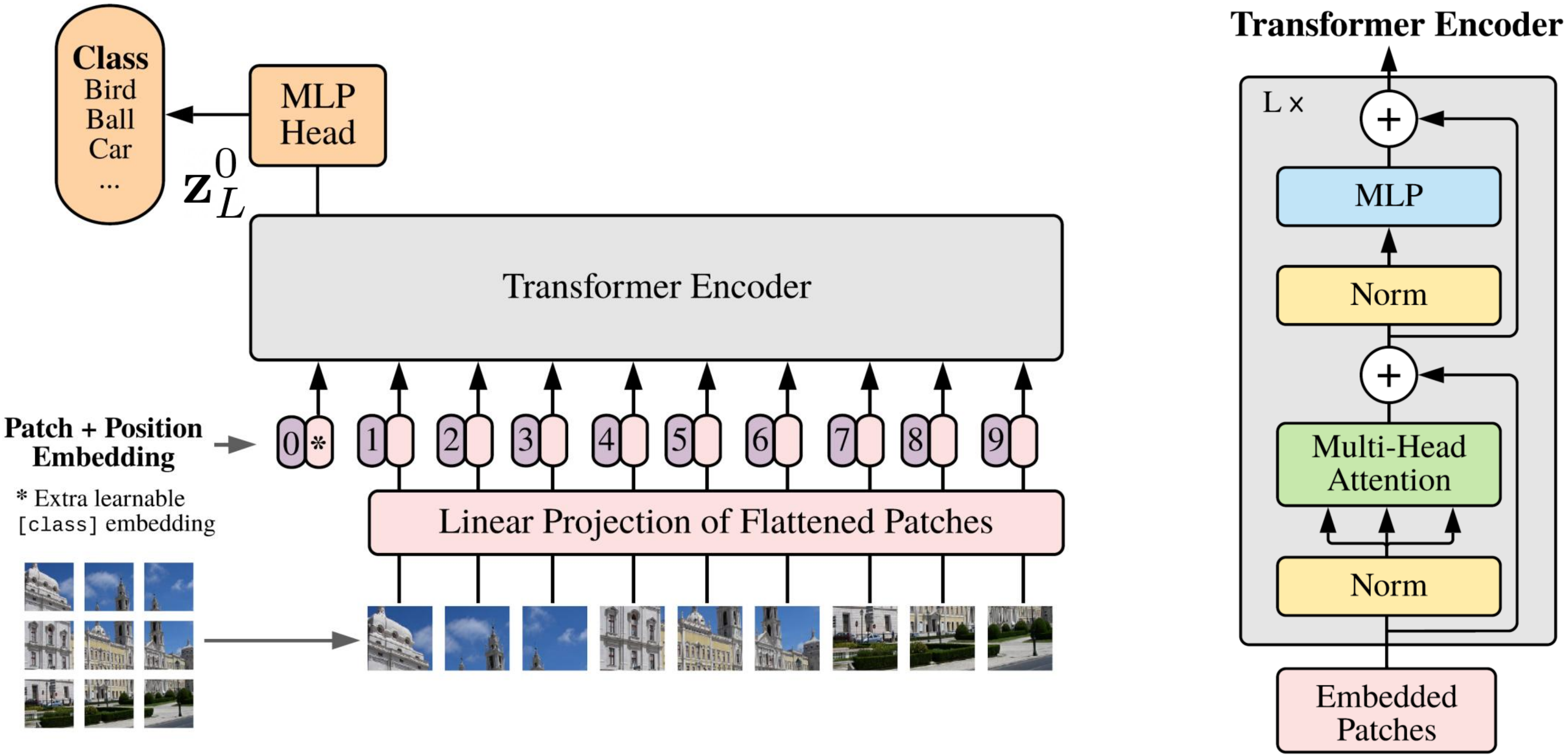
$\mathbf{z}_L^0$ Will be used as the image representation

After L attention layers



AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE. Dosovitskiy et al., ICLR'21

Vision Transformer (ViT)



AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE. Dosovitskiy et al., ICLR'21

Vision Transformer (ViT)

