Transformers for Vision

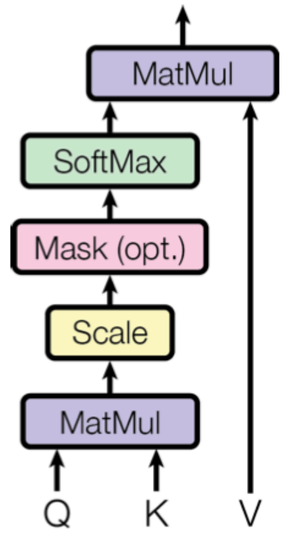
Deep Learning 1

Baranouskaya Darya (talk by Ildus Sadrtdinov)

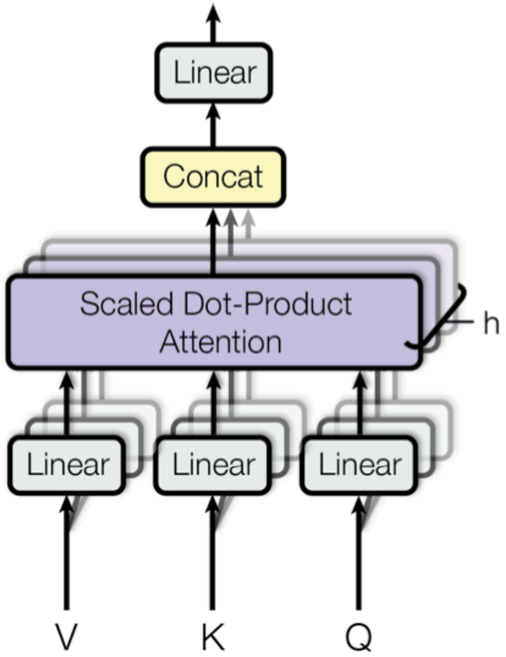
19.01.2024

Transformer

Scaled Dot-Product Attention



Multi-Head Attention



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

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

Transformer

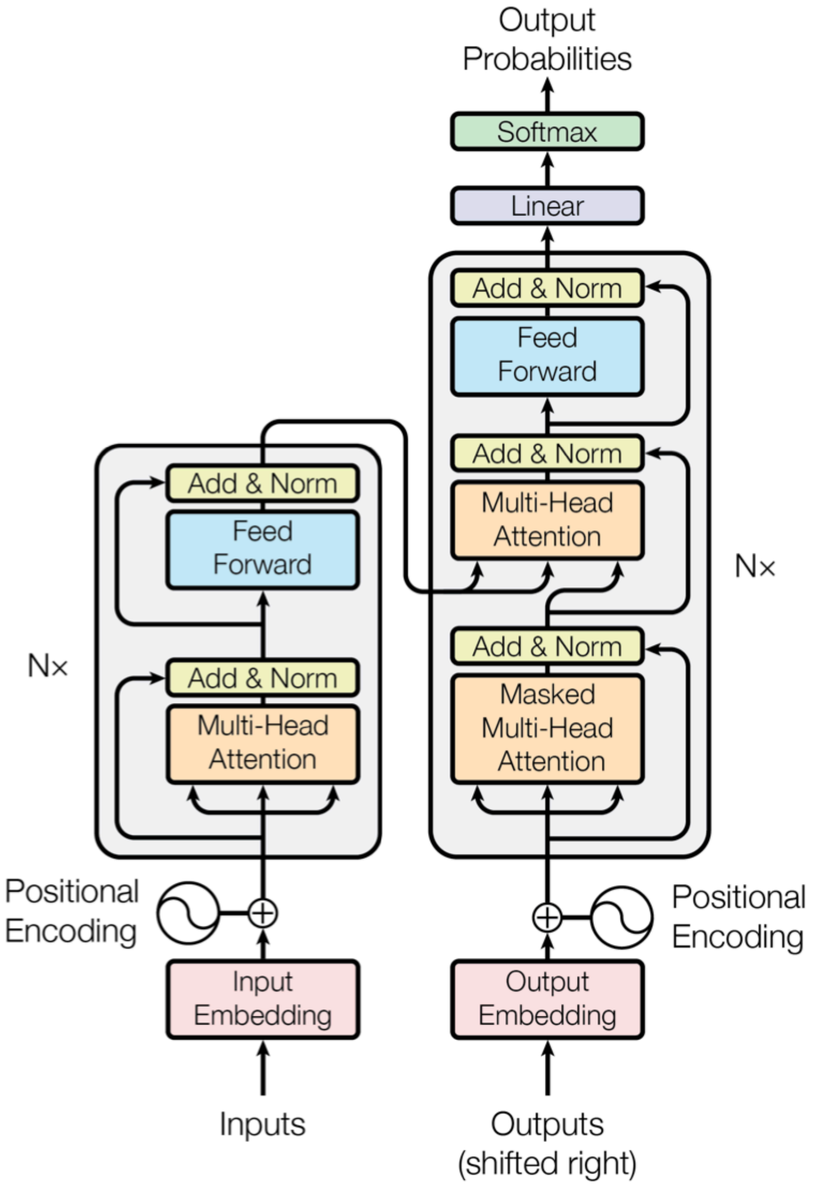
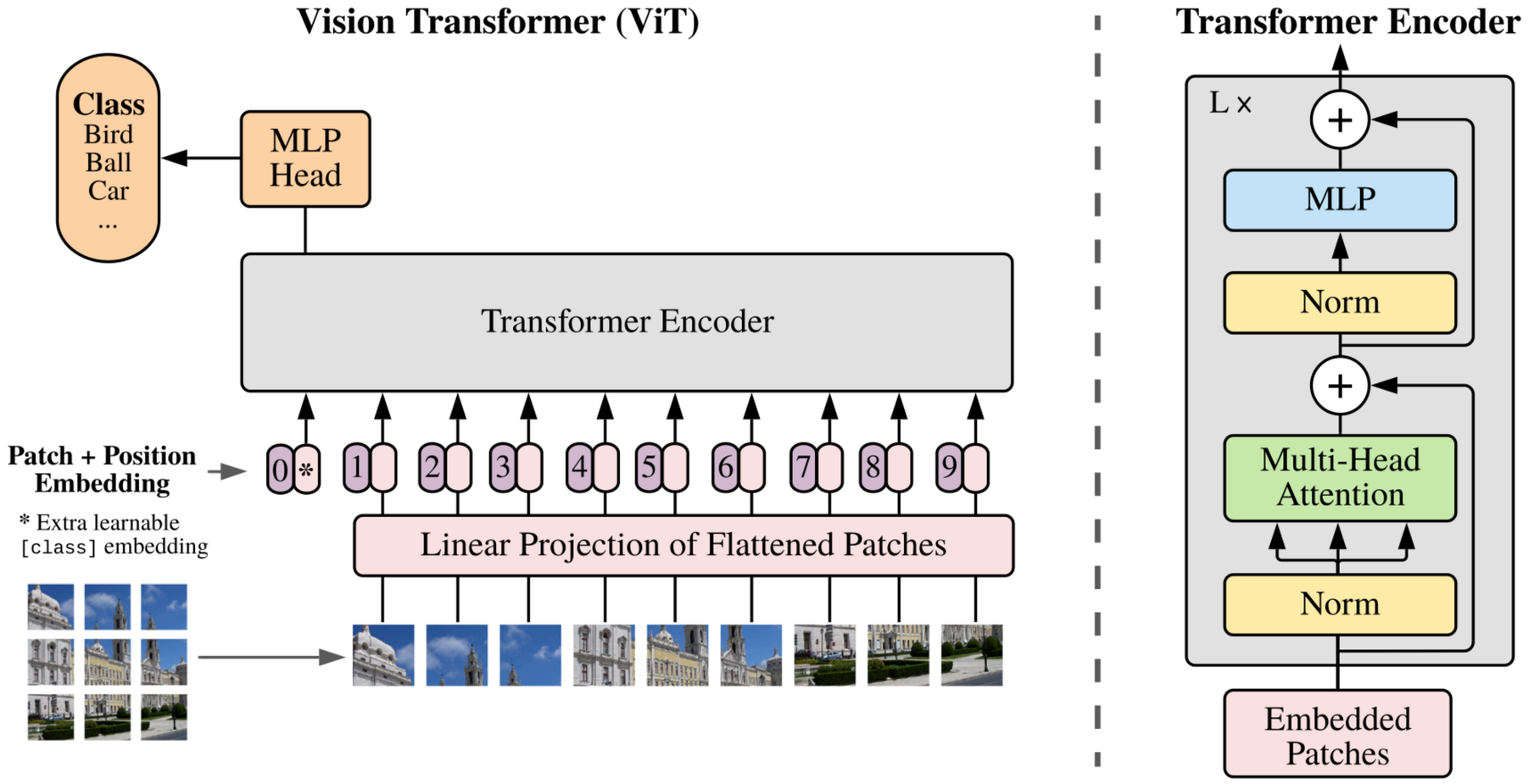


Figure 1: The Transformer - model architecture.

