Local attention mechanism

2023년도 하계 세미나

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

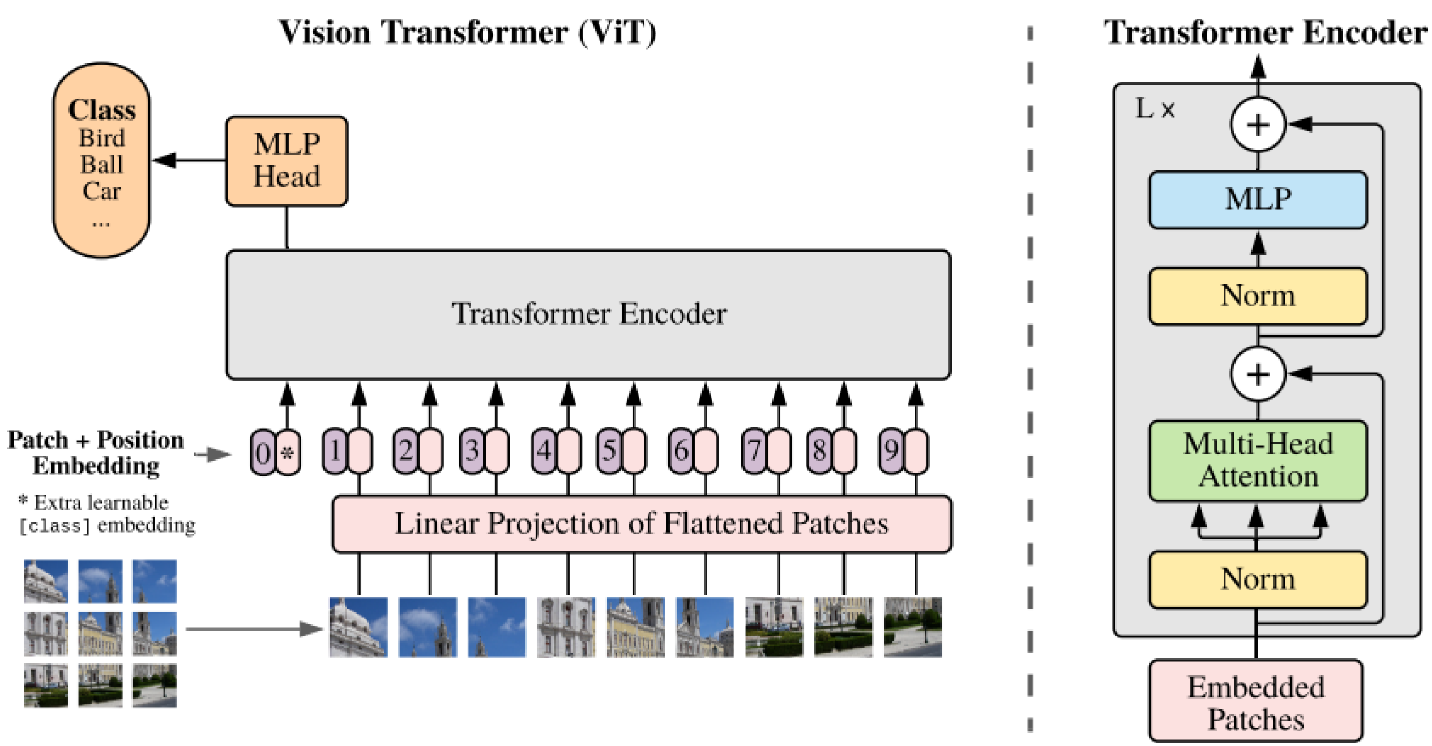
Presented By
문승훈

Outline

- Background
  - Vision transformer
  - Self attention
  - Local attention
  - Shifted window attention
- Neighborhood Attention Transformer
  - CVPR 2023 (60 citations)
- Slide-Transformer: Hierarchical Vision Transformer with Local Self-Attention
  - CVPR 2023
- Conclusion

Background

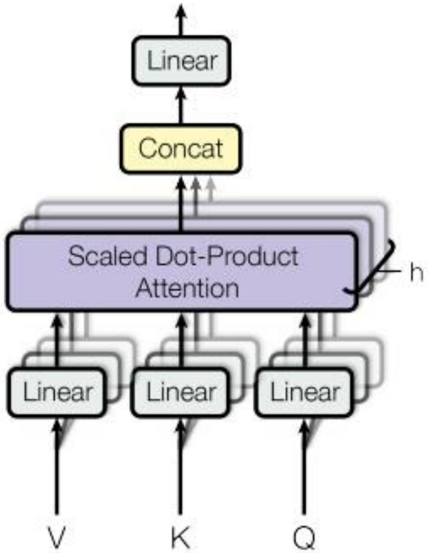
- Vision transformer[1)]
  - Applies attention mechanism instead of CNNs to vision tasks
  - Divides input images into patches, linearize them, and uses attention to capture similarities
  - Utilizes global receptive field



Vision transformer overview

Background

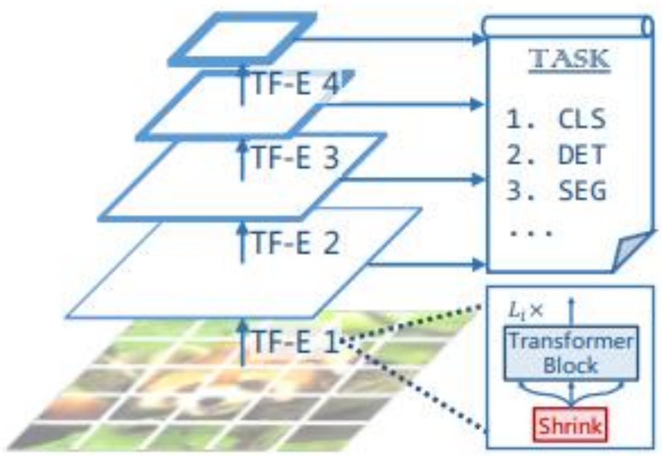
- Self attention(SA)[1)]
  - Embeds query, key, and value via linear projection
    - Query: element for which we want to calculate attention
    - Key: other elements
    - Value: information associated with the elements
  - Calculates attention score using scaled similarity between query and key
  - $Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$
  - Recent studies focused on variances in receptive field size
    - High computation complexity problem



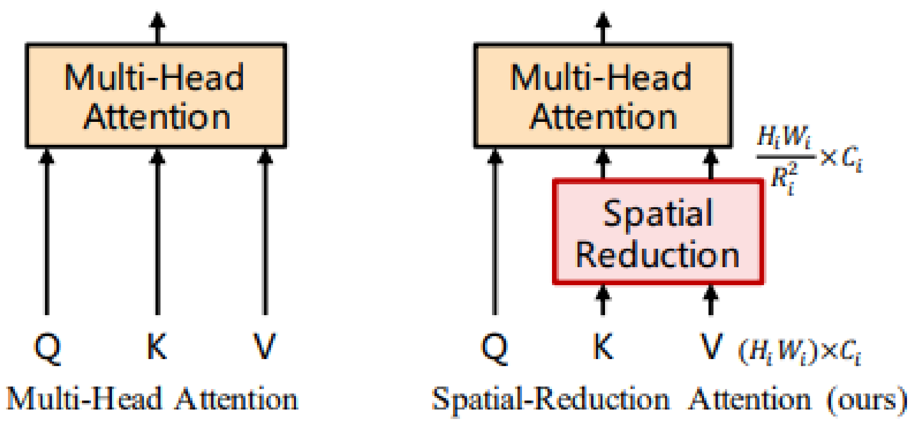
Multi-head SA

Background

- Sparse global attention(SGA)[1)]
  - Select sparse key and value positions from the feature map
  - Information from the selected input portions is than aggregated
    - Efficiency achieved by not considering the respective remaining parts
  - Tends to find optimal trade-off between the size of receptive field and computational cost