- Pretrain on a large-scale dataset
- Fine-tune on different tasks

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

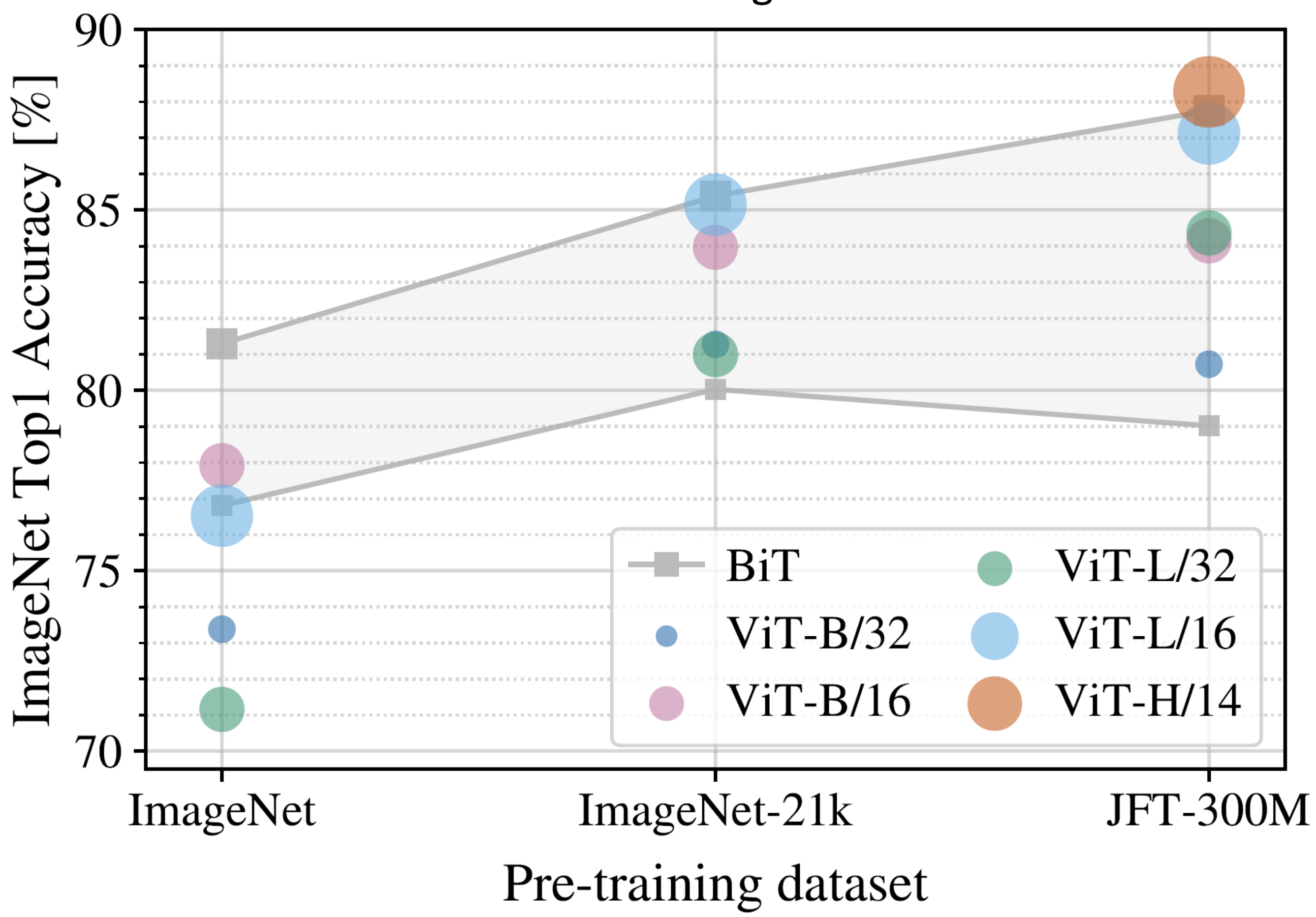
AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE. Dosovitskiy et al., ICLR'21

Vision Transformer (ViT)

1k classes
and 1.3M images

21k classes and
14M images

18k classes and
303M



Big Transfer (BiT)
- ResNets-based transfer

Vision transformer works better when pre-trained on large-scale dataset

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE. Dosovitskiy et al., ICLR'21