VIT



«An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale» A.Dosovitskiy et. al, 2020

How to train your ViT?

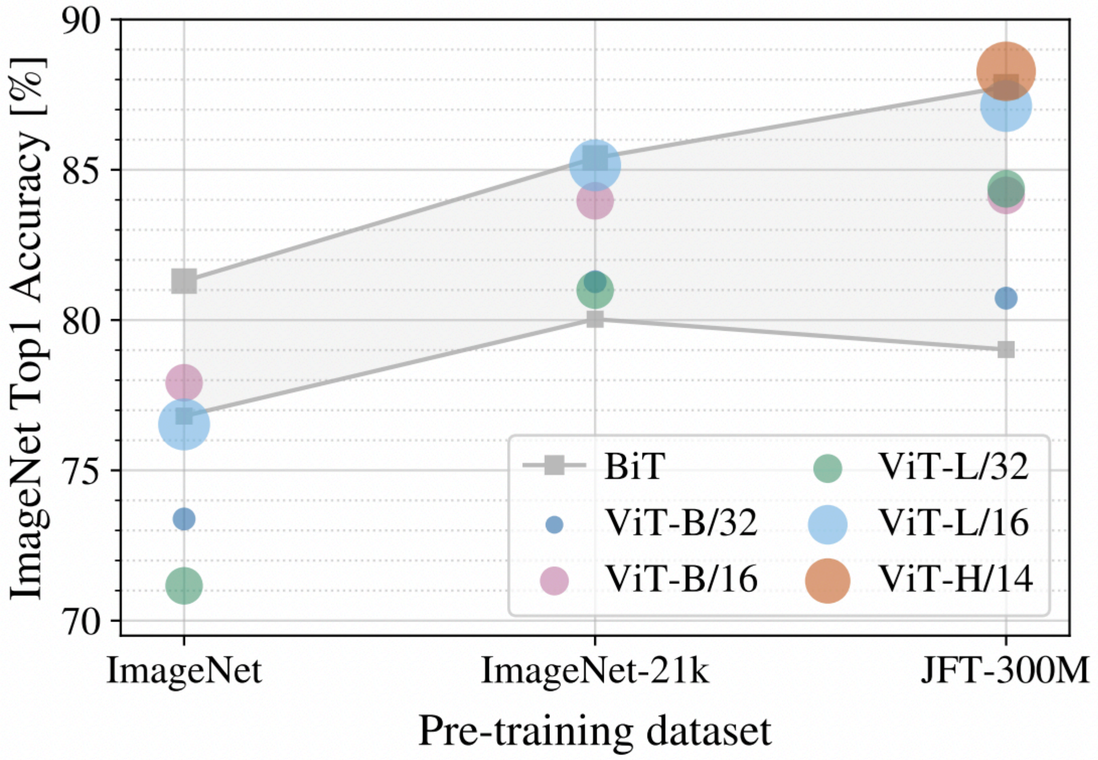
- Use checkpoints that were pre-trained on more upstream data
- Applying data augmentation and model regularization
- To fine-tune select the model with the best validation performance on pretrain data
- Use warmup and a high weight decay (of 0.1)!
- Adam $\beta_1 = 0.9$, $\beta_2 = 0.999$

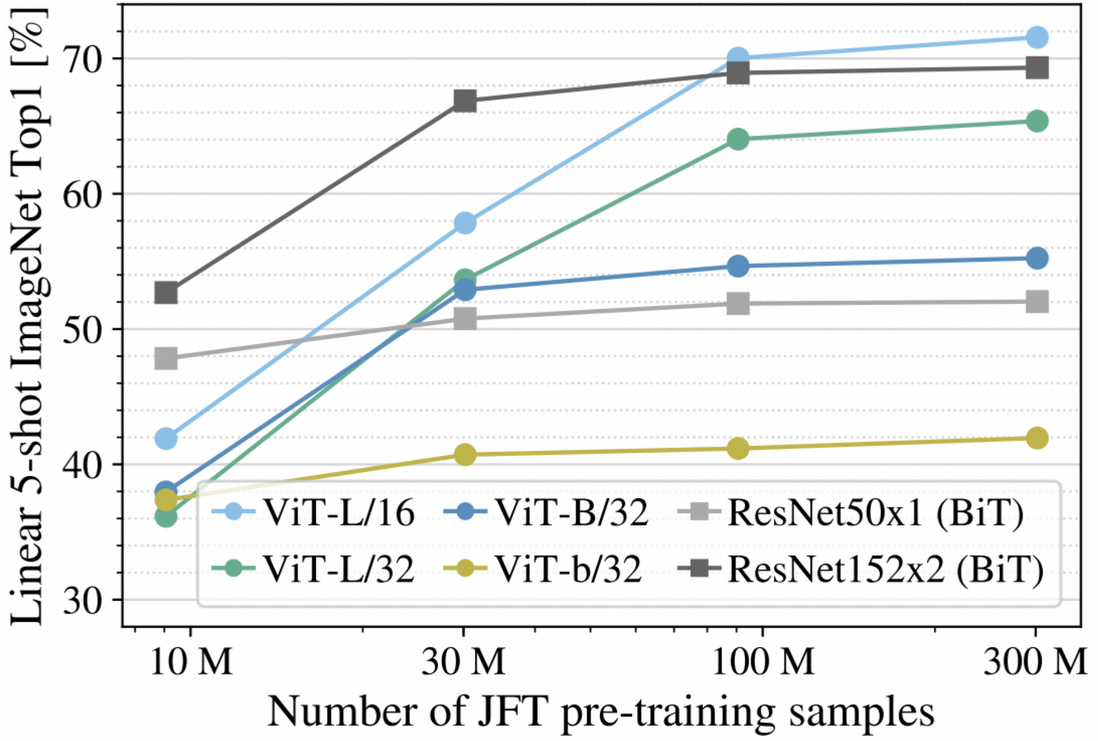
.

«How to train your ViT? Data, Augmentation, and Regularization in Vision Transformers » A. Steiner et. al, 2022

VIT

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





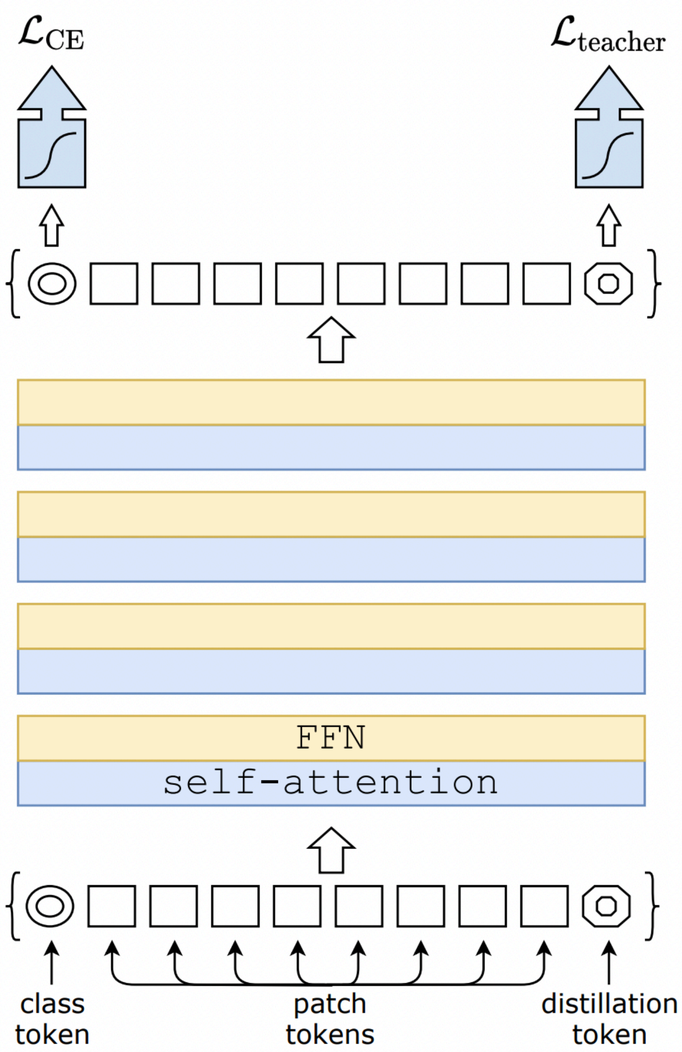
«An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale» A.Dosovitskiy et. al, 2020