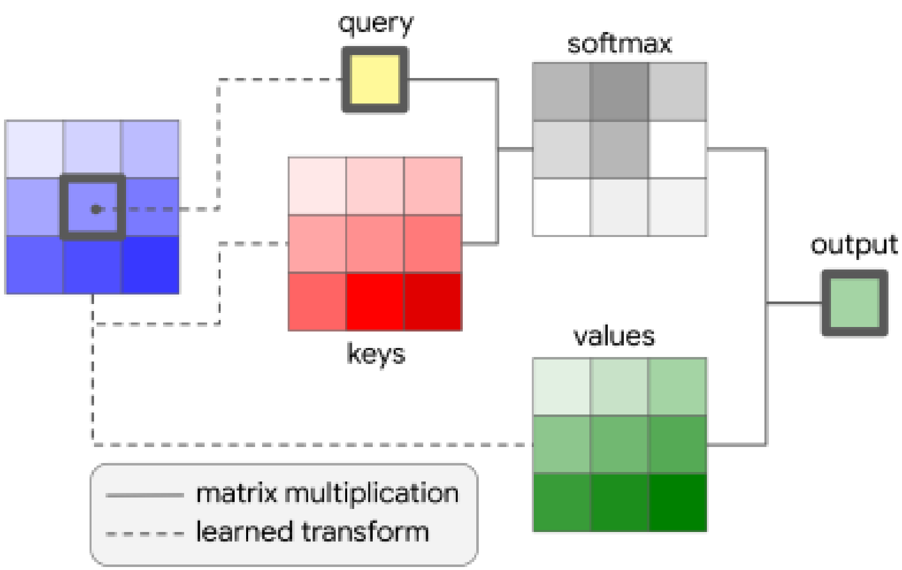
Pyramid vision transformer architecture



Multi-head SA vs Spatial reduction attention

Background

- Local attention(LA)[1)]
  - Replace all spatial convolution operations to self attention layers
  - Constrains receptive field of each query to its neighboring pixels or windows
  - Inherits advantages from CNNs while reducing computational cost
    - Local inductive bias
    - Translation equivariance

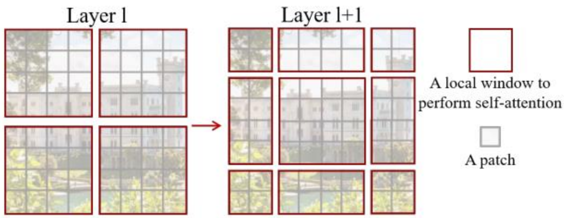


Local attention layer, spatial extent of k = 3

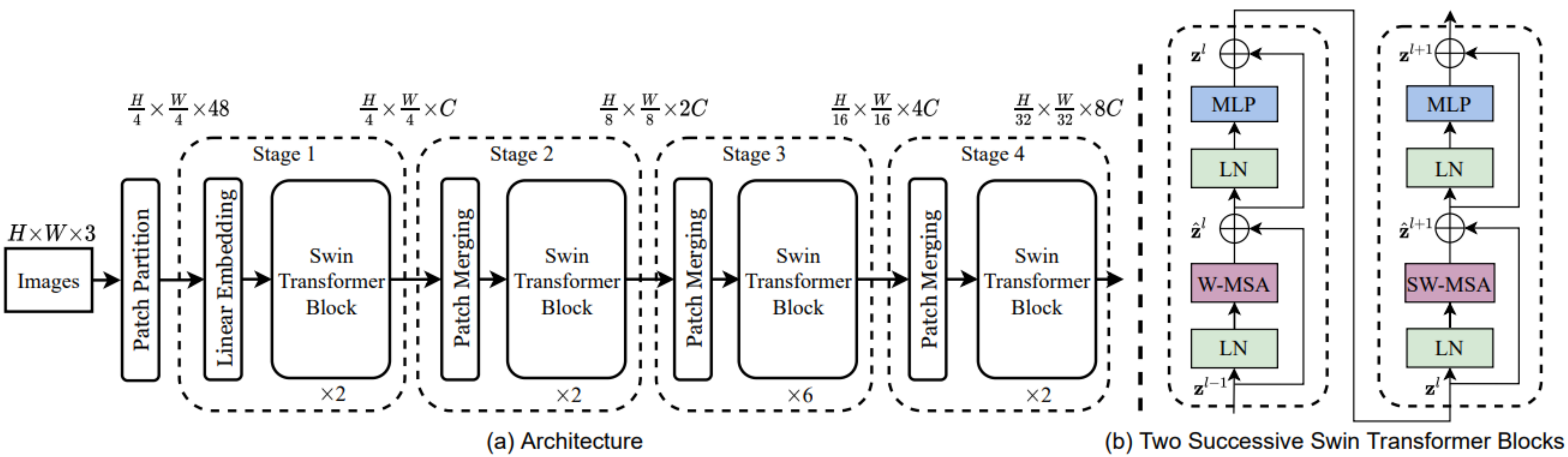
1) LIU, Ze, et al. Swin transformer: Hierarchical vision transformer using shifted windows. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021. p. 10012-10022.

Background

- Shifted window attention(WSA)[1)]
  - Shifts window to utilize sparse global receptive field
    - Avoids situations where features from different windows become isolated
    - Facilitate connections across windows to model extra information
  - Achieves efficient yet effective feature extraction
    - Hierarchical feature extraction
    - Captures both global and local context
    - Linear computational complexity



Shifted window illustration



Swin transformer architecture

Neighborhood Attention Transformer[1)]

- SA involves computational cost overload
- WSA lacks uniform attention span for each span
- Neighborhood attention(NA) localizes SA to each pixel's nearest neighbors
  - Utilizes local inductive biases
  - Maintains translational equivariance
  - Allows uniform receptive field growth without extra operations

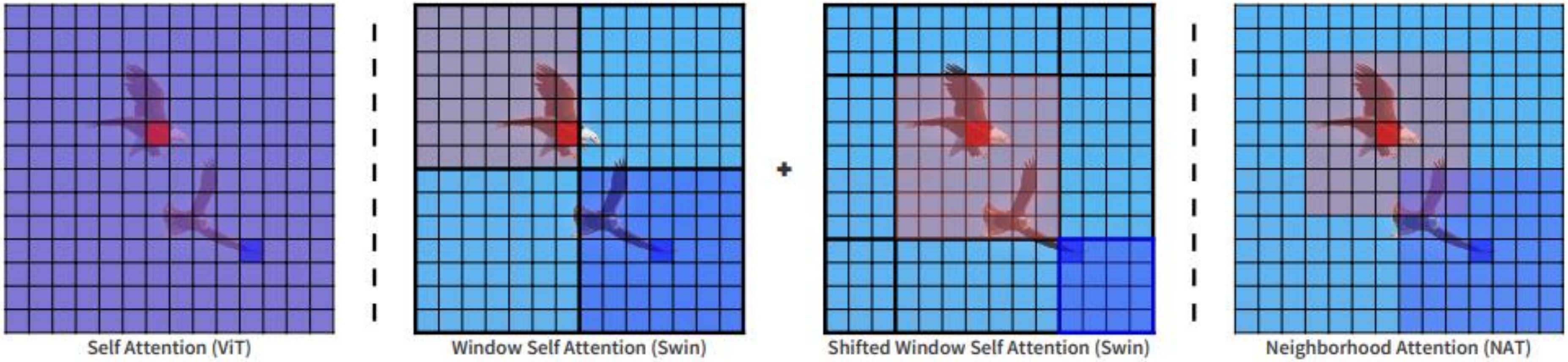


Illustration of attention spans in SA, SWA and NA

1) HASSANI, Ali, et al. Neighborhood attention transformer. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023. p. 6185-6194.

