Non-Global Attention Mechanisms In Vision Transformers

Paper Reading by Zhiying Lu

2023.04.27

智能多媒体内容计算实验室
Intelligent Multimedia Content Computing Lab

Neighborhood Attention Transformer

Ali Hassani[1], Steven Walton[1], Jiachen Li[1], Shen Li[3], Humphrey Shi[1,2]
[1]SHI Lab @ U of Oregon & UIUC, [2]Picsart AI Research (PAIR), [3]Meta/Facebook AI

CVPR2023

Slide-Transformer: Hierarchical Vision Transformer with Local Self-Attention

Xuran Pan* Tianzhu Ye* Zhuofan Xia Shiji Song Gao Huang†
Department of Automation, BNRist, Tsinghua University

CVPR2023

BiFormer: Vision Transformer with Bi-Level Routing Attention

Lei Zhu[1] Xinjiang Wang[2] Zhanghan Ke[1] Wayne Zhang[2] Rynson Lau[1†]
[1] City University of Hong Kong [2] SenseTime Research
{lzhu68-c,zhanghake2-c}@my.cityu.edu.hk, {wangxinjiang,wayne.zhang}@sensetime.com
Rynson.Lau@cityu.edu.hk

CVPR2023

- 作者介绍
- 研究背景
- NAT
- Slide-Transformer
- BiFormer
- 后话

作者介绍

Ali Hassani

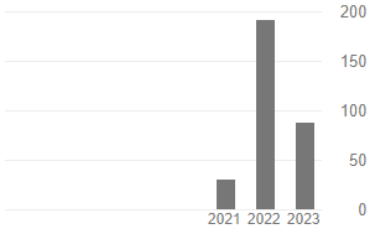
FOLLOW

University of Oregon
Verified email at uoregon.edu - Homepage
Attention-based models Representation learning

TITLE	CITED BY	YEAR
Escaping the Big Data Paradigm with Compact Transformers A Hassani, S Walton, N Shah, A Abuduweili, J Li, H Shi arXiv preprint arXiv:2104.05704	194	2021
Convmlp: Hierarchical convolutional mlps for vision J Li, A Hassani, S Walton, H Shi arXiv preprint arXiv:2109.04454	36	2021
Neighborhood Attention Transformer A Hassani, S Walton, J Li, S Li, H Shi IEEE/CVF Conference on Computer Vision and Pattern Recognition	35	2023

Cited by

	All	Since 2018
Citations	311	311
h-index	6	6
i10-index	4	4



Humphrey Shi

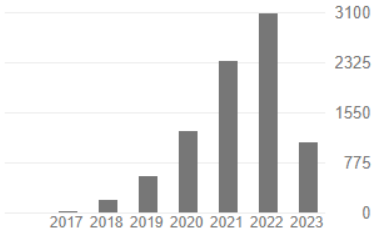
FOLLOW

Oregon & UIUC; Picsart AI Research (PAIR)
Verified email at illinois.edu - Homepage
Computer Vision AI Machine Learning

TITLE	CITED BY	YEAR
CCNet: Criss-Cross Attention for Semantic Segmentation Z Huang, X Wang, Y Wei, L Huang, H Shi, W Liu, TS Huang IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)	1916	2020
Ntire 2017 challenge on single image super-resolution: Methods and results R Timofte, E Agustsson, LV Gool, MH Yang, L Zhang, B Lim, S Son, H Kim, ... IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops	1274	2017
Revisiting Dilated Convolution: A Simple Approach for Weakly- and Semi- Supervised Semantic Segmentation Y Wei, H Xiao, H Shi, Z Jie, J Feng, TS Huang IEEE Conference on Computer Vision and Pattern Recognition (CVPR)	512	2018
HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation B Cheng, B Xiao, J Wang, H Shi, TS Huang, L Zhang IEEE Conference on Computer Vision and Pattern Recognition (CVPR)	445	2020

Cited by

	All	Since 2018
Citations	8635	8600
h-index	36	36
i10-index	57	57



Public access VIEW ALL

0 articles	15 articles
not available	available

作者介绍

Xuran Pan (潘旭冉)

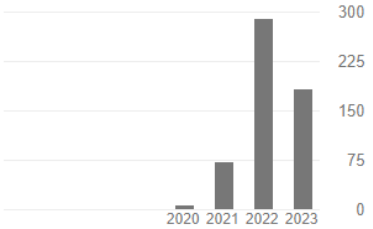
FOLLOW

Tsinghua University
Verified email at mails.tsinghua.edu.cn - Homepage
Computer Vision 3D Vision Neural Architecture Design

TITLE	CITED BY	YEAR
3D Object Detection with Pointformer X Pan, Z Xia, S Song, LE Li, G Huang IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2021	184	2021
Implicit Semantic Data Augmentation for Deep Networks Y Wang*, X Pan*, S Song, H Zhang, C Wu, G Huang Advances in Neural Information Processing Systems (NeurIPS) 2019, 12635-12644	115	2019
Vision Transformer with Deformable Attention Z Xia*, X Pan*, S Song, LE Li, G Huang IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2022	93	2022

Cited by

	All	Since 2018
Citations	555	555
h-index	7	7
i10-index	6	6



Public access VIEW ALL

Gao Huang （黄高）

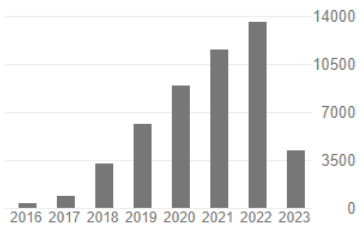
FOLLOW

Associate Professor, Tsinghua University
Verified email at tsinghua.edu.cn - Homepage
Machine Learning Computer Vision Dynamic Neural Networks Efficient Deep Learning

TITLE	CITED BY	YEAR
Convolutional Networks with Dense Connectivity G Huang, Z Liu, G Pleiss, L Van Der Maaten, K Weinberger IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)	34677 *	2019
Learning efficient convolutional networks through network slimming Z Liu, J Li, Z Shen, G Huang, S Yan, C Zhang IEEE International Conference on Computer Vision (ICCV), 2736-2744	2096	2017
Deep Networks with Stochastic Depth G Huang, Y Sun, Z Liu, D Sedra, KQ Weinberger European Conference on Computer Vision (ECCV), 646-661	2084	2016
Trends in extreme learning machines: A review G Huang, GB Huang, S Song, K You Neural Networks 61, 32-48	1670	2015
Rethinking the value of network pruning	1224	2018