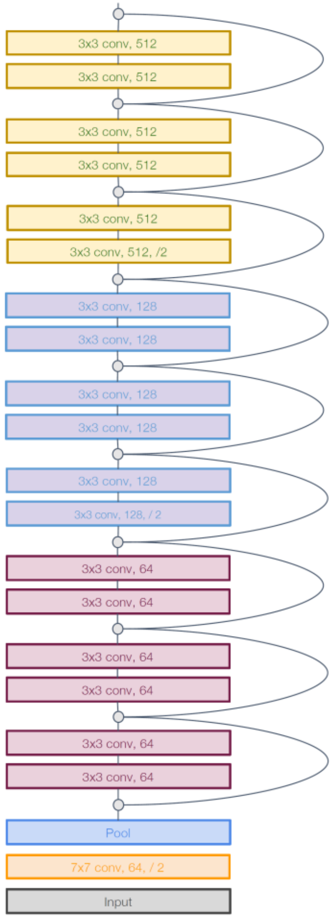
ViT vs CNN

Stage 3:
256 x 14 x 14

Stage 2:
128 x 28 x 28

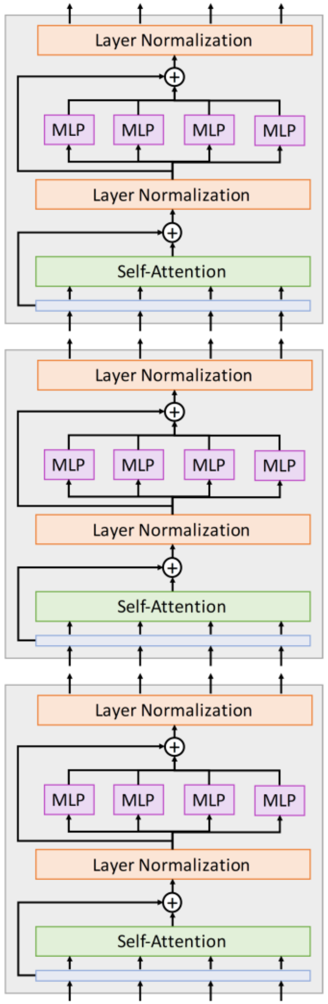
Stage 1:
64 x 56 x 56

Input:
3 x 224 x 224



CNN

Hierarchical features are useful since objects in images can occur at various scales