Neighborhood Attention Transformer[1)]

- Neighborhood attention
  - Dot product between the query of the input at the ith position and key of k-nearest neighbors
  - Value defined as ith input's k-nearest neighboring projections
  - Operation repeated for every pixel in the feature map
  - NA approaches SA as window size grows

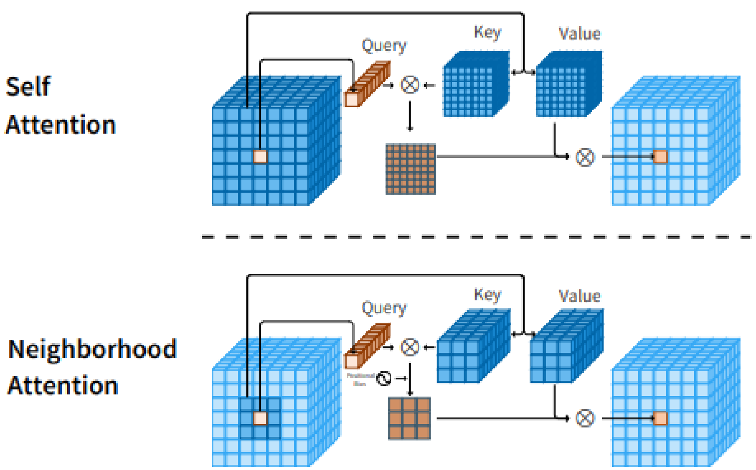


Illustration of query-key-value structure of SA and NA for a single pixel

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

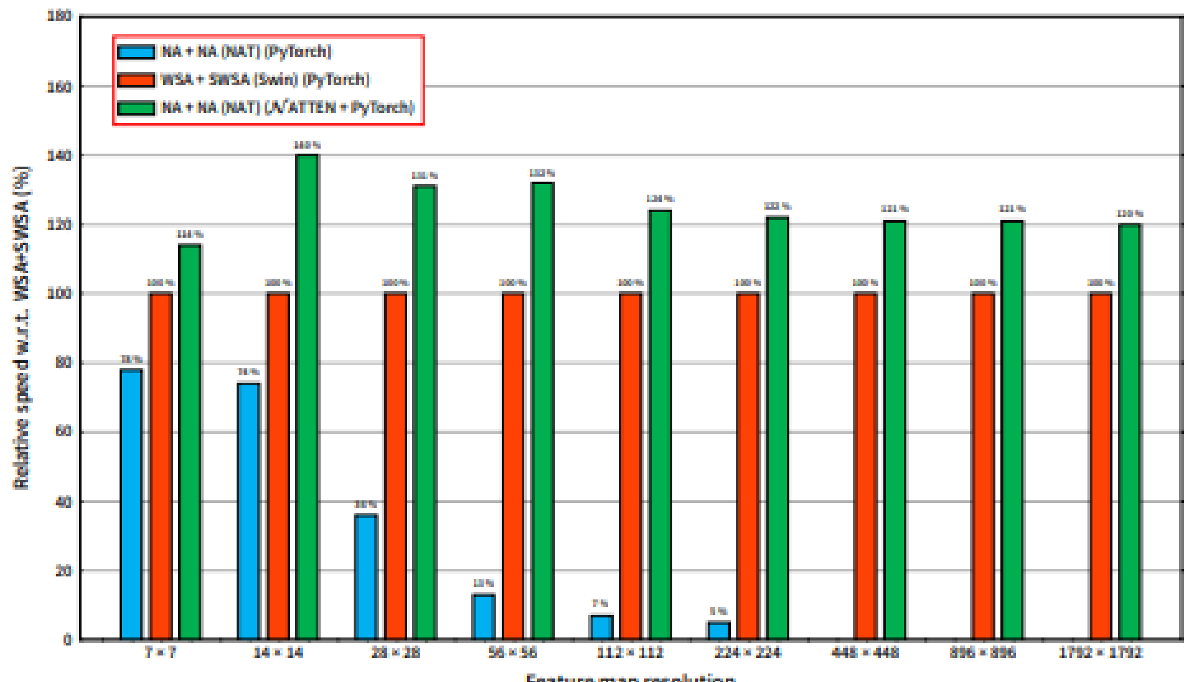
$$\mathrm{NA}_k(i) = softmax\left(\frac{\mathbf{A}_i^k}{\sqrt{d}}\right)\mathbf{V}_i^k$$

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

Neighborhood attention mechanism

Neighborhood Attention Transformer[1)]

- NATTEN: Proposed Python package for efficient pixel-wise operation
  - Pixel-wise SA has not been well-explored due to computational cost
    - Matrix multiplication parallelizable in commonly used computational platforms(cuBLAS, cnDNN, etc)
    - Pixel-wise operation conducted with stacking number of highly inefficient operations
  - NATTEN allows NA-based models to run up to 40% faster than similar Swin counterparts

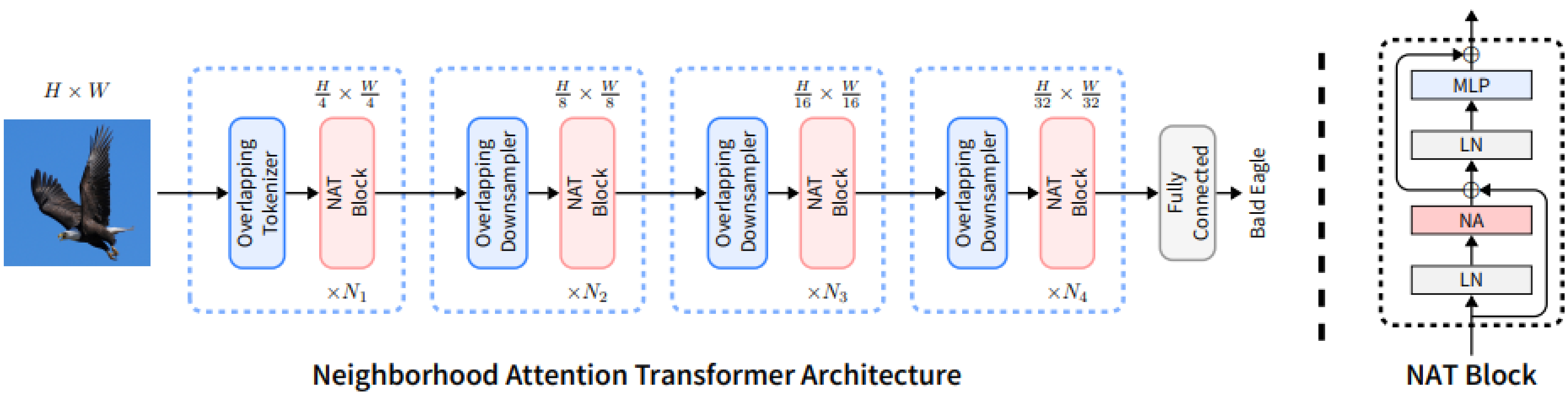


NAT's layer-wise relative speed with respect to Swin transformer

1) HASSANI, Ali, et al. Neighborhood attention transformer. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023. p. 6185-6194.