Cited by

	All	Since 2018
Citations	49635	47818
h-index	42	40
i10-index	71	68



Public access VIEW ALL

9 articles 41 articles
not available available

作者介绍

Rynson W.H. Lau

关注

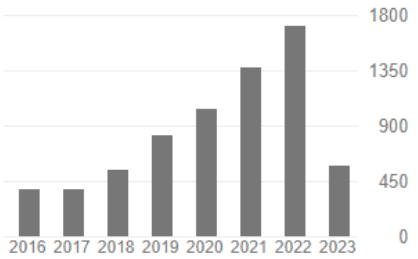
City University of Hong Kong
在 cityu.edu.hk 的电子邮件经过验证 - 首页

Computer Graphics Computer Vision Virtual Reality

标题	引用次数	年份
CREST: Convolutional Residual Learning for Visual Tracking Y Song, C Ma, L Gong, J Zhang, RWH Lau, MH Yang IEEE International Conference on Computer Vision (ICCV), 2574-2583	575	2017
VITAL: Visual Tracking via Adversarial Learning Y Song, C Ma, X Wu, L Gong, L Bao, W Zuo, C Shen, RWH Lau, MH Yang IEEE Computer Vision and Pattern Recognition (CVPR), 8990-8999	522	2018
Visual Tracking via Locality Sensitive Histograms S He, Q Yang, RWH Lau, J Wang, MH Yang IEEE Computer Vision and Pattern Recognition (CVPR), 2427-2434	390	2013
Spatial Attentive Single-Image Deraining with a High Quality Real Rain Dataset T Wang, X Yang, K Xu, S Chen, Q Zhang, RWH Lau IEEE Computer Vision and Pattern Recognition (CVPR), 12262-12271	361	2019

引用次数 查看全部

	总计	2018 年至今
引用	9603	6084
h 指数	48	34
i10 指数	158	91



开放获取的出版物数量 查看全部

11 篇文章 72 篇文章

- 作者介绍
- 研究背景
- NAT
- Slide-Transformer
- BiFormer
- 后话

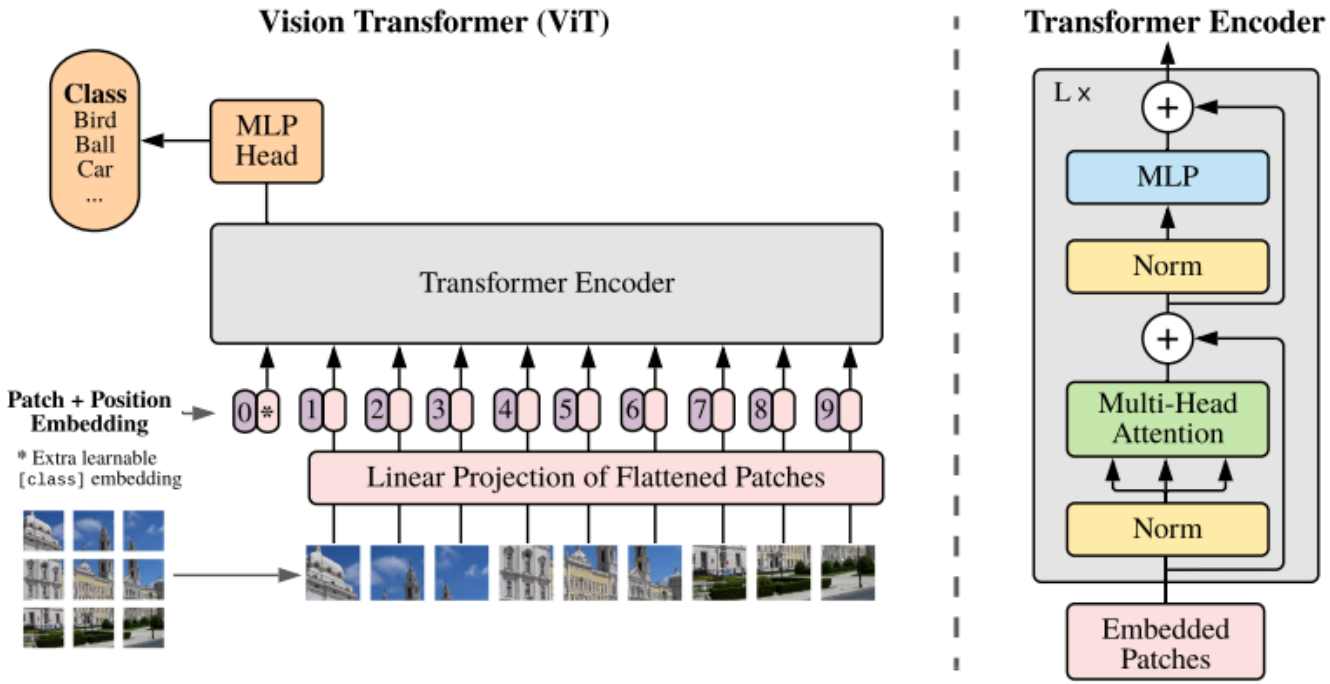
研究背景

研究背景由两个部分组成

1. Vision Transformer
2. Non-Global Attention

Vision Transformers

Vision Transformer[3]