DeiT

$$\mathcal{L}_{\text{global}} = (1-\lambda)\mathcal{L}_{\text{CE}}(\psi(Z_{\text{s}}), y) + \lambda\tau^2 \text{KL}(\psi(Z_{\text{s}}/\tau), \psi(Z_{\text{t}}/\tau))$$

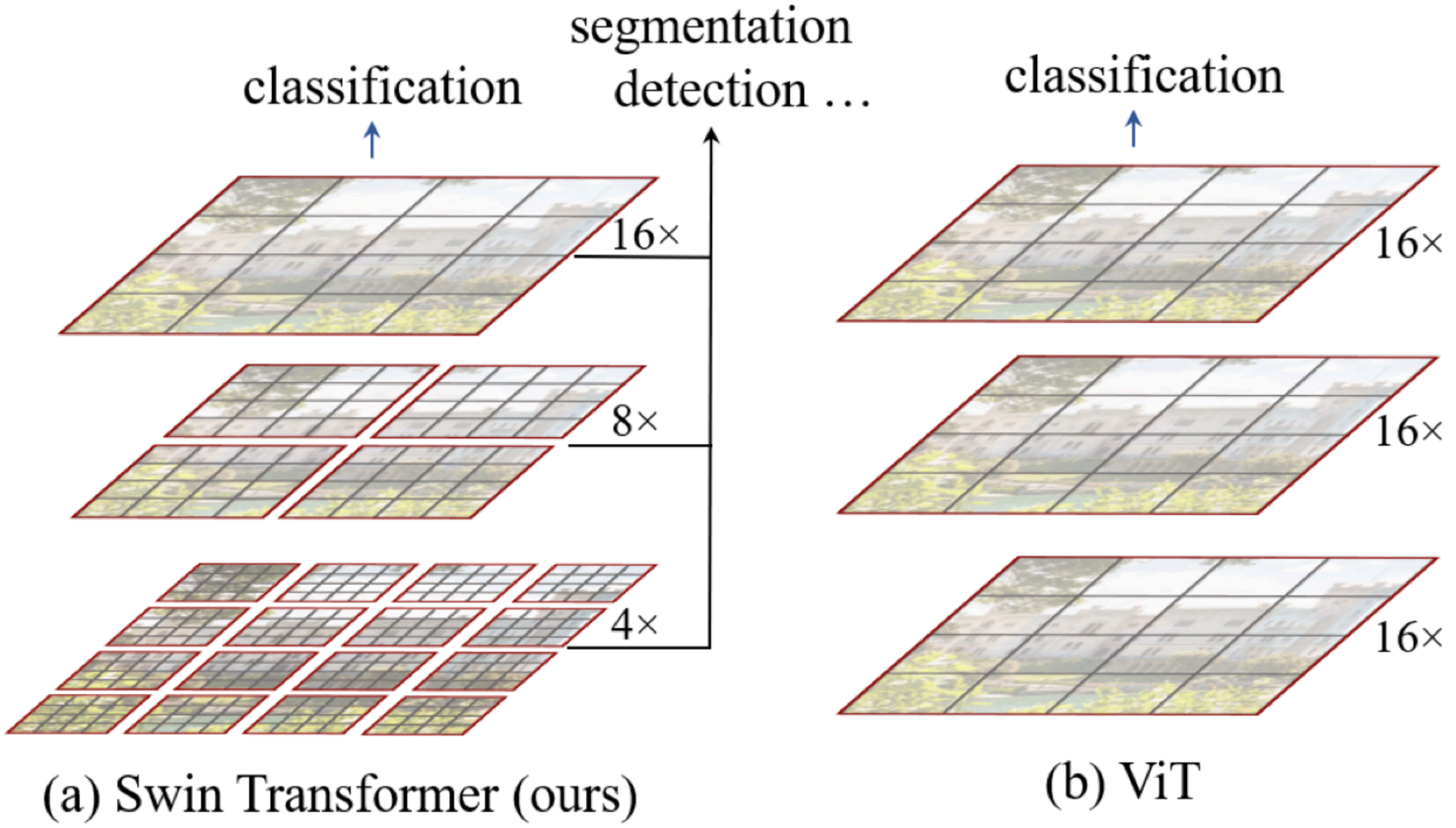
$$\mathcal{L}^{\text{hardDistill}}_{\text{global}} = \frac{1}{2}\mathcal{L}_{\text{CE}}(\psi(Z_s), y) + \frac{1}{2}\mathcal{L}_{\text{CE}}(\psi(Z_s), y_{\text{t}})$$

method ↓	Supervision label	Supervision teacher	ImageNet top-1 (%) Ti 224	S 224	B 224	B↑384
DeiT– no distillation	✓	✗	72.2	79.8	81.8	83.1
DeiT– usual distillation	✗	soft	72.2	79.8	81.8	83.2
DeiT– hard distillation	✗	hard	74.3	80.9	83.0	84.0
DeiT⚗: class embedding	✓	hard	73.9	80.9	83.0	84.2
DeiT⚗: distil. embedding	✓	hard	74.6	81.1	83.1	84.4
DeiT⚗: class+distillation	✓	hard	74.5	81.2	83.4	84.5



«Training data-efficient image transformers & distillation through attention» H.Touvron et. al, 2021

SWIN



«Swin Transformer: Hierarchical Vision Transformer using Shifted Windows» Ze Liu et. al, 2021