Neighborhood Attention Transformer[1)]

- Neighborhood attention transformer
  - Embeds inputs using 2 consecutive overlapping 3 x 3 convolutions with 2 x 2 strides
    - Establishes useful inductive biases
    - Resulting in a spatial size 1/4th the size of the input
    - Better trade-off between performance and computational cost
  - Consists of 4 levels followed by respective down-samplers that cut spatial size in half
    - Each level consists of multiple NAT blocks with multi-headed NA



Overview of NAT

Neighborhood Attention Transformer[1)]

- Complexity analysis
  - QKV linear projections → each hwd^2 FLOPs in common → total $3hwd^2$ FLOPs
  - SA has quadratic complexity for both attention weights and output → $2h^2w^2d$ FLOPs
  - WSA divides each QKV into $\frac{h}{k} \times \frac{w}{k}$ windows of shape $k \times k$ → $2hwdk^2$ FLOPs
  - NA case → $2hwdk^2$ FLOPs
    - $\mathbb{A}_i^k$ of size $h \times w \times k^2$ and $\mathbb{V}_i^k$ of size $h \times w \times k^2 \times d$ → $2hwdk^2$ FLOPs
  - Convolutions → hwd^2k^2 FLOPs
  - WSA and NA have identical computational cost

Module	FLOPs	Memory
○ **Self Attn (SA)**	$3hwd^2 + 2h^2w^2d$	$3d^2 + h^2w^2$
○ **Window Self Attn (WSA)**	$3hwd^2 + 2hwdk^2$	$3d^2 + hwk^2$
○ **Neighborhood Attn (NA)**	$3hwd^2 + 2hwdk^2$	$3d^2 + hwk^2$
• **Convolution**	hwd^2k^2	d^2k^2

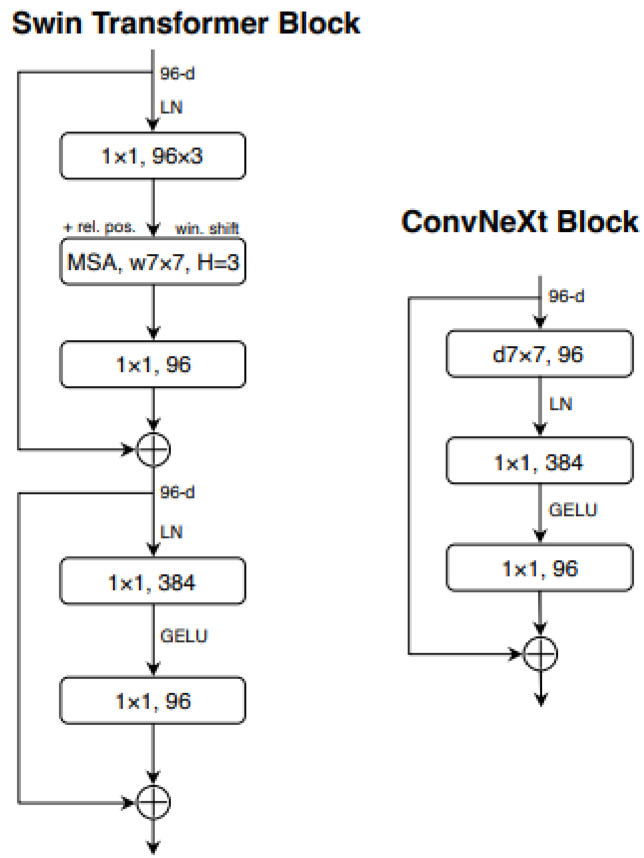
FLOPs and memory usage in different attention patterns and convolutions

Variant	Layers	Dim × Heads	MLP ratio	# of Params	FLOPs
○ **NAT-Mini**	3, 4, 6, 5	32 × 2	3	20 M	2.7 G
○ **NAT-Tiny**	3, 4, 18, 5	32 × 2	3	28 M	4.3 G
○ **NAT-Small**	3, 4, 18, 5	32 × 3	2	51 M	7.8 G
○ **NAT-Base**	3, 4, 18, 5	32 × 4	2	90 M	13.7 G

Comparison of NAT variants

Neighborhood Attention Transformer[1)]

- Baselines
  - Swin transformer[2)]
    - Shifted window attention
  - ConvNeXt[3)]
    - Pure convolutional network
    - Utilizes patch embedding in convolutional stem
    - Divides inputs into multiple paths
    - Adopts DW convolutions
    - Separate down-sampling layers



Block design for Swin transformer and ConvNeXt

Neighborhood Attention Transformer[1)]

- Experiments: classification

Model	# of Params	FLOPs	Thru. (imgs/sec)	Memory (GB)	Top-1 (%)
○ **NAT-M**	20 M	2.7 G	2135	2.4	81.8
○ **Swin-T**	28 M	4.5 G	1730	4.8	81.3
● **ConvNeXt-T**	28 M	4.5 G	2491	3.4	82.1
○ **NAT-T**	28 M	4.3 G	1541	2.5	**83.2**
○ **Swin-S**	50 M	8.7 G	1059	5.0	83.0
● **ConvNeXt-S**	50 M	8.7 G	1549	3.5	83.1
○ **NAT-S**	51 M	7.8 G	1051	3.7	**83.7**
○ **Swin-B**	88 M	15.4 G	776	6.7	83.5
● **ConvNeXt-B**	89 M	15.4 G	1107	4.8	83.8
○ **NAT-B**	90 M	13.7 G	783	5.0	**84.3**

ImageNet-1K classification performance

Neighborhood Attention Transformer[1)]

- Experiments: Object detection and instance segmentation

Backbone	# of Params	FLOPs	Thru. (FPS)	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Mask R-CNN - 3x schedule									
○ NAT-M	40 M	225 G	54.1	46.5	68.1	51.3	41.7	65.2	44.7
○ Swin-T	48 M	267 G	45.1	46.0	68.1	50.3	41.6	65.1	44.9
• ConvNeXt-T	48 M	262 G	52.0	46.2	67.0	50.8	41.7	65.0	44.9
○ NAT-T	48 M	258 G	44.5	47.7	69.0	52.6	42.6	66.1	45.9
○ Swin-S	69 M	359 G	31.7	48.5	70.2	53.5	43.3	67.3	46.6
○ NAT-S	70 M	330 G	34.8	48.4	69.8	53.2	43.2	66.9	46.5
Cascade Mask R-CNN - 3x schedule									
○ NAT-M	77 M	704 G	27.8	50.3	68.9	54.9	43.6	66.4	47.2
○ Swin-T	86 M	745 G	25.1	50.4	69.2	54.7	43.7	66.6	47.3
• ConvNeXt-T	86 M	741 G	27.3	50.4	69.1	54.8	43.7	66.5	47.3
○ NAT-T	85 M	737 G	24.9	51.4	70.0	55.9	44.5	67.6	47.9
○ Swin-S	107 M	838 G	20.3	51.9	70.7	56.3	45.0	68.2	48.8
• ConvNeXt-S	108 M	827 G	23.0	51.9	70.8	56.5	45.0	68.4	49.1
○ NAT-S	108 M	809 G	21.7	52.0	70.4	56.3	44.9	68.1	48.6
○ Swin-B	145 M	982 G	17.3	51.9	70.5	56.4	45.0	68.1	48.9
• ConvNeXt-B	146 M	964 G	19.5	52.7	71.3	57.2	45.6	68.9	49.5
○ NAT-B	147 M	931 G	18.6	52.5	71.1	57.1	45.2	68.6	49.0

COCO object detection and instance segmentation performance

Neighborhood Attention Transformer[1)]

- Experiments: Semantic segmentation

Backbone	# of Params	FLOPs	Thru. (FPS)	mIoU single scale	mIoU multi scale
○ NAT-M	50 M	900 G	24.5	45.1	46.4
○ Swin-T	60 M	946 G	21.3	44.5	45.8
• ConvNeXt-T	60 M	939 G	23.3	46.0	46.7
○ NAT-T	58 M	934 G	21.4	47.1	48.4
○ Swin-S	81 M	1040 G	17.0	47.6	49.5
• ConvNeXt-S	82 M	1027 G	19.1	48.7	49.6
○ NAT-S	82 M	1010 G	17.9	48.0	49.5
○ Swin-B	121 M	1188 G	14.6	48.1	49.7
• ConvNeXt-B	122 M	1170 G	16.4	49.1	49.9
○ NAT-B	123 M	1137 G	15.6	48.5	49.7