1、图像分patch，每个patch经过embedding后变为一个token，224=14*16

2、patch token和额外的cls token组成sequence，输入到级联的多层Encoder提取特征

3、利用 Multi-head Self-Attention捕捉token之间的注意力，利用MLP进行特征表征

Vision Transformers

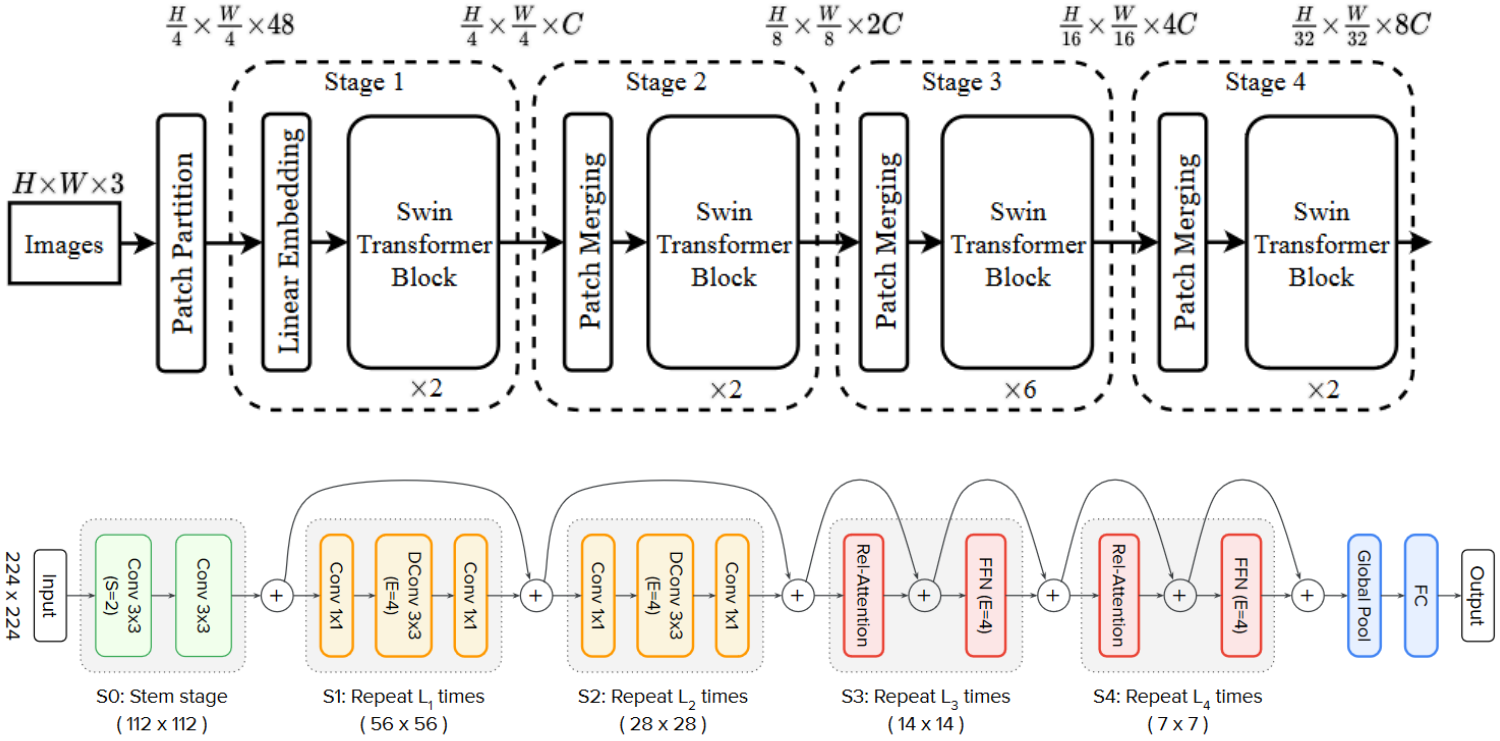


Figure 4: Overview of the proposed CoAtNet.

SOTA的Transformer架构一般采用金字塔型，逐级下采样并增加维度

Self-Attention

Important Components

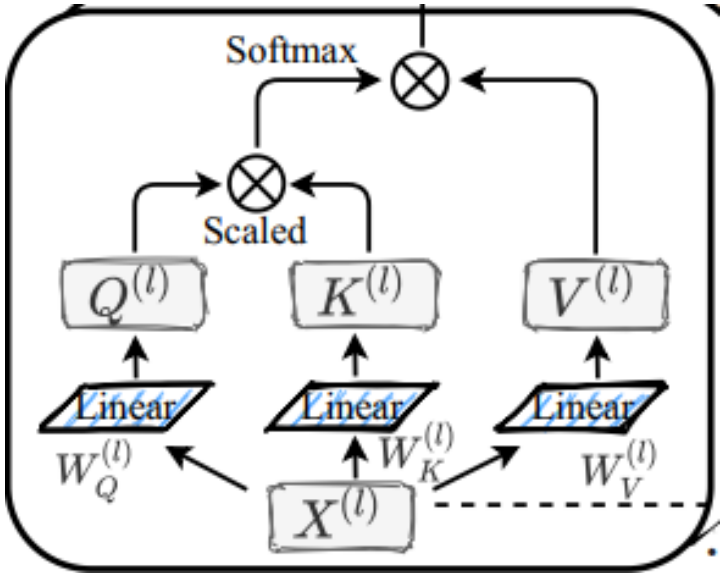
(1) Feature shape [B, N, C]

(2) 当进行global范围的self-attn时，每一个head形成的attn map的大小为[B, N, N]

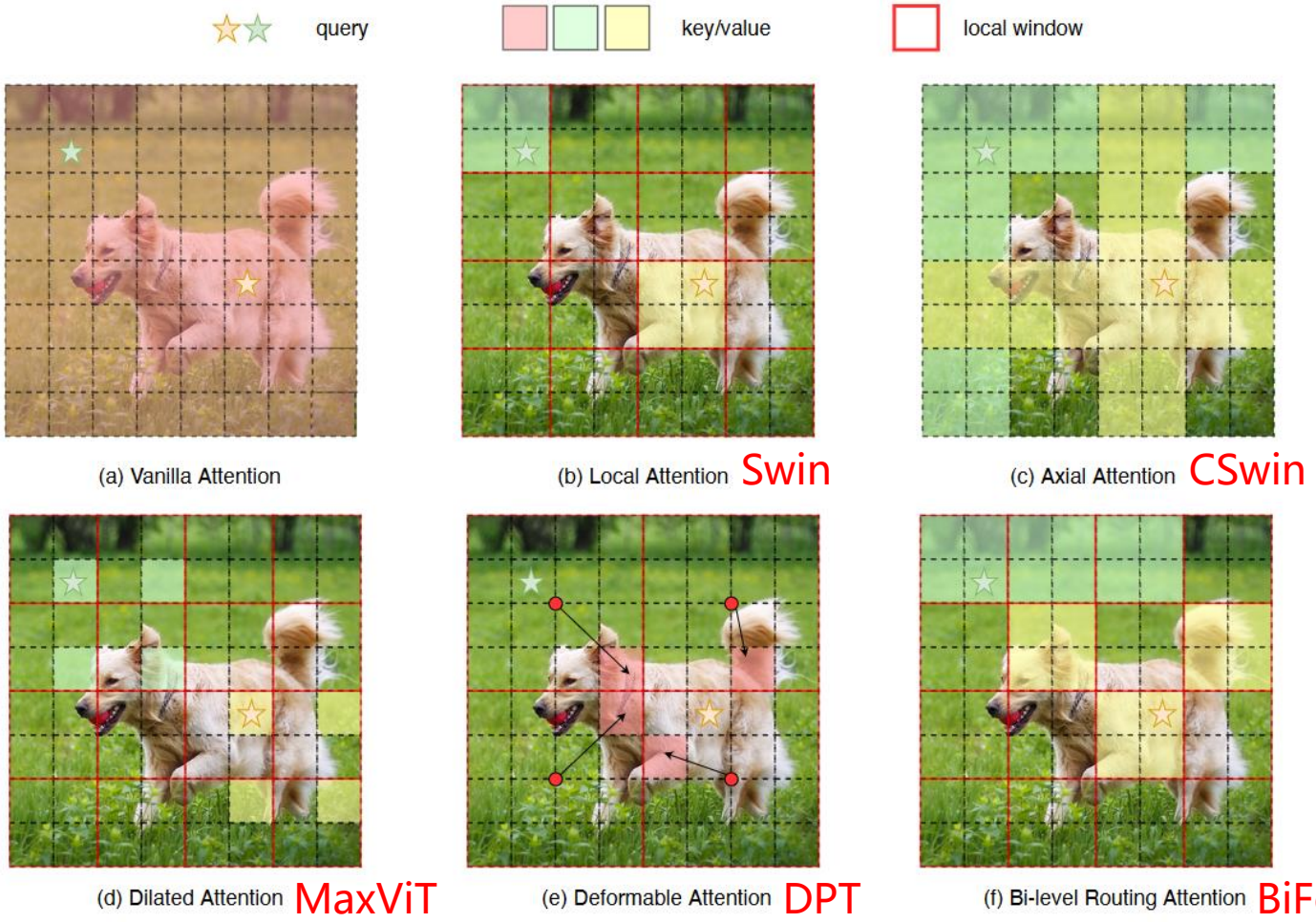
(3) 当输入尺寸变大时，或者在网络的浅层，直接使用vanilla self-attn会带来巨大的显存和计算负担，因此一些操作考虑将self-attn的范围进行限制