SWIN

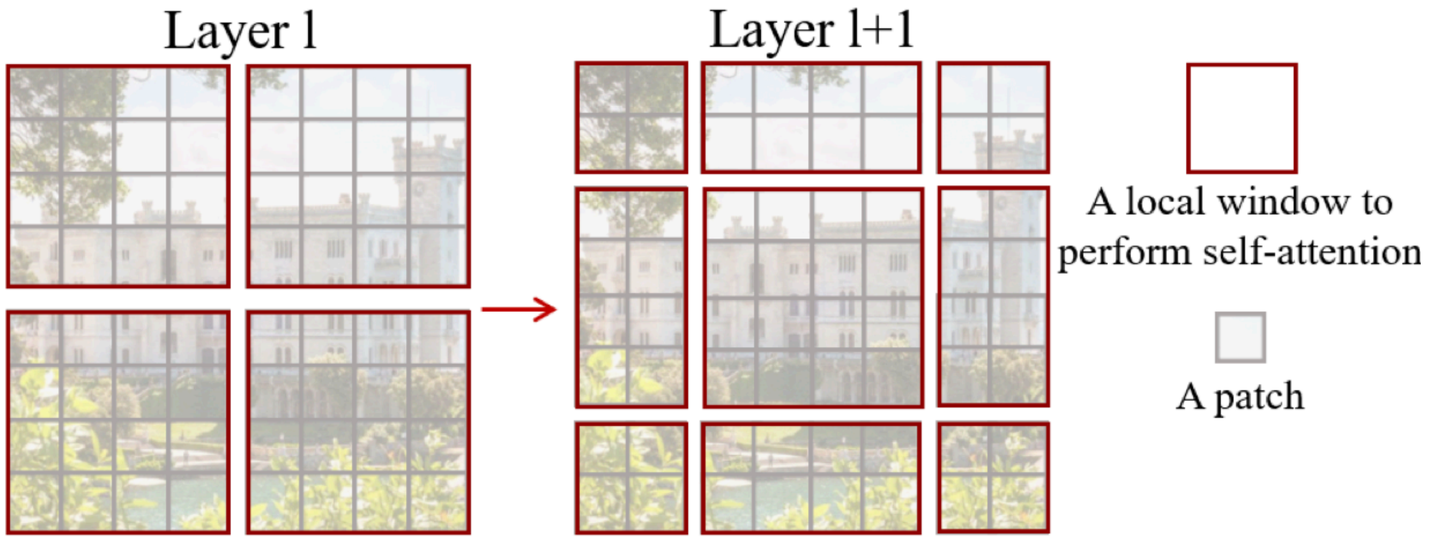


Figure 2. An illustration of the *shifted window* approach for computing self-attention in the proposed Swin Transformer architecture. In layer l (left), a regular window partitioning scheme is adopted, and self-attention is computed within each window. In the next layer $l + 1$ (right), the window partitioning is shifted, resulting in new windows. The self-attention computation in the new windows crosses the boundaries of the previous windows in layer l, providing connections among them.

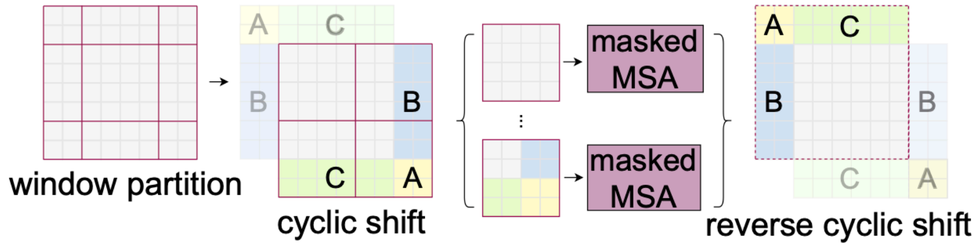


Figure 4. Illustration of an efficient batch computation approach for self-attention in shifted window partitioning.

«Swin Transformer: Hierarchical Vision Transformer using Shifted Windows» Ze Liu et. al, 2021

SWIN

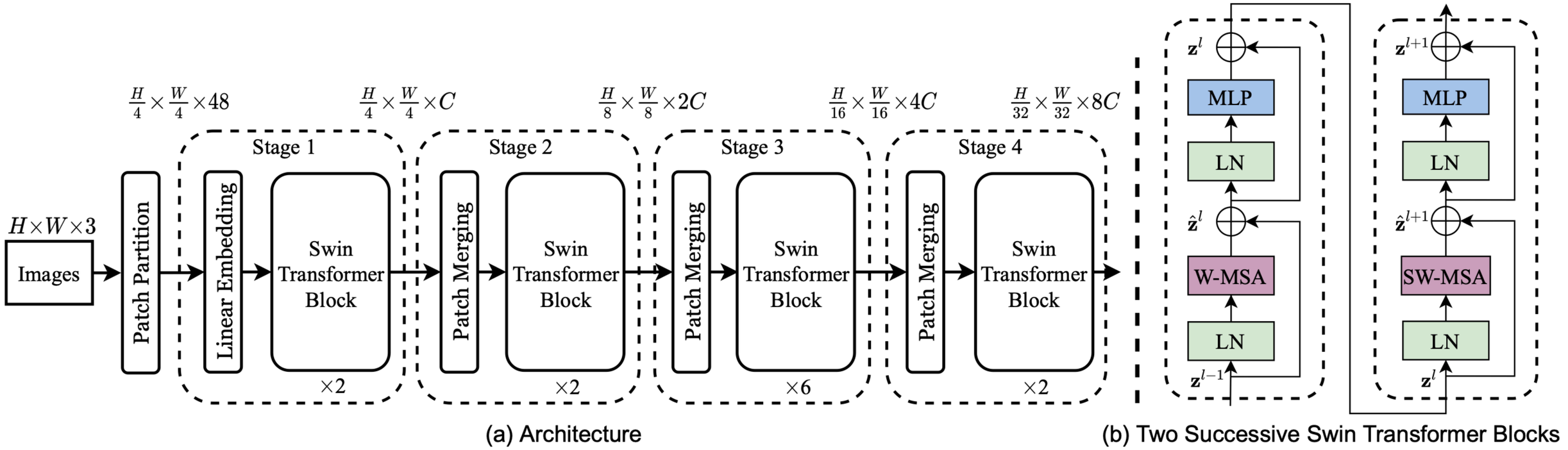
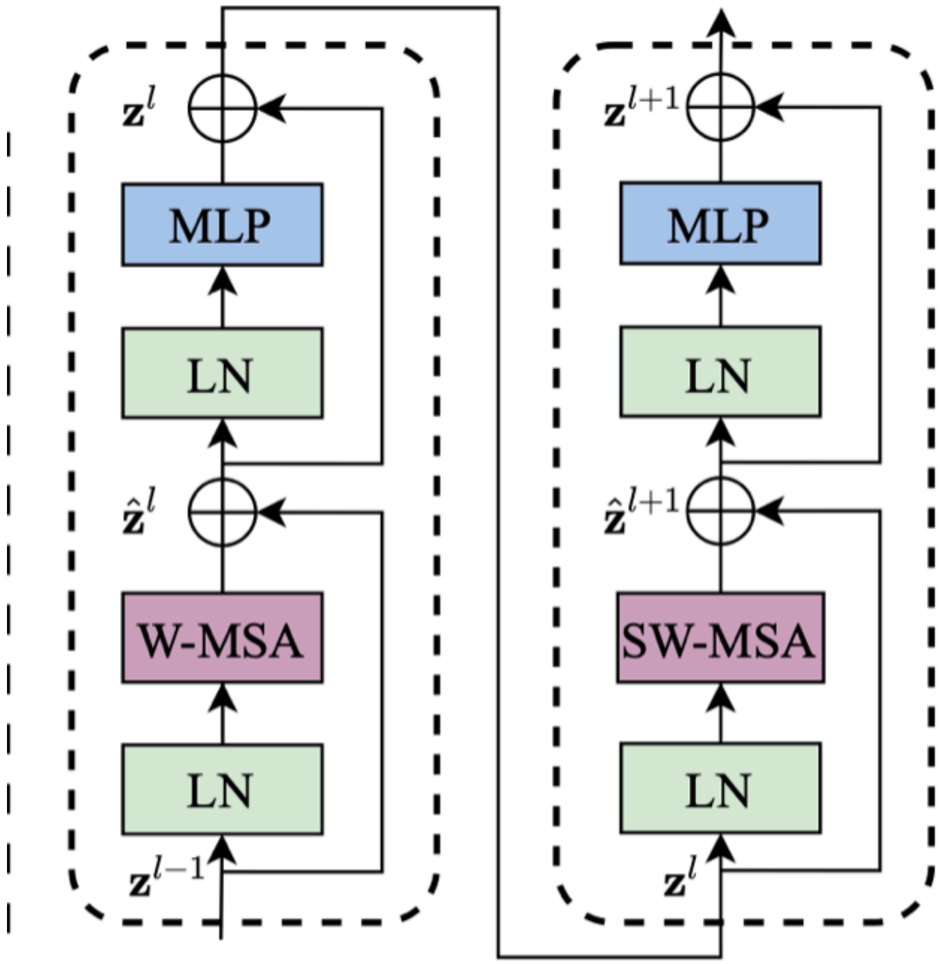


Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

«Swin Transformer: Hierarchical Vision Transformer using Shifted Windows» Ze Liu et. al, 2021

SWIN



$$\hat{\mathbf{z}}^l = \text{W-MSA}\left(\text{LN}\left(\mathbf{z}^{l-1}\right)\right) + \mathbf{z}^{l-1},$$
$$\mathbf{z}^l = \text{MLP}\left(\text{LN}\left(\hat{\mathbf{z}}^l\right)\right) + \hat{\mathbf{z}}^l,$$
$$\hat{\mathbf{z}}^{l+1} = \text{SW-MSA}\left(\text{LN}\left(\mathbf{z}^l\right)\right) + \mathbf{z}^l,$$
$$\mathbf{z}^{l+1} = \text{MLP}\left(\text{LN}\left(\hat{\mathbf{z}}^{l+1}\right)\right) + \hat{\mathbf{z}}^{l+1},$$