ADE20K semantic segmentation performance

Neighborhood Attention Transformer[1)]

- Experiments: ablation study

Attention	ImageNet Top-1	MSCOCO AP^B	MSCOCO AP^m	ADE20K mIoU	# of Params	FLOPs	Thru. (imgs/sec)	Memory (GB)
○ SWSA	81.3%	46.0	41.6	45.8	28.28 M	4.51 G	1730	4.8
○ SASA	81.6%	46.0	41.4	46.4	28.27 M	4.51 G	2021	4.0
○ NA	81.8%	46.2	41.5	46.4	28.28 M	4.51 G	2021	4.0

Performance comparison of different attention mechanisms

Attention	Down--sampler	# of Layers	# of Heads	MLP Ratio	Top-1 (%)	# of Params	FLOPs (G)	Thru. (imgs/sec)	Memory (GB)
○ SWSA	Patch	2, 2, 6, 2	3	4	81.29	28.3 M	4.5	1730	4.8
○ SWSA	Conv	2, 2, 6, 2	3	4	81.78	30.3 M	4.9	1692	4.8
○ SWSA	Conv	3, 4, 18, 5	2	3	82.72	27.9 M	4.3	1320	3.0
○ SASA	Conv	3, 4, 18, 5	2	3	82.54	27.9 M	4.3	1541	2.5
○ NA	Conv	3, 4, 18, 5	2	3	83.20	27.9 M	4.3	1541	2.5

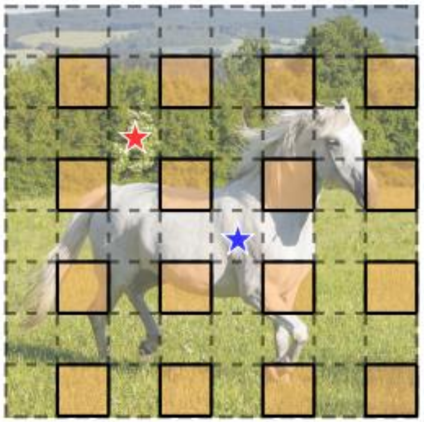
Ablation study on NAT with Swin-T as the baseline

Kernel size	ImageNet Top-1 (%)	ImageNet Thru.	MSCOCO AP^b	MSCOCO AP^m	MSCOCO Thru.	ADE20K mIoU	ADE20K Thru.
3×3	81.4	2015 imgs/sec	46.1	41.4	46.8 fps	46.0	23.6 fps
5×5	81.6	1810 imgs/sec	46.8	42.0	45.5 fps	46.3	22.9 fps
7×7	83.2	1537 imgs/sec	47.7	42.6	44.5 fps	48.4	21.4 fps
9×9	83.1	1253 imgs/sec	48.5	43.3	39.4 fps	48.1	20.2 fps

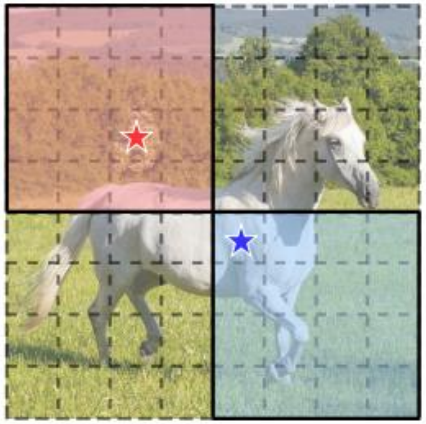
NAT-Tiny performance with different kernel sizes

Slide-Transformer[1)]

- Existing attention mechanisms to restrict receptive field size show distinct limitations
  - Sparse global attention
    - Inferior in capturing local features
    - Susceptible to key and value positions
      - ∴ informative features in other regions may be discarded
  - Window attention
    - Hinders cross-window communication
    - Involves extra designs such as window shifts
      - ∴ Sets restrictions on the model architecture



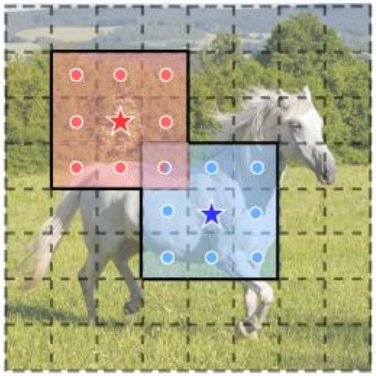
Sparse global attention



Window attention

Slide-Transformer[1)]

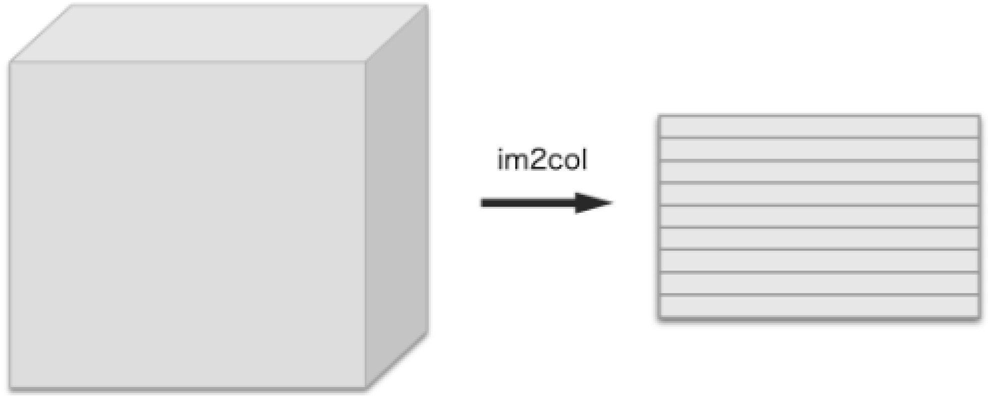
- Local attention mechanism was proposed
  - Constrain receptive field of each query in its own neighboring pixels
  - Takes advantages from both convolution and attention
    - Local inductive bias from query-centric attention pattern
    - Translation equivariance like traditional convolution
    - Sets least restrictions on the model architecture design
- Limitations remaining
  - Huge increase in inference stage due to inefficient Im2Col function
  - Relies on CUDA kernels which restricts applicability on edge devices
- Importance on possessing both high efficiency and high generalizability



Local attention

Slide-Transformer[1)]

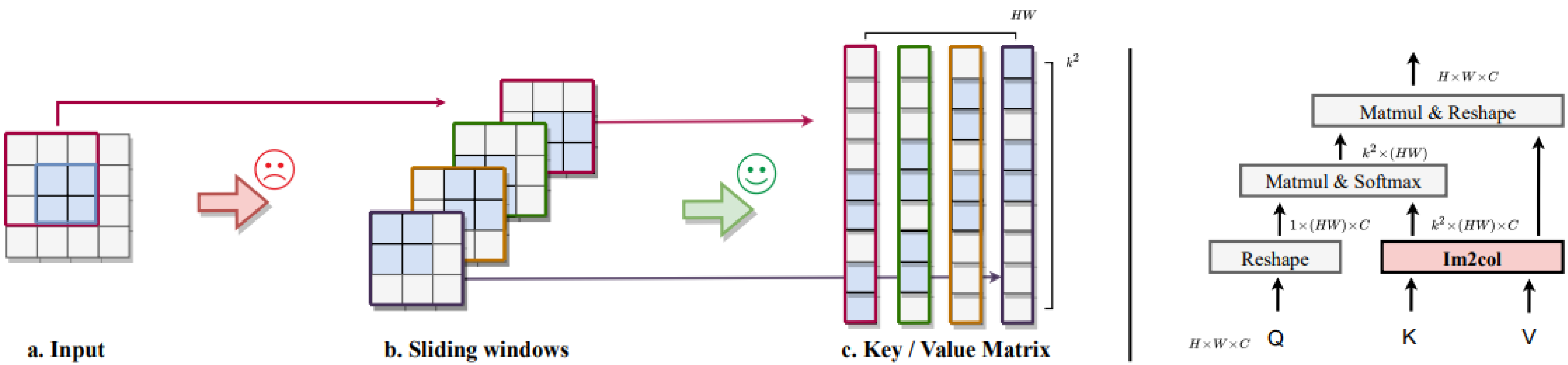
- Im2Col function
  - Transforms multidimensional data into matrix type
    - Facilitate matrix operations
  - Enables efficient weighted multidimensional calculations
- Im2Col function is inefficient in terms of local attention
  - Generates the key and value matrix from column-based view
  - Each column represents local region