Self-Attention

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}$$

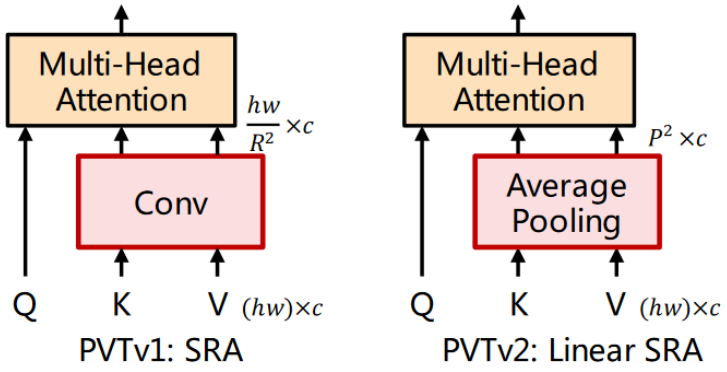


Non-Global Attention



Non-Global Attention

MSG-Transformer



PVT

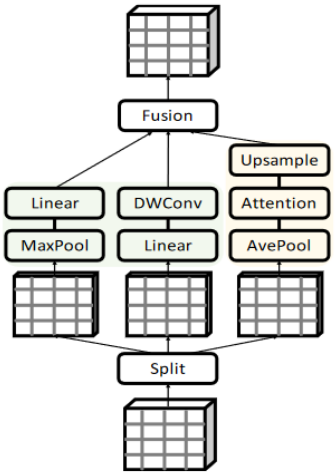


Figure 3: **The details of Inception mixer.**

Inception Transformer

Non-Global Attention

Non-Global Attention存在的问题

(1) 很多操作划分是hand-crafted的，不够灵活

(2) 使用window-attention时，为了保持全局性，还需要shift-window

(3) 在进行窗口划分和移动时，涉及到大量的reshape操作，严重影响推理速度

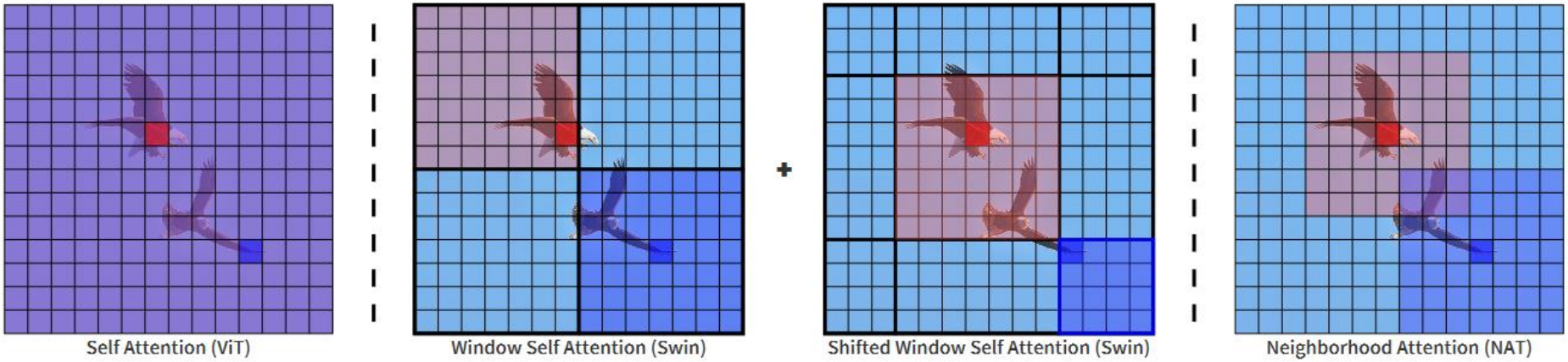
(4) 使用压缩维度的操作时，可能会模糊每一个patch自己的特征，造成一定的损失

- 作者介绍
- 研究背景
- NAT
- Slide-Transformer
- BiFormer
- 后话

NAT

Neighborhood Attention Transformer

Ali Hassani[1], Steven Walton[1], Jiachen Li[1], Shen Li[3], Humphrey Shi[1,2]
[1]SHI Lab @ U of Oregon & UIUC, [2]Picsart AI Research (PAIR), [3]Meta/Facebook AI