«Swin Transformer: Hierarchical Vision Transformer using Shifted Windows» Ze Liu et. al, 2021

SWIN

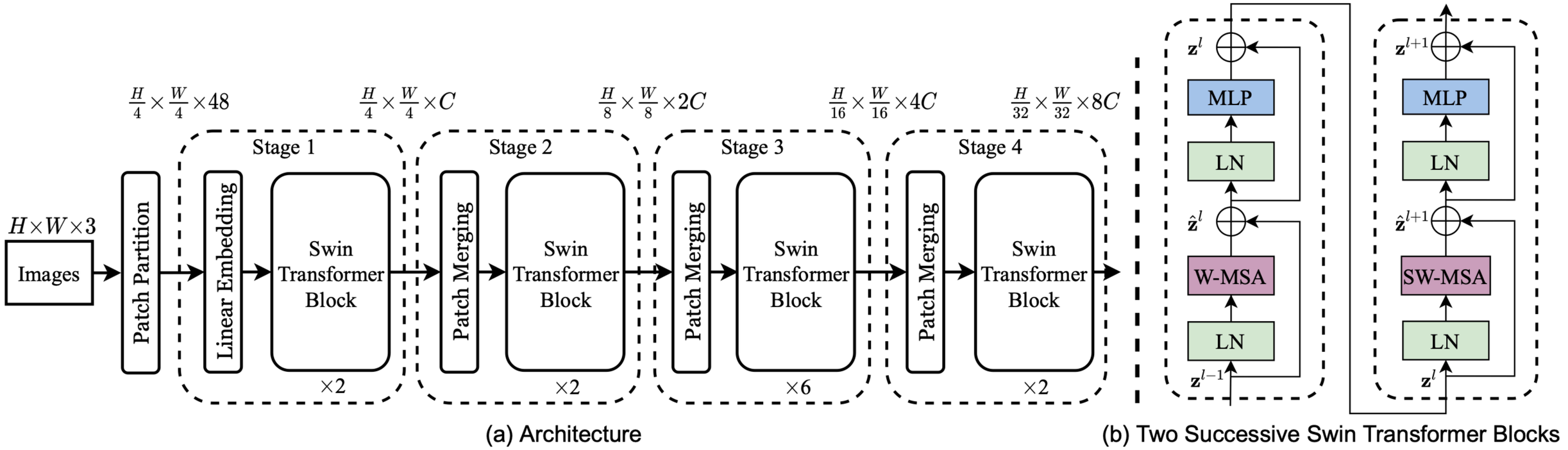
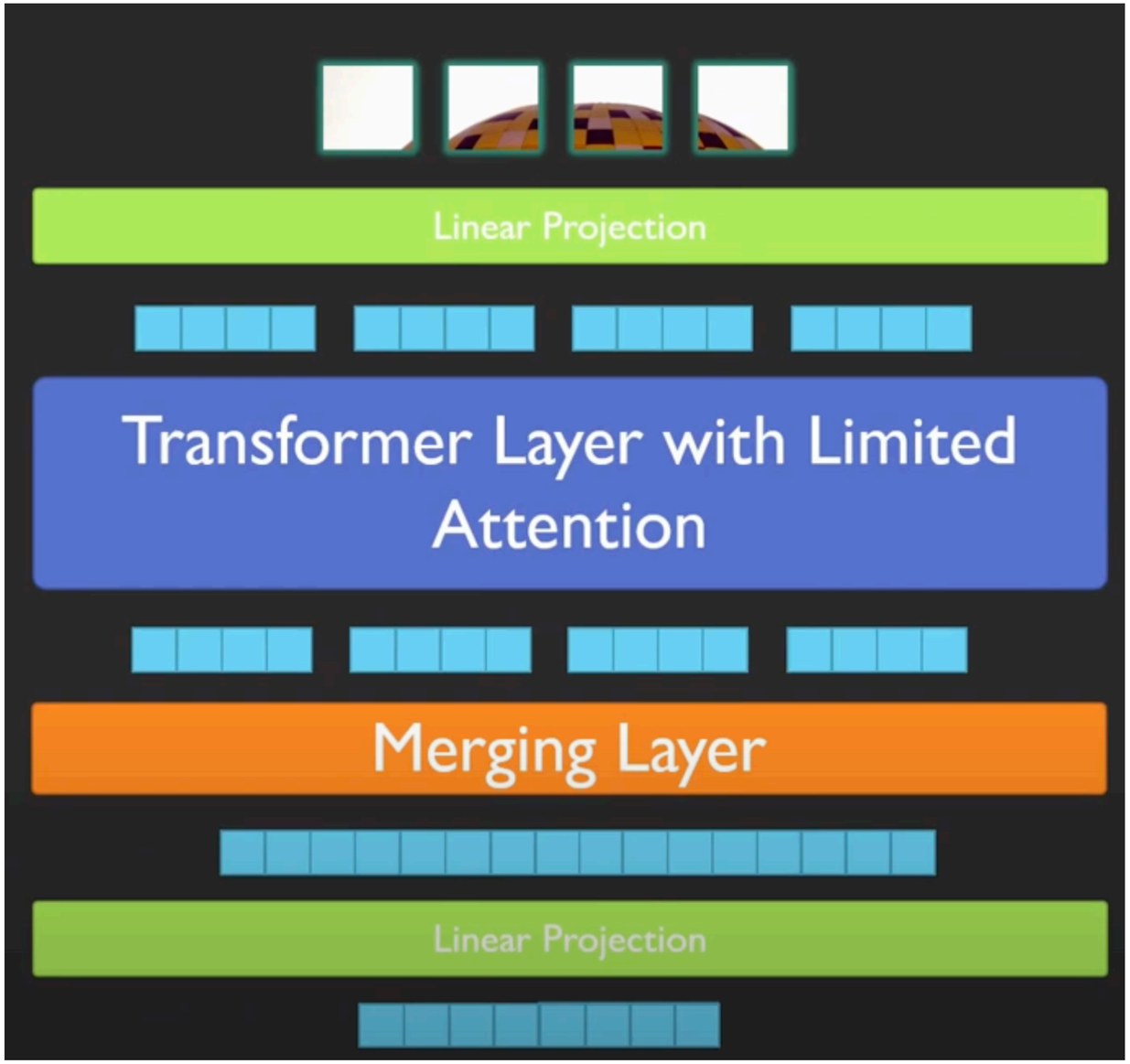


Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

«Swin Transformer: Hierarchical Vision Transformer using Shifted Windows» Ze Liu et. al, 2021

SWIN



«Swin Transformer: Hierarchical Vision Transformer using Shifted Windows» Ze Liu et. al, 2021

SWIN

Relative position bias

$$\text{Attention}(Q, K, V) = \text{SoftMax}(QK^T/\sqrt{d} + B)V,$$

$$Q, K, V \in \mathbb{R}^{M^2 \times d}$$

$$B \in \mathbb{R}^{M^2 \times M^2}$$

B are taken from $\hat{B}$, $\hat{B} \in \mathbb{R}^{(2M-1)\times(2M-1)}$

«Swin Transformer: Hierarchical Vision Transformer using Shifted Windows» Ze Liu et. al, 2021

SWIN

(a) Regular ImageNet-1K trained models

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

(b) ImageNet-22K pre-trained models

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

«Swin Transformer: Hierarchical Vision Transformer using Shifted Windows» Ze Liu et. al, 2021