Im2Col function

Slide-Transformer[1)]

- Difficulties in implementing local attention
  - Different receptive region for each query in the feature map
  - Im2Col function is adopted to sample keys and values for all respective queries
    - Local windows flattening into columns conducted independently by slicing the feature map
    - Disrupts data locality
    - Leads to huge time consumption

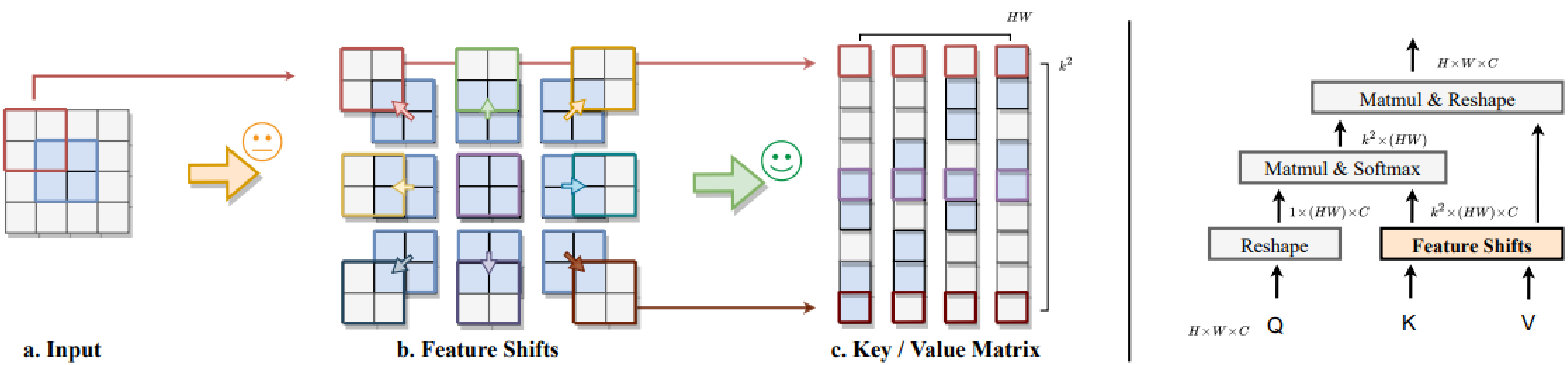


Local attention implementation with Im2Col function

1) PAN, Xuran, et al. Slide-Transformer: Hierarchical Vision Transformer with Local Self-Attention. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023. p. 2082-2091.

Slide-Transformer[1)]

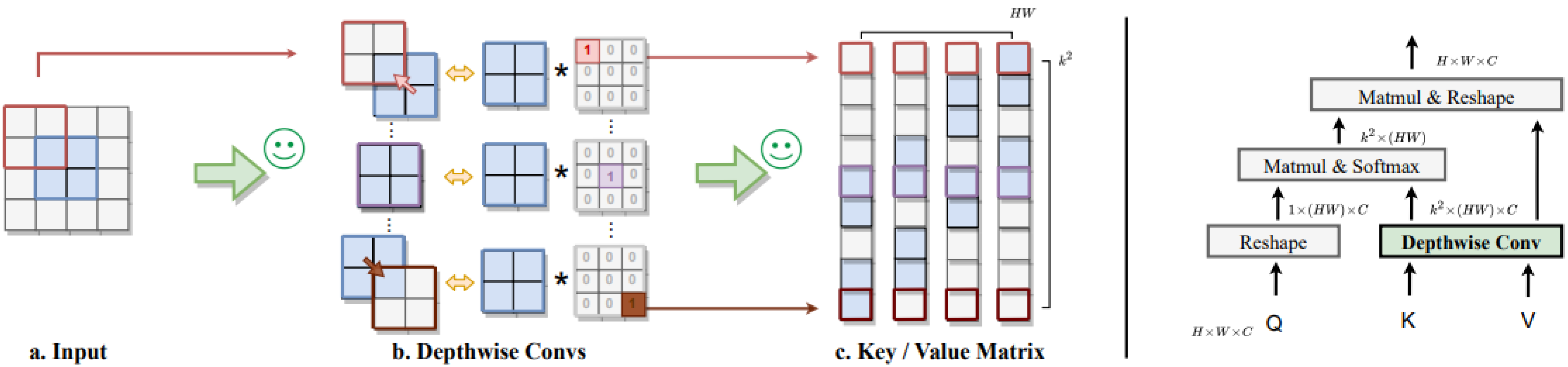
- New perspective on Im2Col
  - Original column-based view
    - Each column corresponds to local window centered at particular query
    - Receptive windows of all queries are sampled and placed in order
  - New row-based view
    - Each row corresponds to shifting input towards certain direction
    - Flatten shifted features into rows and concatenate
    - Can recover the same dimensional output



Local attention implementation with feature shifts

Slide-Transformer[1)]

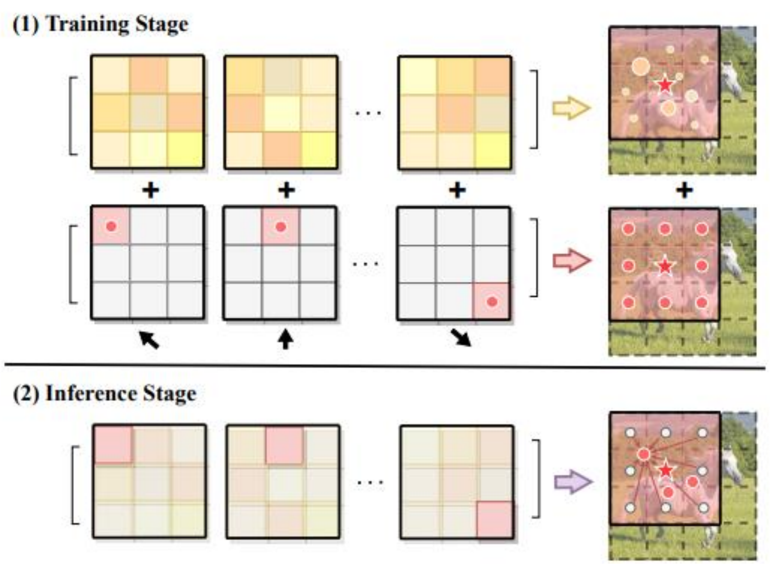
- Shift as depth-wise convolution
  - Simply shifting still involves inefficient slicing operations
  - Apply depth-wise convolution with designed kernels as a replacement for feature shift
    - Depth-wise convolutions can be boiled down to single-group convolution
    - Outputs equivalent to previous feature shifts
    - Avoids inefficient slicing operation
    - Optimized implementation of convolution operations on many edge devices