ViT

3rd block:
768 x 14 x 14

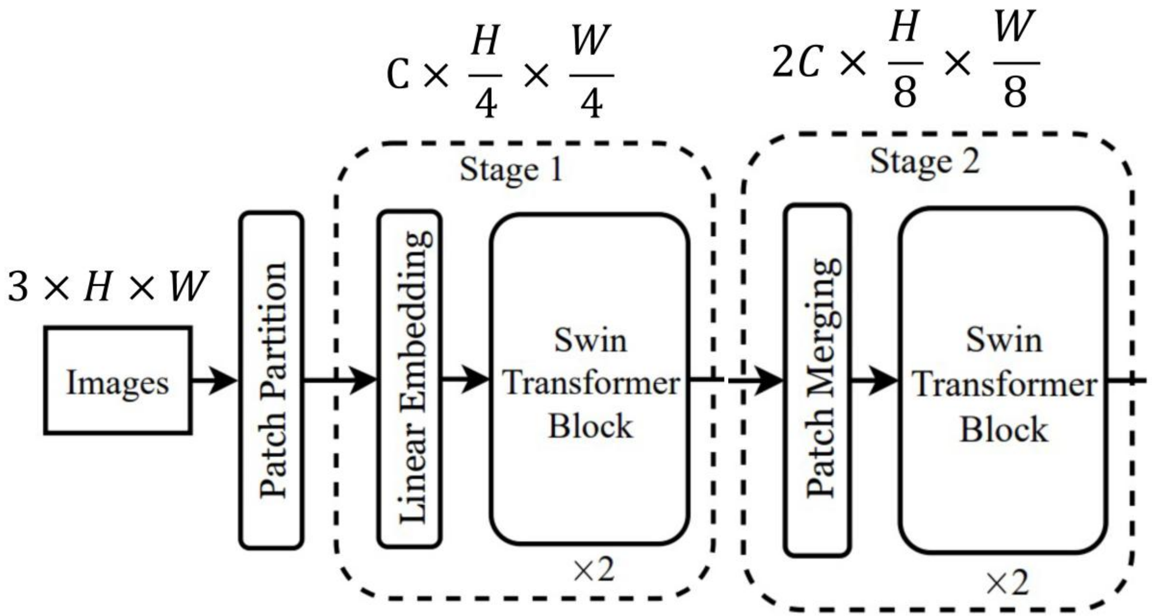
2nd block:
768 x 14 x 14

1st block:
768 x 14 x 14

Input:
3 x 224 x 224

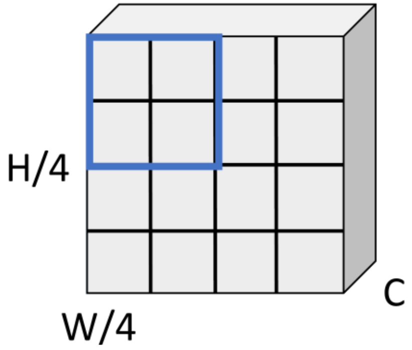
In a ViT, all blocks have same resolution and number of channels (Isotropic architecture)

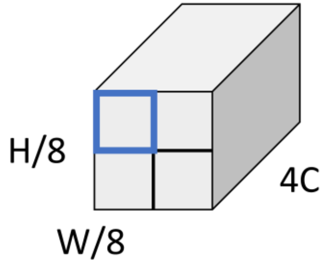
Hierarchical ViT: Swin Transformer



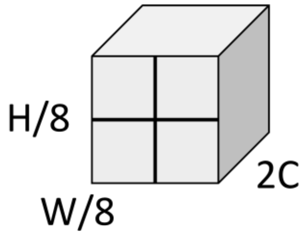
Divide image into 4x4 patches and project to C dimensions

Merge 2x2 neighborhoods; now patches are (effectively) 8x8





Concatenate groups of 2x2 features

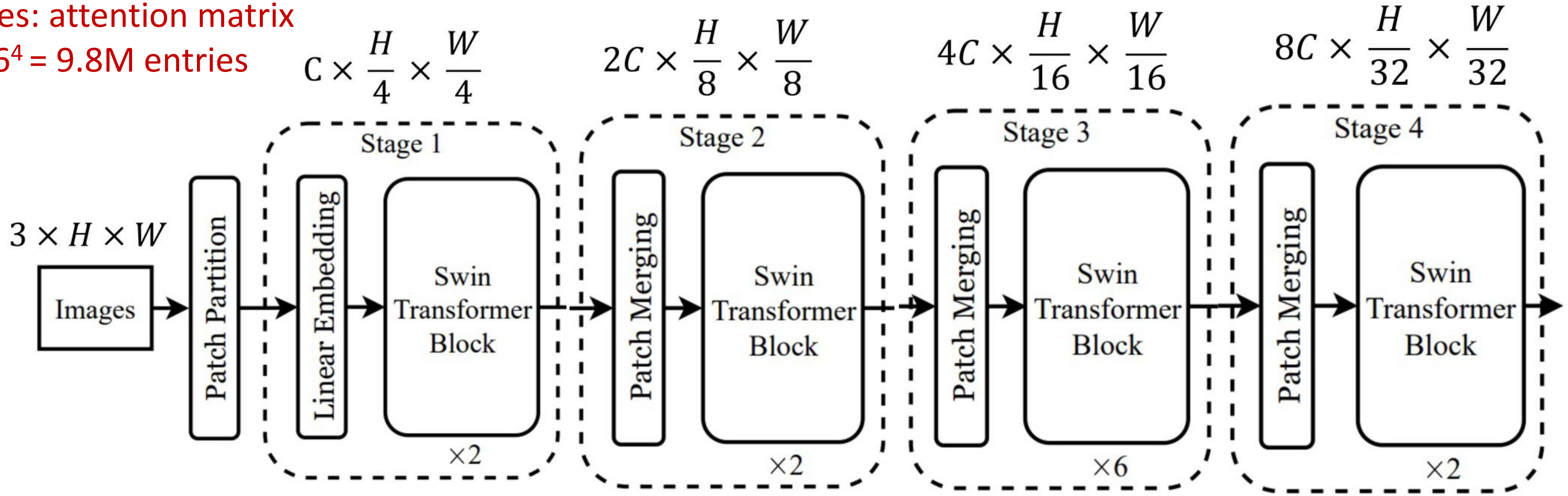


Linear projection from 4C to 2C channels (1x1 conv)