NAT

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

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

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

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

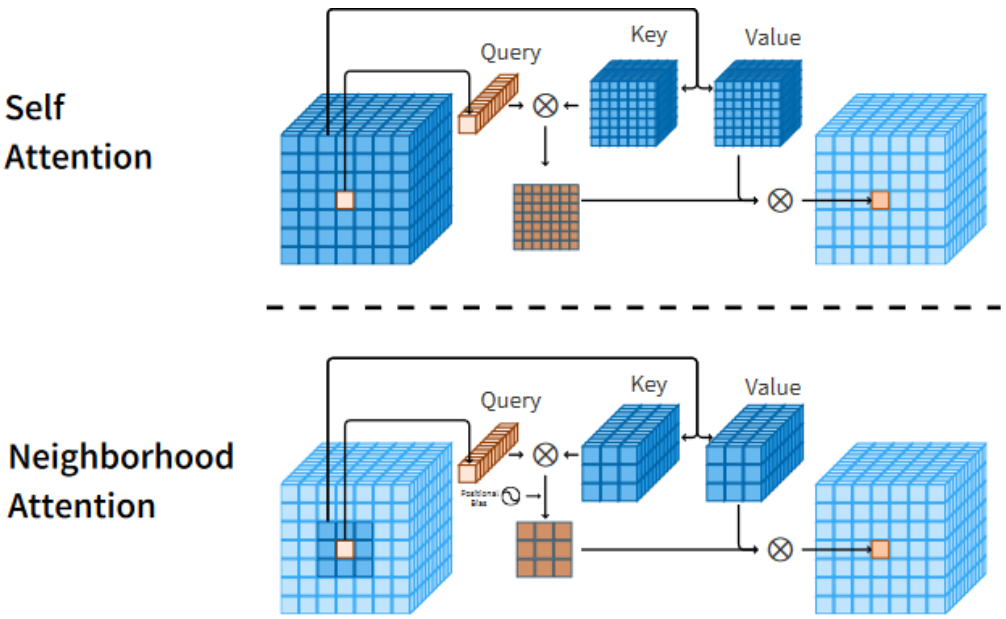
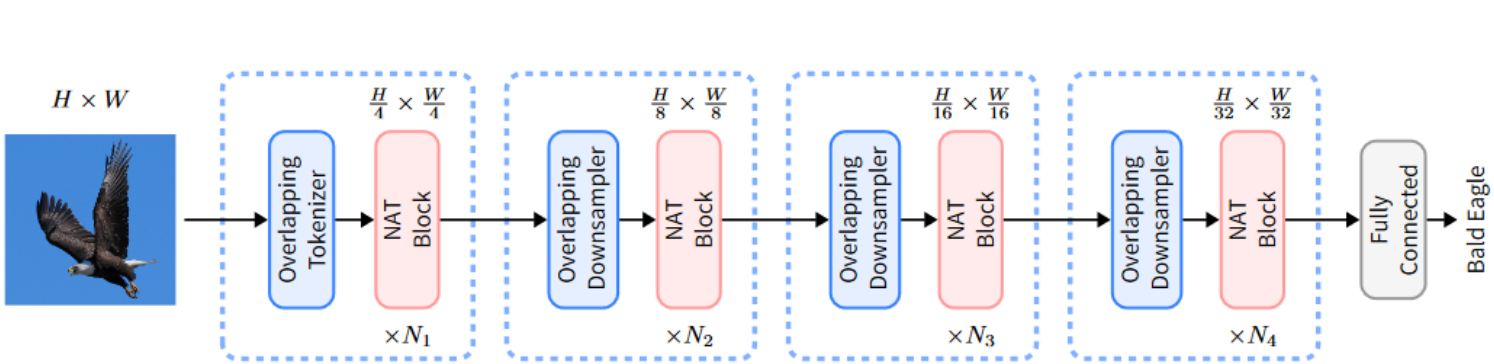


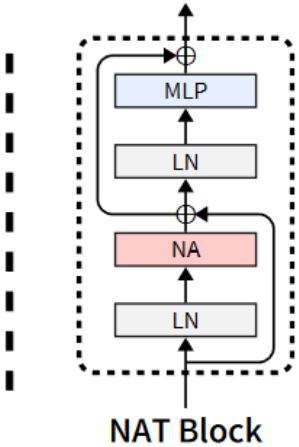
Figure 2. Illustration of the query-key-value structure of Neighborhood Attention (NA) vs Self Attention (SA) for a single pixel. SA allows each pixel to attend to every other pixel, whereas NA localizes attention for each pixel to a neighborhood around itself. Therefore, each pixel's attention span is usually different from the next.

NAT

```
self.gamma1 = nn.Parameter(layer_scale_init_value * torch.ones((dim)), requires_grad=True)
self.gamma2 = nn.Parameter(layer_scale_init_value * torch.ones((dim)), requires_grad=True)
if self.use_layer_scale:
    x = x + self.drop_path(self.gamma1 * self.attn(self.norm1(x))) # (N, H, W, C)
    x = x + self.drop_path(self.gamma2 * self.mlp(self.norm2(x))) # (N, H, W, C)
```

Neighborhood Attention Transformer Architecture



NAT Block

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

Table 1. NAT Variants.

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

Table 2. Computational cost and memory usage in different attention patterns and convolutions. SA has a quadratic complexity with respect to resolution, while WSA, NA, and convolutions have a linear complexity.

NAT

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**

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

Table 7. Ablation study on NAT, with Swin-T as the baseline.

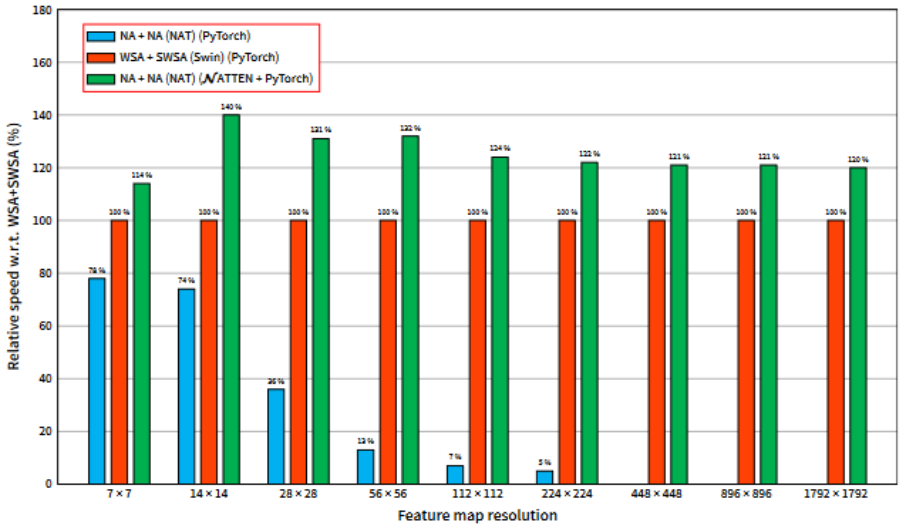


Figure 4. NAT's layer-wise relative speed with respect to Swin. Two NA layers with kernel size 7^2, are up to 40% faster than a pair of WSA and SWSA layers with the same kernel size. Latency is measured on a single A100 GPU. PyTorch implementation of NA runs out of memory at resolutions 448^2 and higher.