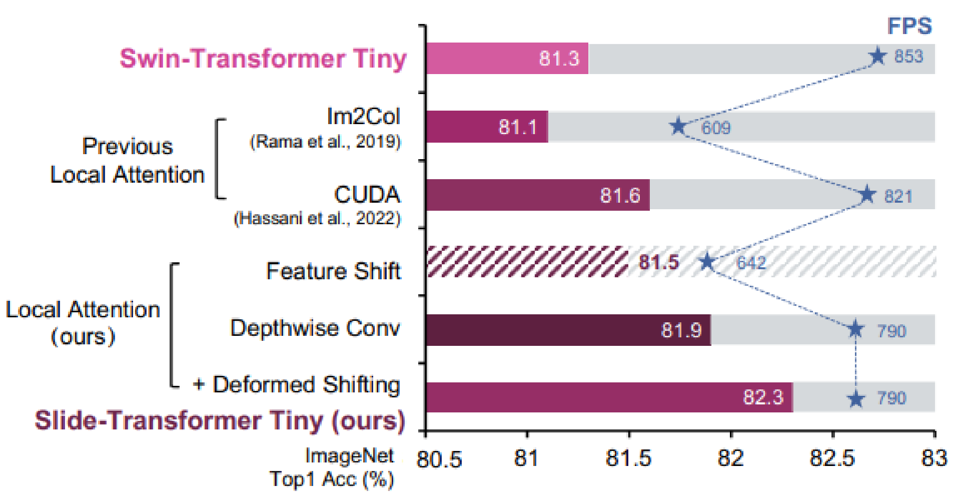
Local attention implementation with shift as depth-wise convolution

Slide-Transformer[1)]

- Deformed shifting module
  - Designed kernel weights still constrain keys and values to the fixed neighboring positions
    - Makes it hard to capture diverse features
  - Deformed shifting module to handle the limitation
    - Further enhance flexibility of local attention
    - Parallel convolution path of learnable kernel parameters with random initialization
    - Use re-parameterization (merging) to transform two parallel paths into single convolution



Deformed shifting module with re-parameterization

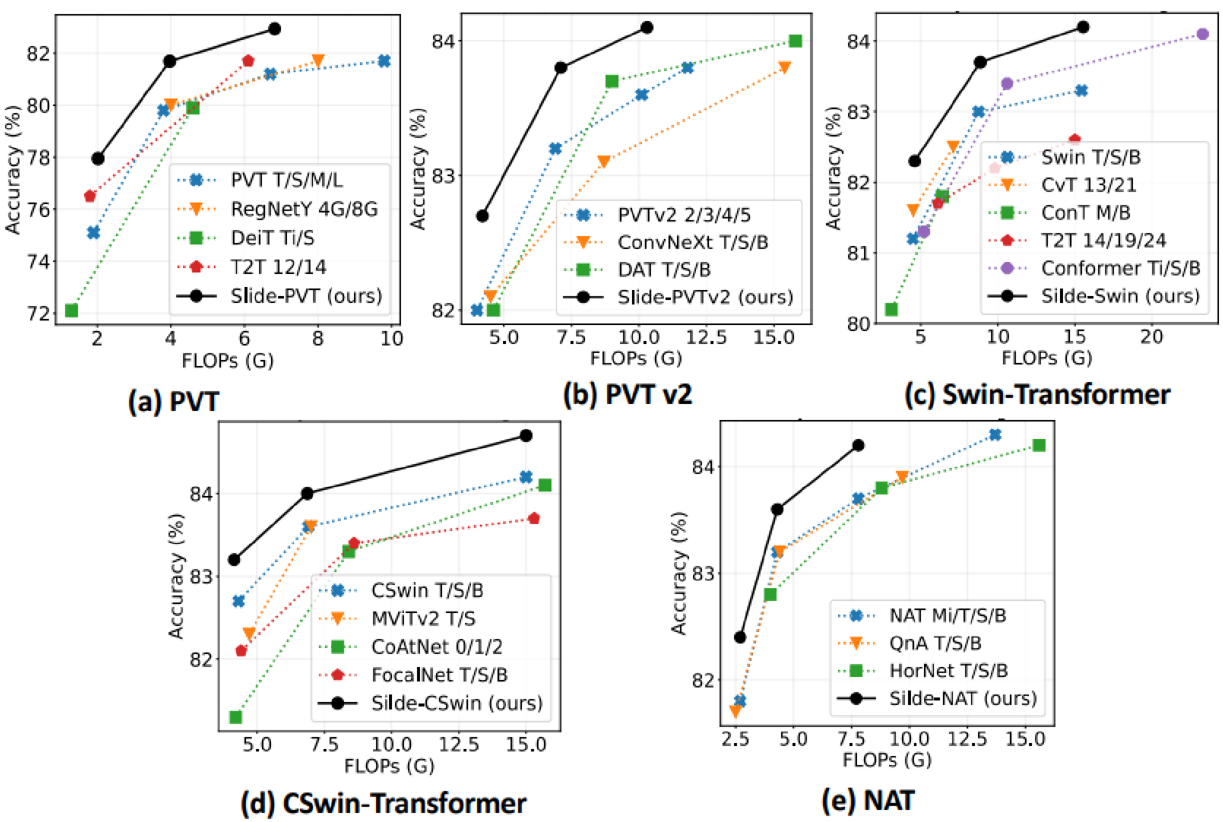


Performance and inference speed comparison on local attention implementations

1) PAN, Xuran, et al. Slide-Transformer: Hierarchical Vision Transformer with Local Self-Attention. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023. p. 2082-2091.

Slide-Transformer[1)]

- Experiments: classification



Method	Params	Flops	Top-1
PVT-T [30]	13.2M	1.9G	75.1
Slide-PVT-T	12.2M	2.0G	**78.0** (+2.9)
PVT-S	24.5M	3.8G	79.8
Slide-PVT-S	22.7M	4.0G	**81.7** (+1.9)
PVTv2-B1 [31]	13.1M	2.1G	78.7
Slide-PVTv2-B1	13.0M	2.2G	**79.5** (+0.7)
PVTv2-B2	25.4M	4.0G	82.0
Slide-PVTv2-B2	22.8M	4.2G	**82.7** (+0.7)
Swin-T [20]	29M	4.5G	81.3
Slide-Swin-T	29M	4.6G	**82.3** (+1.0)
Swin-S	50M	8.7G	83.0
Slide-Swin-S	51M	8.9G	**83.7** (+0.7)
Swin-B	88M	15.4G	83.5
Slide-Swin-B	89M	15.5G	**84.2** (+0.7)
CSwin-S [8]	35M	6.9G	83.6
Slide-CSwin-S	35M	6.9G	**84.0** (+0.4)
CSwin-B	78M	15.0G	84.2
Slide-CSwin-B	78M	15.0G	**84.7** (+0.5)
NAT-T [8]	28M	4.3G	83.2
Slide-NAT-T	28M	4.3G	**83.6** (+0.4)
NAT-S	51M	7.8G	83.7
Slide-NAT-S	51M	7.8G	**84.3** (+0.6)

Comparisons of FLOPs and paramters against accuracy on ImageNet-1K

Slide-Transformer[1)]