Liu et al, “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows”, CVPR 2021

Hierarchical ViT: Swin Transformer

Problem: 224x224 image with 56x56 grid of 4x4 patches: attention matrix has 56^4 = 9.8M entries



Divide image into 4x4 patches and project to C dimensions

Merge 2x2 neighborhoods; now patches are (effectively) 8x8

Merge 2x2 neighborhoods; now patches are (effectively) 16x16

Merge 2x2 neighborhoods; now patches are (effectively) 32x32

Solution: don't use full attention, instead use attention over patches

Liu et al, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows", CVPR 2021

Hierarchical ViT: Swin Transformer

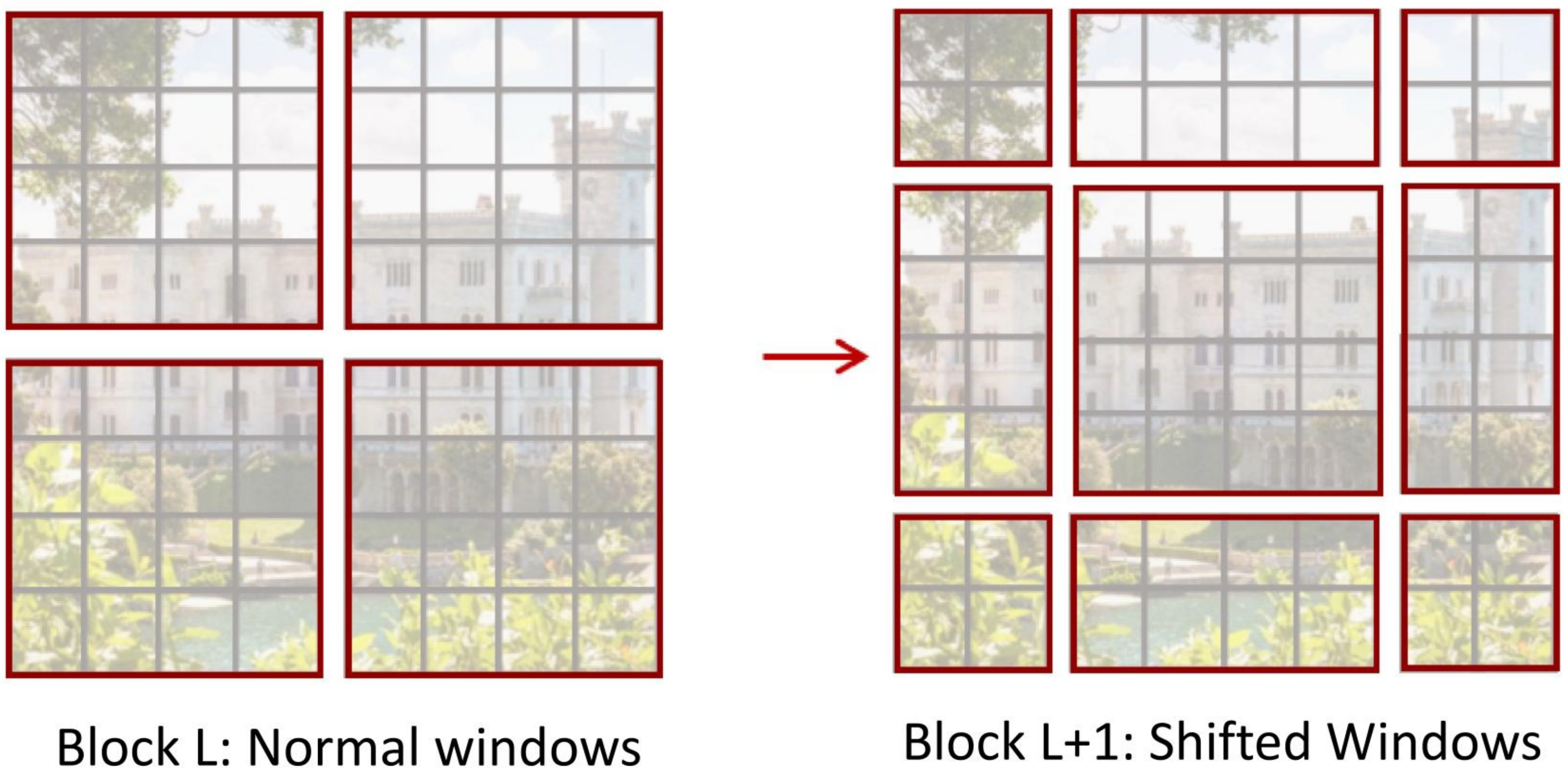
- With H x W grid of **tokens**, each attention matrix is H*H*W*W – **quadratic** in image size
- Window attention
  - Divide the image into windows of M x M tokens (here M=4)
  - Only compute attention within each window
  - Total size of attention matrices $M^4(H/M)(W/M) = M^2HW$
  - Linear in image size for fixed M! Swin uses M=7 throughout the network