SWIN

COCO object detection

(a) Various frameworks

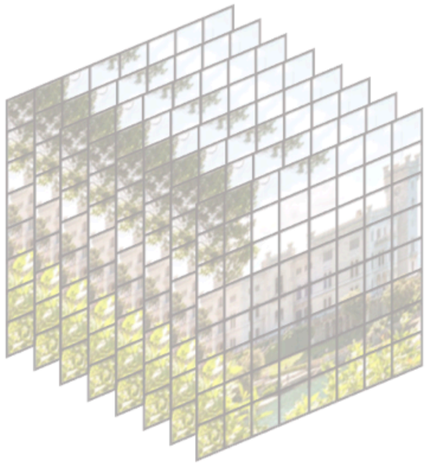
Method	Backbone	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	#param.	FLOPs	FPS
Cascade Mask R-CNN	R-50	46.3	64.3	50.5	82M	739G	18.0
	Swin-T	**50.5**	**69.3**	**54.9**	86M	745G	15.3
ATSS	R-50	43.5	61.9	47.0	32M	205G	28.3
	Swin-T	**47.2**	**66.5**	**51.3**	36M	215G	22.3
RepPointsV2	R-50	46.5	64.6	50.3	42M	274G	13.6
	Swin-T	**50.0**	**68.5**	**54.2**	45M	283G	12.0
Sparse R-CNN	R-50	44.5	63.4	48.2	106M	166G	21.0
	Swin-T	**47.9**	**67.3**	**52.3**	110M	172G	18.4

ADE20K semantic segmentation

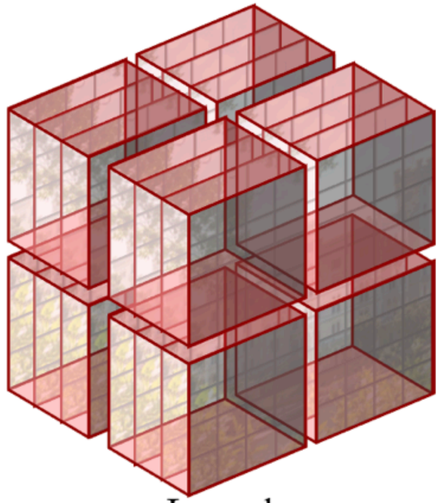
Method	Backbone	val mIoU	test score	#param.	FLOPs	FPS
DANet [23]	ResNet-101	45.2	-	69M	1119G	15.2
DLab.v3+ [11]	ResNet-101	44.1	-	63M	1021G	16.0
ACNet [24]	ResNet-101	45.9	38.5	-		
DNL [71]	ResNet-101	46.0	56.2	69M	1249G	14.8
OCRNet [73]	ResNet-101	45.3	56.0	56M	923G	19.3
UperNet [69]	ResNet-101	44.9	-	86M	1029G	20.1
OCRNet [73]	HRNet-w48	45.7	-	71M	664G	12.5
DLab.v3+ [11]	ResNeSt-101	46.9	55.1	66M	1051G	11.9
DLab.v3+ [11]	ResNeSt-200	48.4	-	88M	1381G	8.1
SETR [81]	T-Large‡	50.3	61.7	308M	-	-
UperNet	DeiT-S†	44.0	-	52M	1099G	16.2
UperNet	Swin-T	46.1	-	60M	945G	18.5
UperNet	Swin-S	49.3	-	81M	1038G	15.2
UperNet	Swin-B‡	51.6	-	121M	1841G	8.7
UperNet	Swin-L‡	**53.5**	**62.8**	234M	3230G	6.2

«Swin Transformer: Hierarchical Vision Transformer using Shifted Windows» Ze Liu et. al, 2021

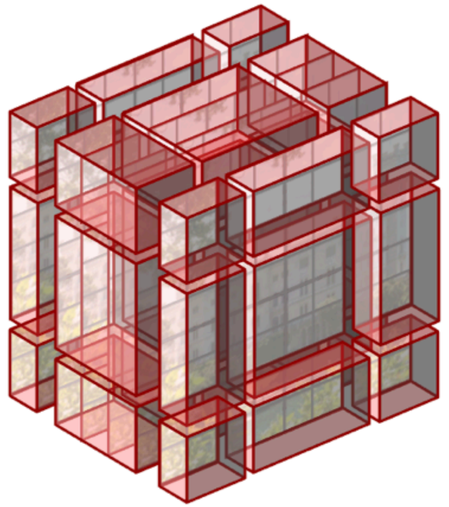
Video Swin Transformer



3D tokens: T'×H'×W' = 8×8×8
Window size: P×M×M = 4×4×4



Layer l
window: 2×2×2=8



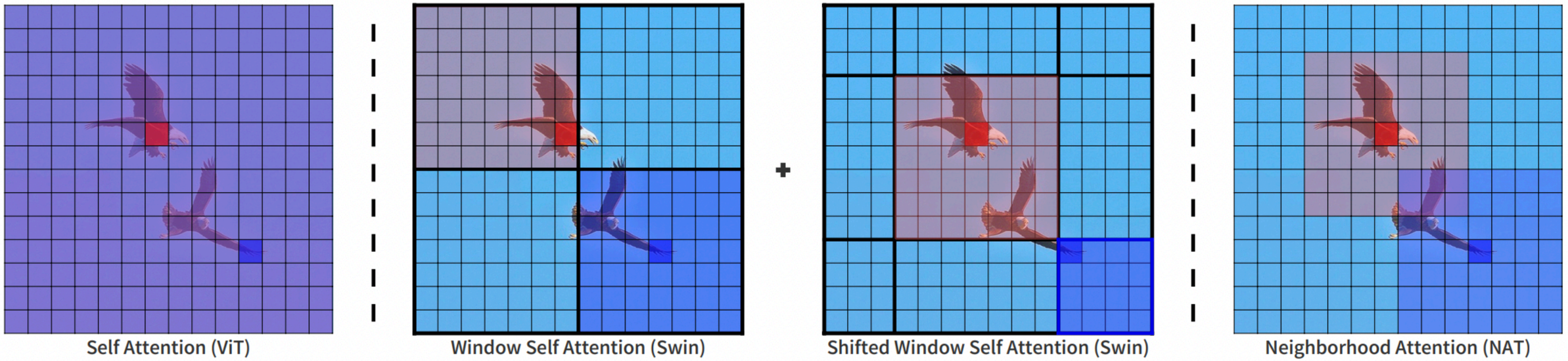
Layer l+1
window: 3×3×3=27

3D local window to perform self-attention

A token

«Video Swin Transformer» Ze Liu et. al, 2021

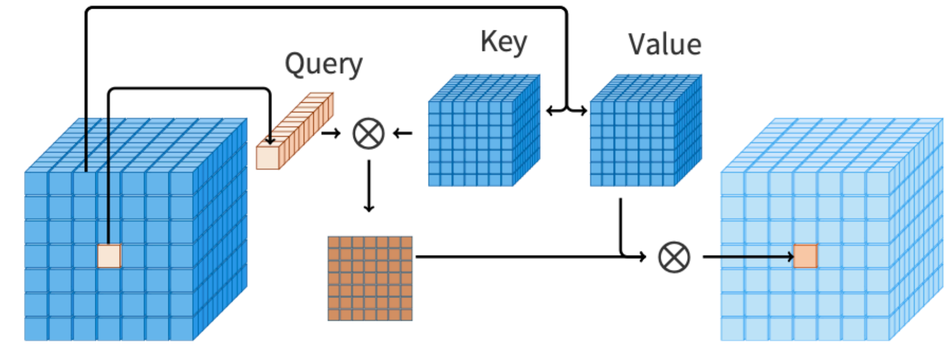
Neighborhood Attention Transformer



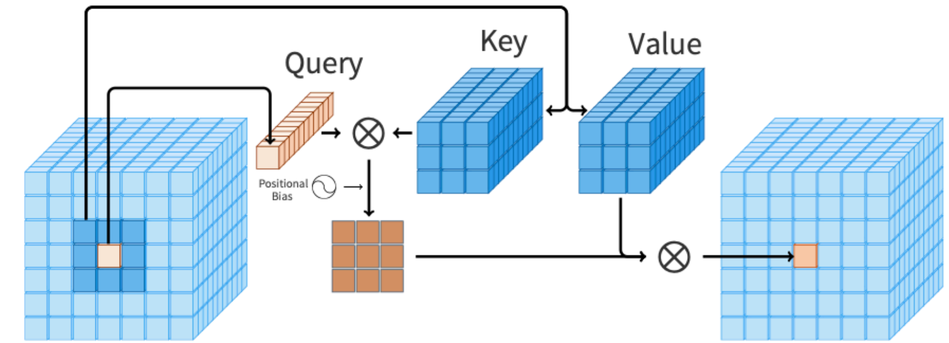
«Neighborhood Attention Transformer» A. Hassani et. al, 2022