NAT

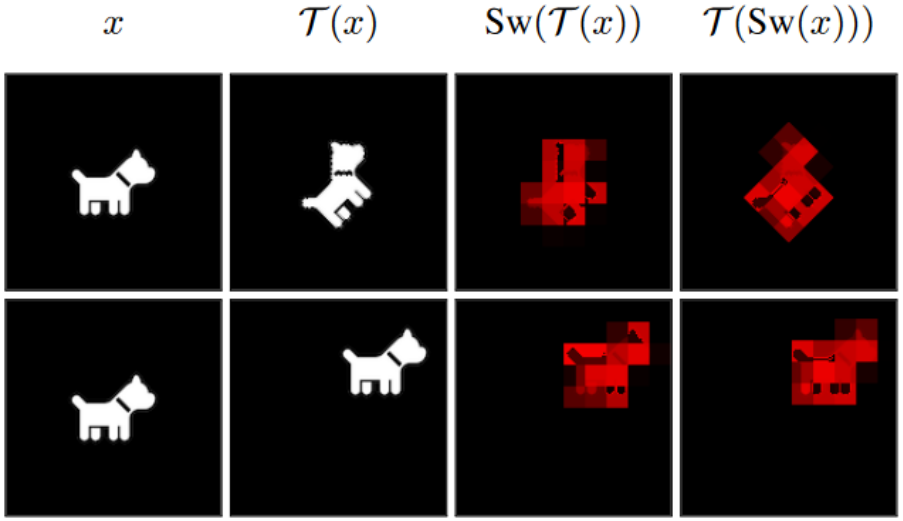


Figure V. Visualization of translations applied to Swin. $\mathcal{T}$ denotes the translation function. "Sw" denotes a WSA+SWSA applied to the input, with a residual connection in between. It is noticeable that this pattern breaks translational equivariance, especially in the case of rotations.

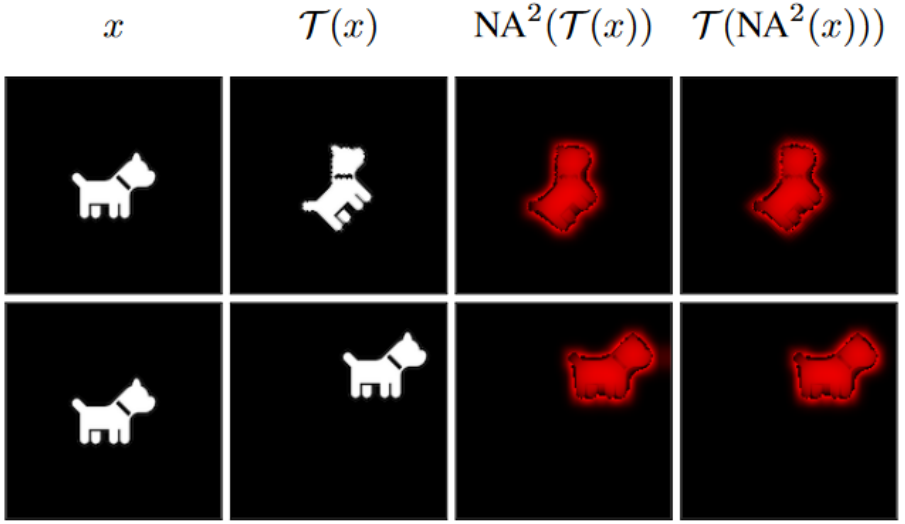


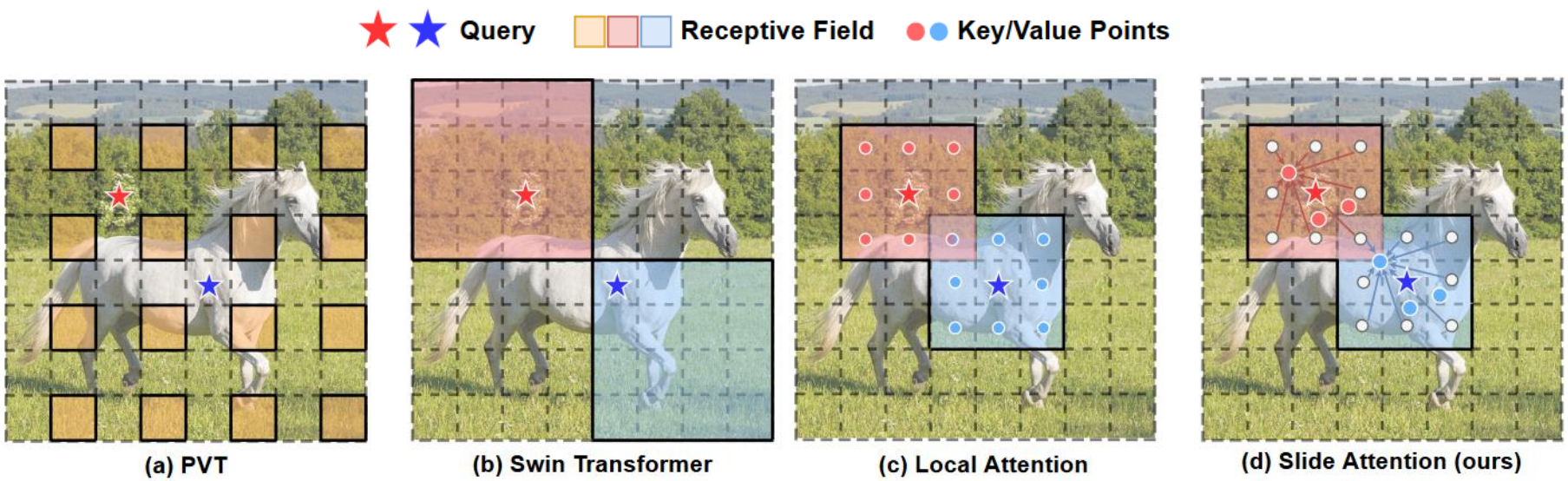
Figure VI. Visualization of translations applied to NAT. $\mathcal{T}$ denotes the translation function. "NA^2" denotes two layers of NA applied to the input, with a residual connection in between. It is easy to see how NA preserves translational equivariance with its sliding window property.