Hierarchical ViT: Swin Transformer

Problem: tokens only interact with other tokens within the same window; no communication across windows

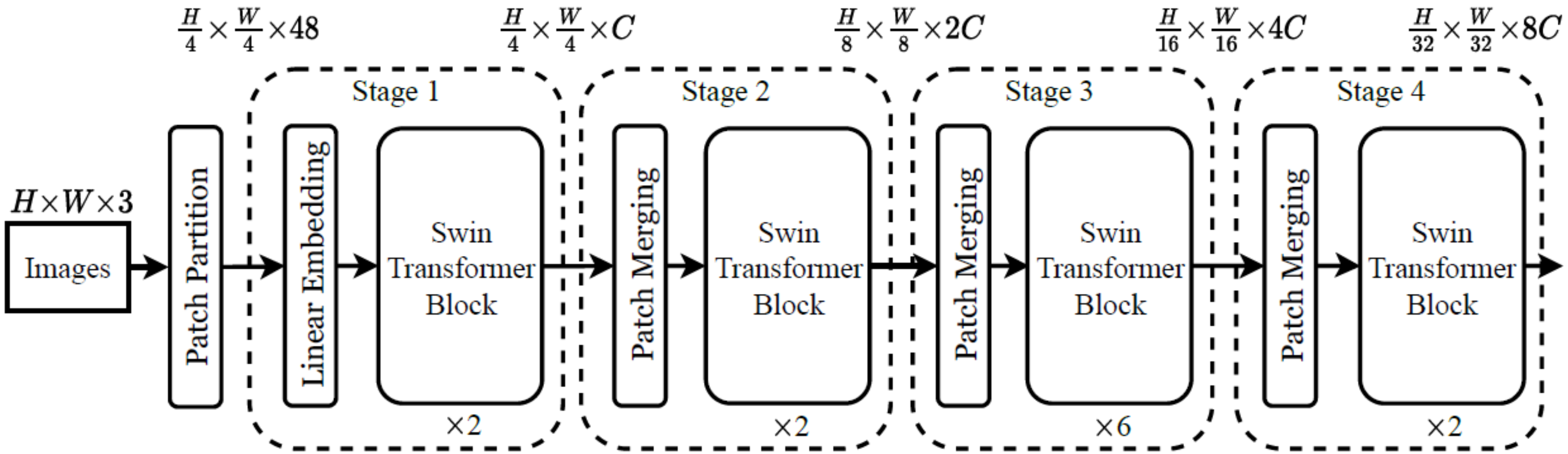
Hierarchical ViT: Swin Transformer

- Shifted Window Attention
- **Solution**: Alternate between normal windows and shifted windows in successive Transformer blocks