Neighborhood Attention Transformer

Self Attention



Neighborhood Attention



- k nearest neighboring key projections

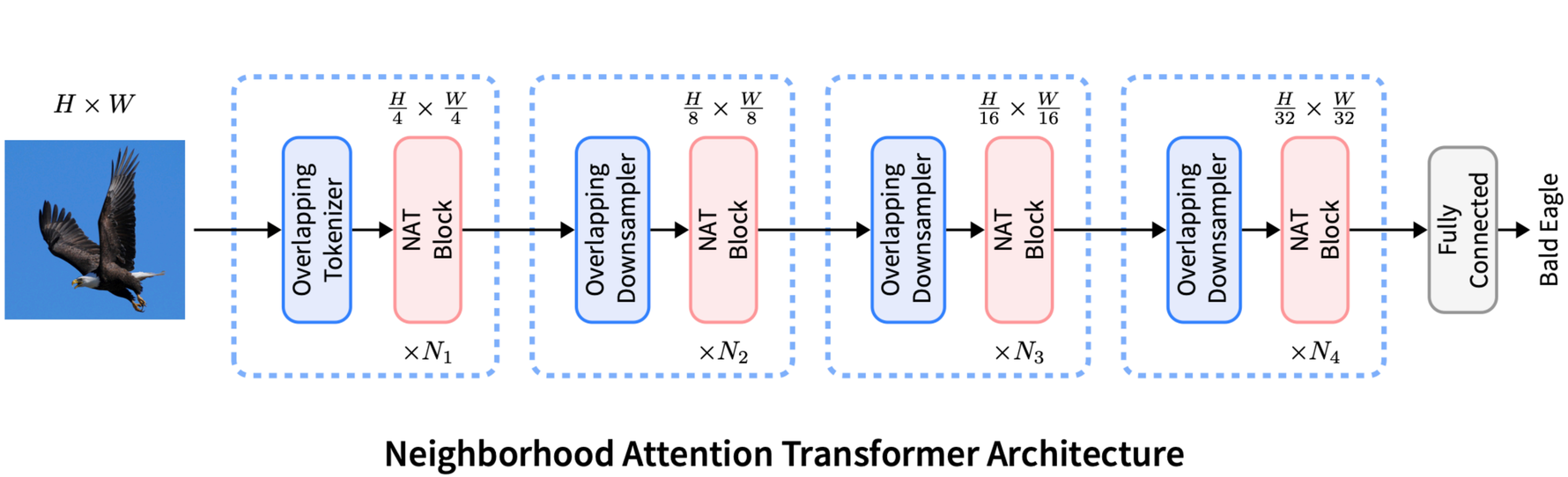
$$\mathbf{A}_i^k = \begin{bmatrix} Q_i K^T_{\rho_1(i)} + B_{(i,\rho_1(i))} \\ Q_i K^T_{\rho_2(i)} + B_{(i,\rho_2(i))} \\ \vdots \\ Q_i K^T_{\rho_k(i)} + B_{(i,\rho_k(i))} \end{bmatrix}, \qquad (2)$$

$$\mathbf{V}_i^k = \begin{bmatrix} V^T_{\rho_1(i)} & V^T_{\rho_2(i)} & \dots & V^T_{\rho_k(i)} \end{bmatrix}^T. \qquad (3)$$

$$\mathrm{NA}_k(i) = softmax\left(\frac{\mathbf{A}_i^k}{\sqrt{d}}\right)\mathbf{V}_i^k, \qquad (4)$$

«Neighborhood Attention Transformer» A. Hassani et. al, 2022

Neighborhood Attention Transformer



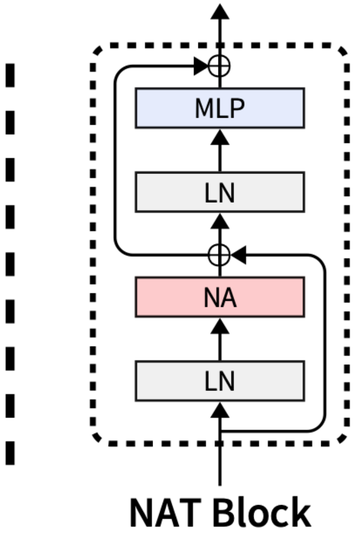
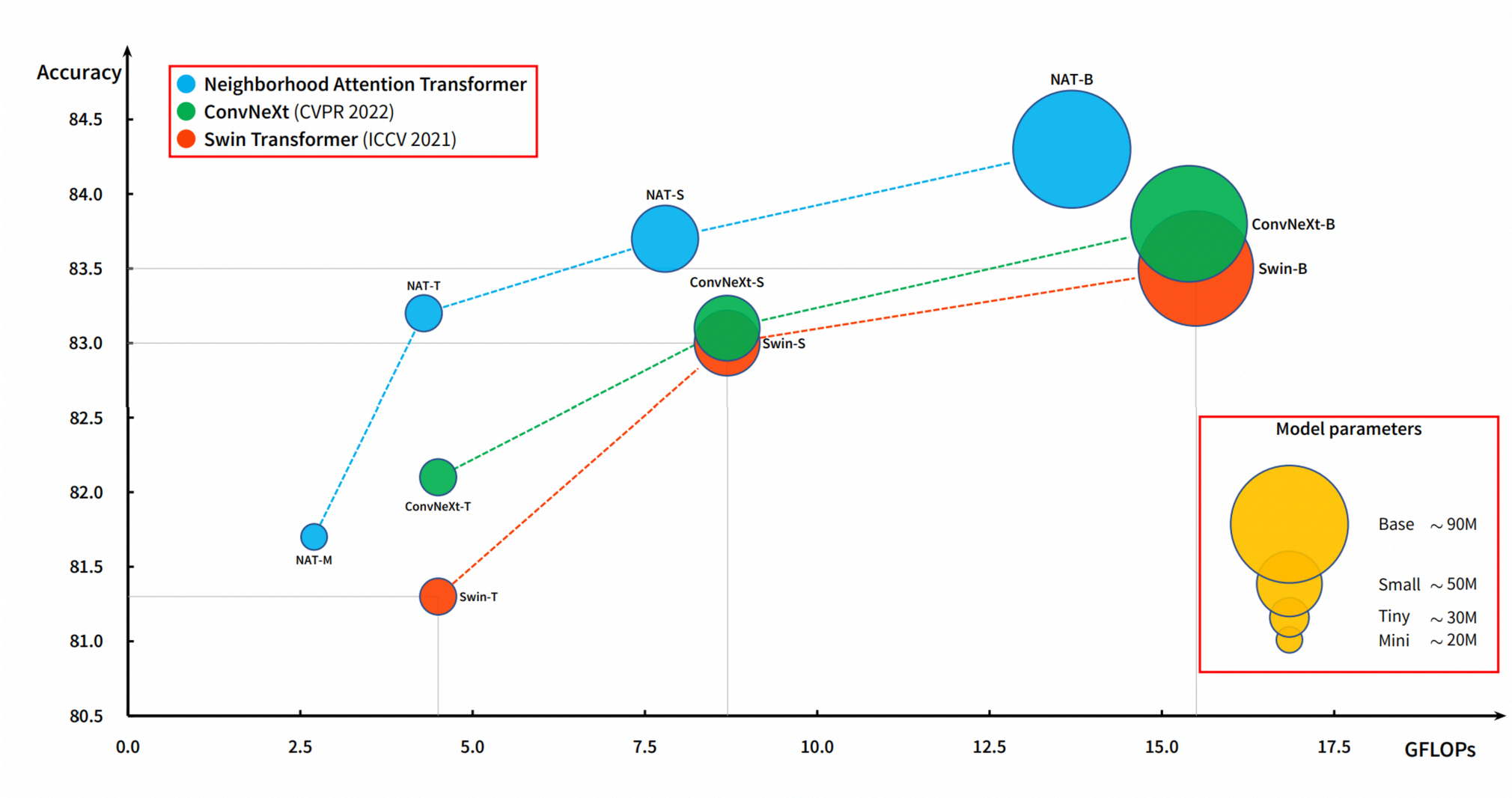


Figure 5. An overview of our model, NAT, with its hierarchical design. The model starts off with a convolutional downsampler, then moves on to 4 sequential levels, each consisting of multiple NAT Blocks, which are transformer-like encoder layers. Each layer is comprised of a multi-headed neighborhood attention (NA), a multi-layered perceptron (MLP), Layer Norm (LN) before each module, and skip connections. Between the levels, feature maps are downsampled to half their spatial size, while their depth is doubled. This allows for easier transfer to downstream tasks through feature pyramids.