- 作者介绍
- 研究背景
- NAT
- Slide-Transformer
- BiFormer
- 后话

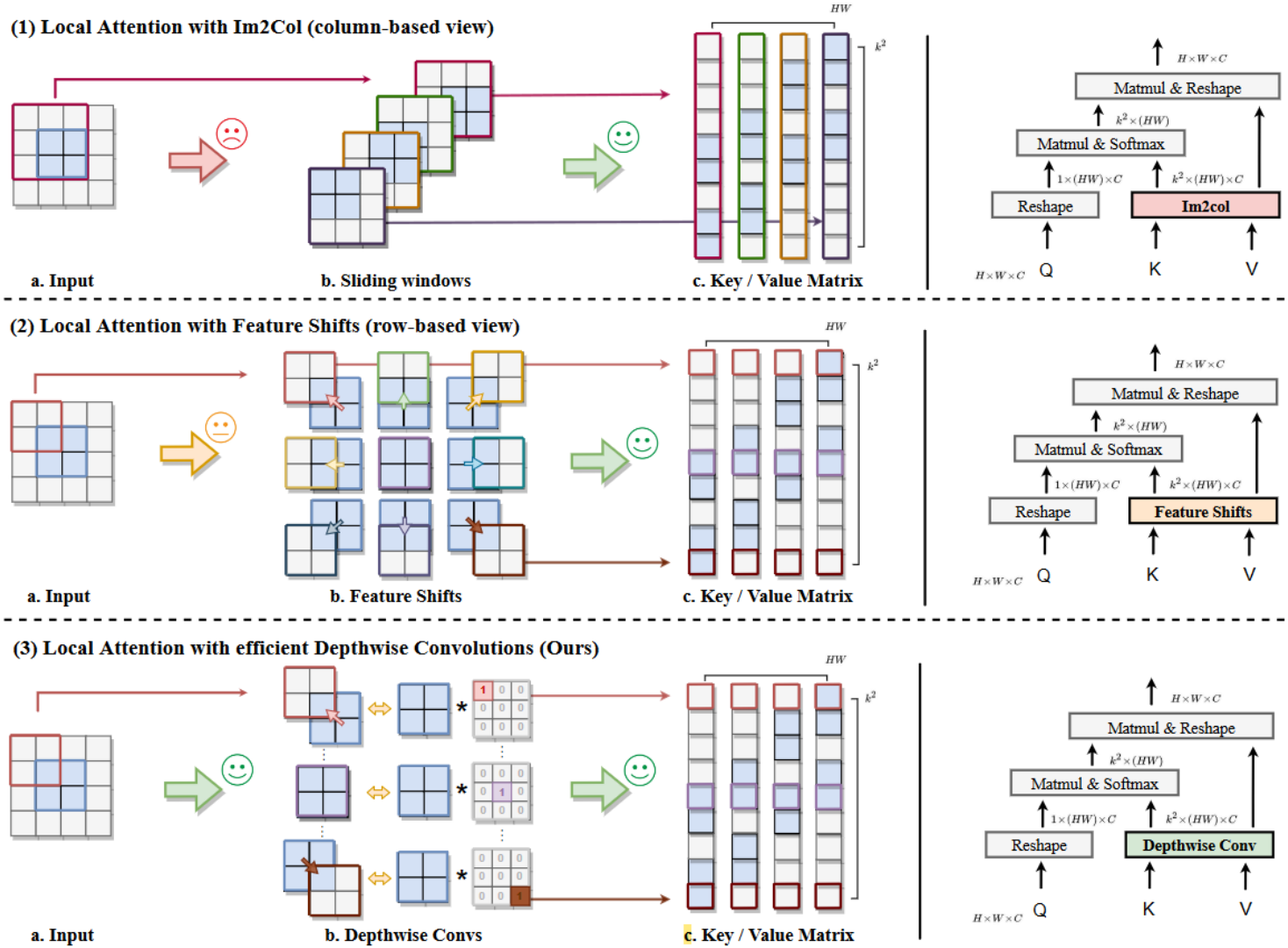
Slide-Transformer

Slide-Transformer: Hierarchical Vision Transformer with Local Self-Attention

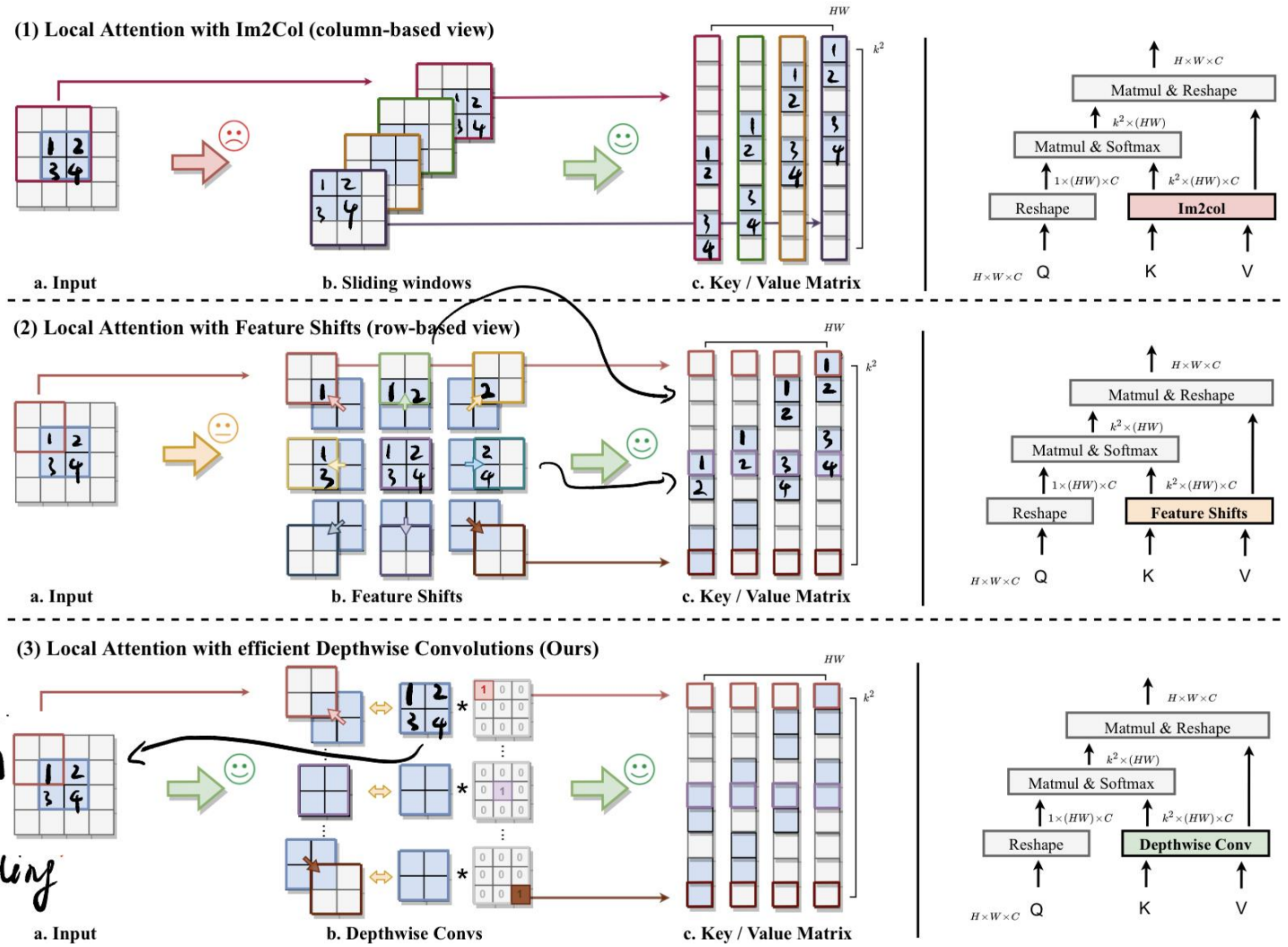
Xuran Pan* Tianzhu Ye* Zhuofan Xia Shiji Song Gao Huang†
Department of Automation, BNRist, Tsinghua University