- Experiments: object detection and instance segmentation

(a) Mask R-CNN Object Detection & Instance Segmentation on COCO

Method	FLOPs	#Param	Schedule	AP^b	AP^b_{50}	AP^b_{75}	AP^b_s	AP^b_m	AP^b_l	AP^m	AP^m_{50}	AP^m_{75}	AP^m_s	AP^m_m	AP^m_l
PVT-T	240G	33M	1x	36.7	59.2	39.3	21.6	39.2	49.0	35.1	56.7	37.3	19.5	37.4	48.5
Slide-PVT-T	219G	32M	1x	40.4	63.4	43.8	25.3	42.8	53.0	38.1	60.4	41.0	20.0	40.1	55.2
PVT-S	305G	44M	1x	40.4	62.9	43.8	22.9	43.0	55.4	37.8	60.1	40.3	20.4	40.3	53.6
Slide-PVT-S	269G	42M	1x	42.8	65.9	46.7	26.6	45.5	57.3	40.1	63.1	43.1	20.3	42.4	59.0
PVT-M	392G	64M	1x	42.0	64.4	45.6	24.4	44.9	57.9	39.0	61.6	42.1	21.3	42.0	55.2
Slide-PVT-M	357G	62M	1x	44.4	66.9	48.6	28.9	47.0	59.4	40.8	63.9	43.8	25.0	43.5	55.9
PVTv2-B1	244G	34M	1x	41.8	64.3	45.9	26.4	44.9	54.3	38.8	61.2	41.6	20.2	41.3	56.1
Slide-PVTv2-B1	222G	33M	1x	42.6	65.3	46.8	27.4	45.6	55.7	39.7	62.6	42.6	24.1	42.9	53.7
PVTv2-B2	309G	45M	1x	45.3	67.1	49.6	28.8	48.4	59.5	41.2	64.2	44.4	22.0	43.7	59.4
Slide-PVTv2-B2	274G	43M	1x	46.0	68.2	50.3	28.8	49.4	61.0	41.9	65.1	45.4	24.6	45.2	57.2
Swin-T	267G	48M	3x	46.0	68.1	50.3	31.2	49.2	60.1	41.6	65.1	44.9	25.9	45.1	56.9
Slide-Swin-T	268G	48M	3x	46.8	69.0	51.6	31.7	50.4	60.1	42.3	66.0	45.8	23.5	45.8	60.8

(b) Cascade Mask R-CNN Object Detection & Instance Segmentation on COCO

Method	FLOPs	#Param	Schedule	AP^b	AP^b_{50}	AP^b_{75}	AP^b_s	AP^b_m	AP^b_l	AP^m	AP^m_{50}	AP^m_{75}	AP^m_s	AP^m_m	AP^m_l
Swin-T	745G	86M	3x	50.4	69.2	54.7	33.8	54.1	65.2	43.7	66.6	47.3	27.3	47.5	59.0
Slide-Swin-T	747G	86M	3x	51.1	69.8	55.4	35.2	54.4	65.8	44.3	67.4	48.0	28.0	48.0	59.2
Swin-S	838G	107M	3x	51.9	70.7	56.3	35.2	55.7	67.7	45.0	68.2	48.8	28.8	48.7	60.6
Slide-Swin-S	838G	107M	3x	52.5	71.3	57.2	35.6	56.1	68.0	45.4	68.9	49.6	29.1	49.2	60.6
Swin-B	981G	145M	3x	51.9	70.5	56.4	35.4	55.2	67.4	45.0	68.1	48.9	28.9	48.3	60.4
Slide-Swin-B	983G	145M	3x	52.7	71.2	57.2	37.0	56.1	68.0	45.5	68.8	49.6	30.1	48.8	60.9

COCO object detection performance

1) PAN, Xuran, et al. Slide-Transformer: Hierarchical Vision Transformer with Local Self-Attention. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023. p. 2082-2091.

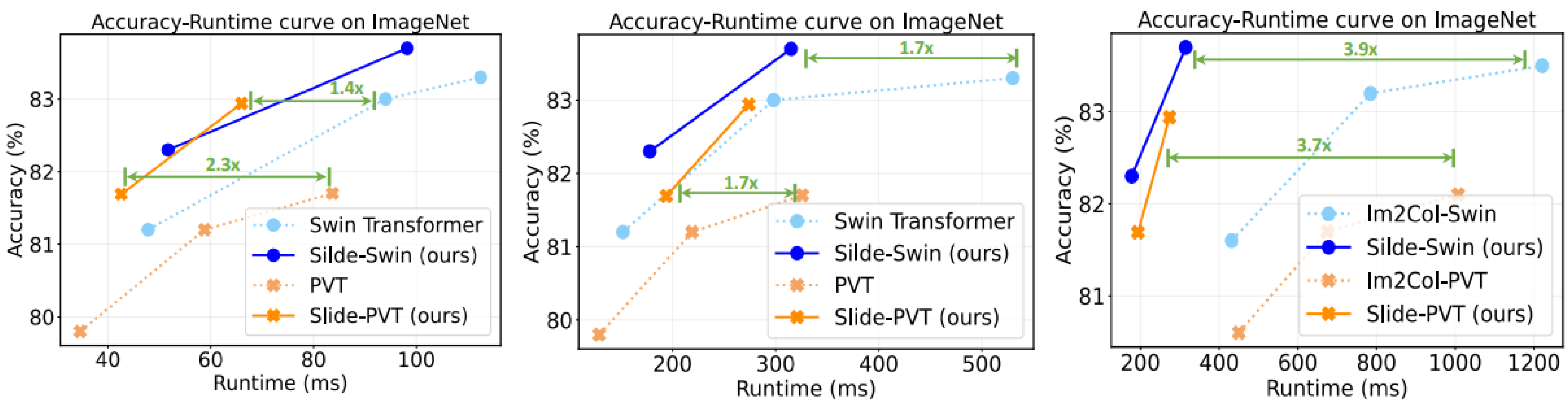
Slide-Transformer[1)]

- Experiments: semantic segmentation

Semantic Segmentation on ADE20K					
Backbone	Method	FLOPs	#Params	mIoU	mAcc
PVT-T	S-FPN	158G	17M	36.57	46.72
Slide-PVT-T	S-FPN	136G	16M	**38.43**	50.05
PVT-S	S-FPN	225G	28M	41.95	53.02
Slide-PVT-S	S-FPN	188G	26M	**42.47**	54.00
Swin-T	UperNet	945G	60M	44.51	55.61
Slide-Swin-T	UperNet	946G	60M	**45.67**	57.13
Swin-S	UperNet	1038G	81M	47.64	58.78
Slide-Swin-S	UperNet	1038G	81M	**48.46**	60.18

ADE20K semantic segmentation performance

- Experiments: Runtime comparison



(a) Metal Performance Shader (*v.s.* Baselines) (b) iPhone 12 (*v.s.* Baselines) (c) iPhone 12 (*v.s.* Im2Col)

Runtime comparison results

Slide-Transformer[1)]

- Experiments: ablation study

(a) Comparison on Swin-T Setting				
Local Attention	FLOPs	#Param	Acc.	FPS
SASA [25]	4.5G	29M	81.6	644
SAN [37]	4.5G	29M	81.4	670
NAT [11]	4.5G	29M	81.8	821
Ours	4.6G	30M	**82.3**	790
(b) Comparison on NAT-Mini Setting				
Local Attention	FLOPs	#Param	Acc.	FPS
SASA [25]	2.7G	20M	81.2	791
SAN [37]	2.7G	20M	81.1	815
NAT [11]	2.7G	20M	81.8	1045
Ours	2.7G	20M	**82.4**	998

Comparison of different local attention modules on different model structures

Stages w/ Slide Attention				FLOPs	#Param	Acc.	Diff.
Stage1	Stage2	Stage3	Stage4				
✓				4.5G	29M	81.8	-0.5
✓	✓			4.6G	29M	**82.3**	**Ours**
✓	✓	✓		4.6G	30M	82.2	-0.1
✓	✓	✓	✓	4.7G	30M	81.3	-1.0
Swin-T [20]				4.5G	29M	81.3	-1.0

Ablation study on applying slide attention on different stages

서강대학교 SOGANG UNIVERSITY

VDS LAB

Conclusion

- Novel attention mechanisms to maximize efficiency by constraining receptive field size
  - Sparse global attention
  - Window attention
  - Local attention
    - Neighborhood Attention Transformer
      - Localizes SA to each pixel's nearest neighbors
      - New Python package NATTEN
    - Slide Transformer
      - New row-based perspective on Im2Col
      - Deformed shifting module

Thank you