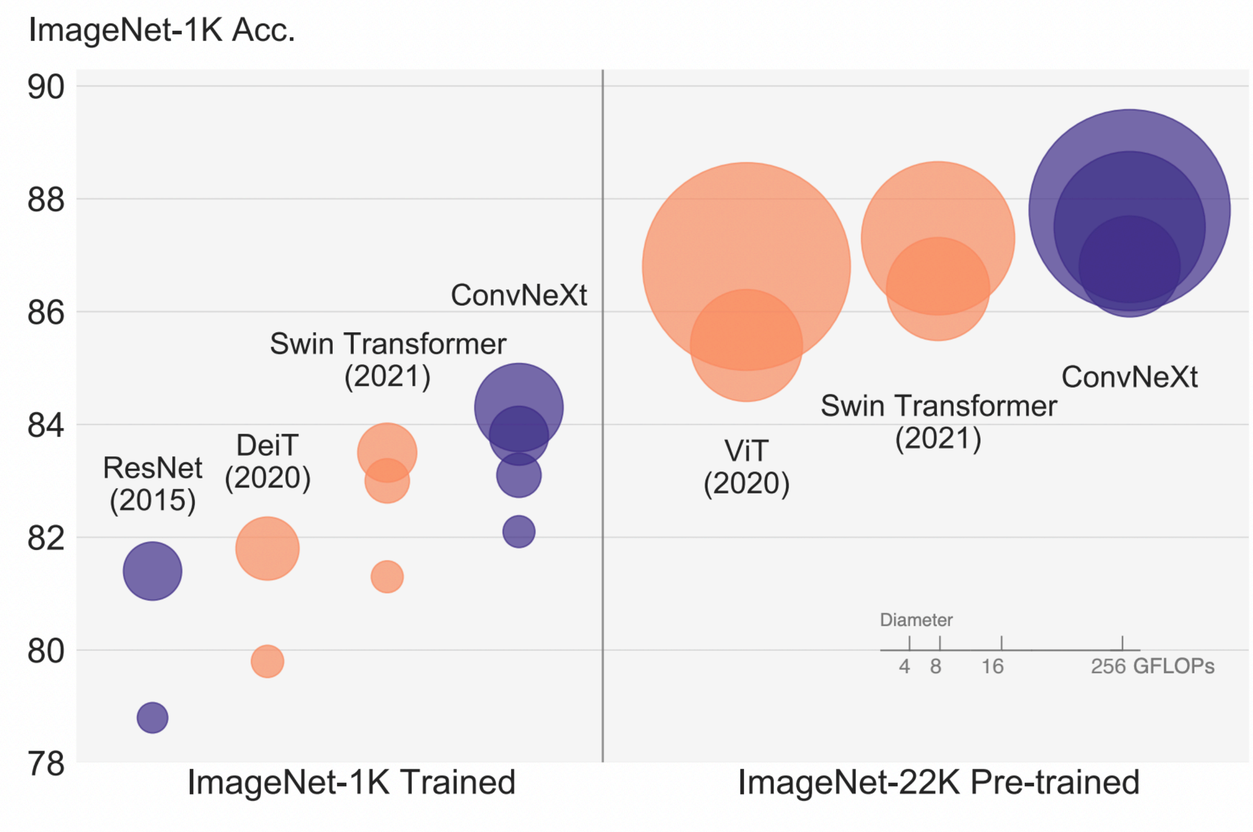
«Neighborhood Attention Transformer» A. Hassani et. al, 2022

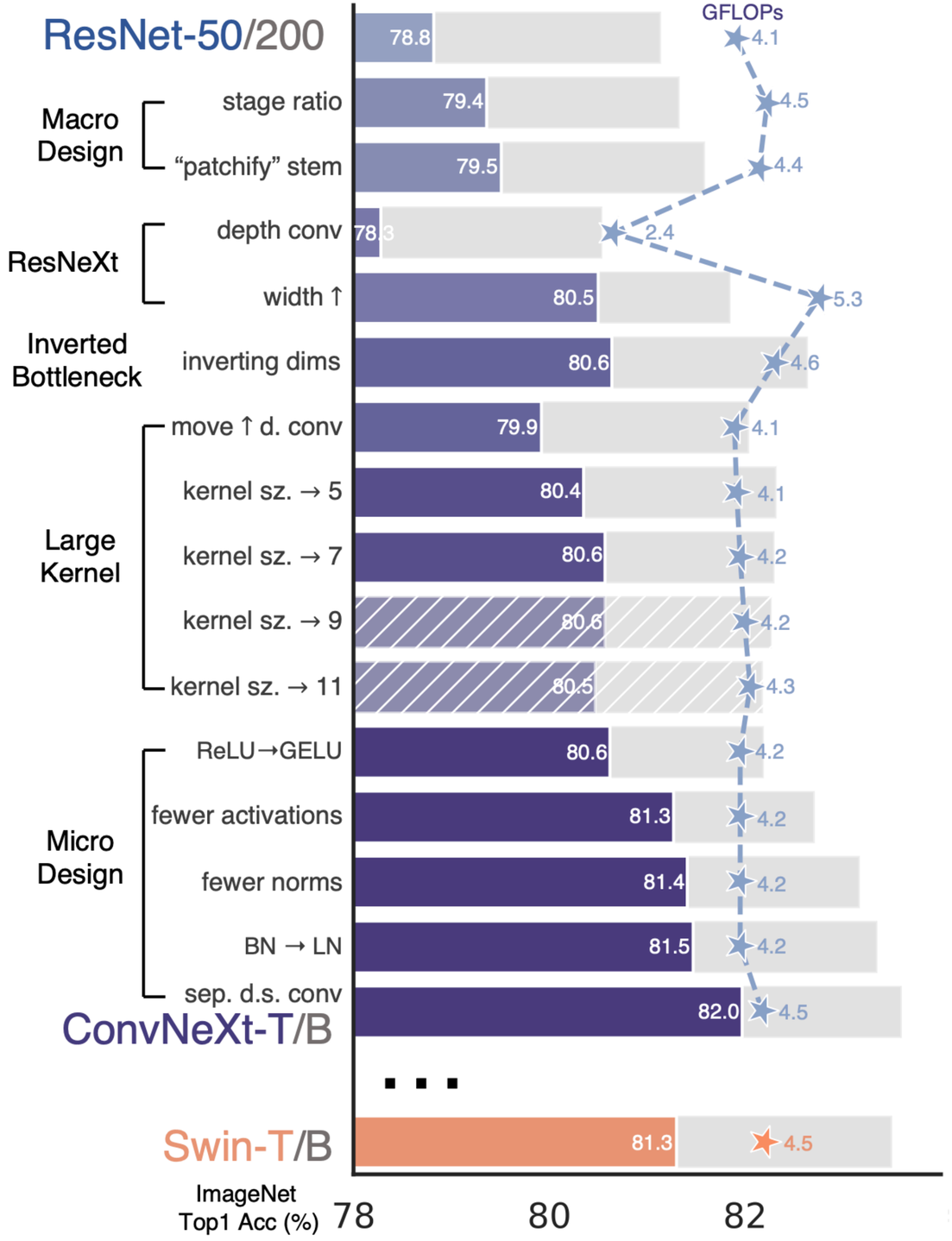
Neighborhood Attention Transformer



«Neighborhood Attention Transformer» A. Hassani et. al, 2022

ConvNext





«A ConvNet for the 2020s» Zhuang Liu et. al, 2022