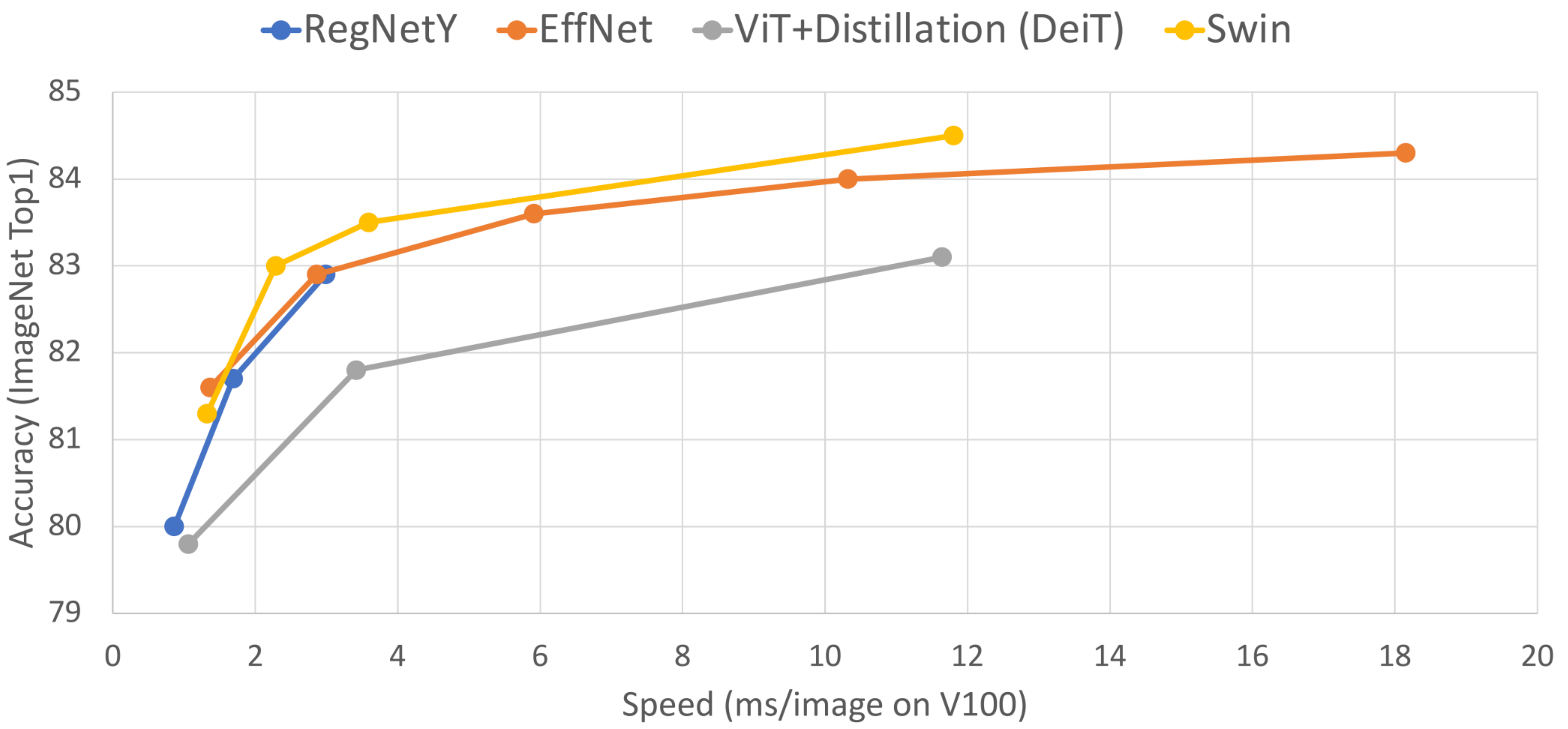
Hierarchical ViT: Swin Transformer

- Architecture variants



- Swin-T: $C = 96$, layer numbers = $\{2, 2, 6, 2\}$
- Swin-S: $C = 96$, layer numbers =$\{2, 2, 18, 2\}$
- Swin-B: $C = 128$, layer numbers =$\{2, 2, 18, 2\}$
- Swin-L: $C = 192$, layer numbers =$\{2, 2, 18, 2\}$

Hierarchical ViT: Swin Transformer



Summary

- Transformers
  - Can capture long-distance dependencies (global attention)
  - Computationally efficient, more parallelizable
- Vision transformers
  - Works better when pre-trained on large scale datasets (e.g., 300M images)
  - Swin transformer

Further Reading

- Transformer: Attention is all you need https://arxiv.org/abs/1706.03762
- Vision transformer: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale https://arxiv.org/abs/2010.11929
- Swin Transformer: Hierarchical Vision Transformer using Shifted Windows https://arxiv.org/abs/2103.14030