Slide-Transformer

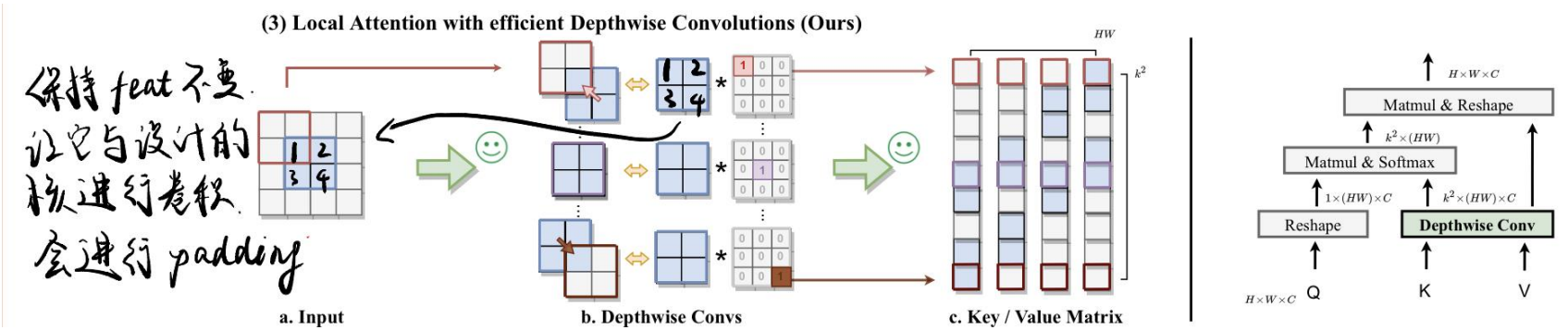


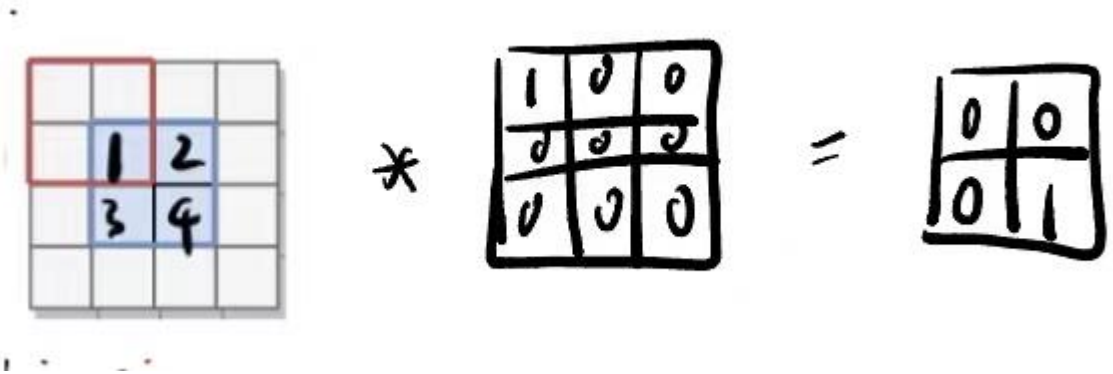
Slide-Transformer



保持 feat 不变
让它与设计的
核进行卷积
会进行 padding

Slide-Transformer





Slide-Transformer

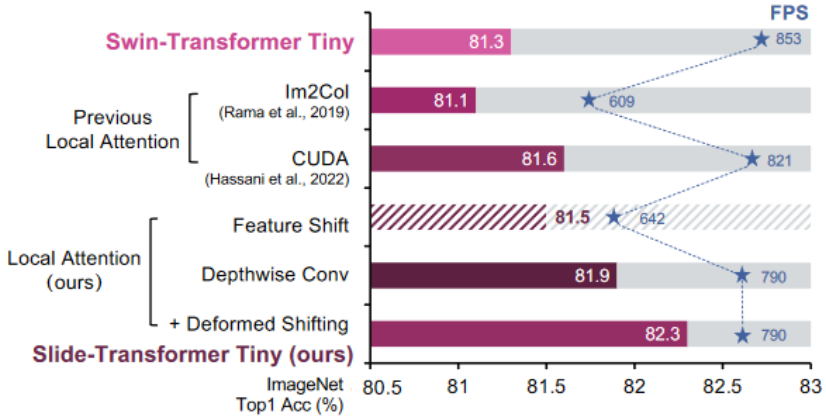


Figure 2. **Performance and inference speed comparison on local attention implementations.** Results are based on Swin-Tiny [21]. Previous works mainly use Im2Col function [27] or carefully designed CUDA kernels [12], where the former is highly inefficient and the latter only shows marginal improvements over other attention patterns, *e.g.,* window attention in Swin-Transformer, and hard to generalize to other devices. Our work first re-interprets the Im2Col function as feature shift operations, then substitute shifts with more efficient depthwise convolutions. When further equipped with a deformed shifting module, our model achieves significant improvements over baselines under competitive inference time. FPS is tested on an RTX3090 GPU.

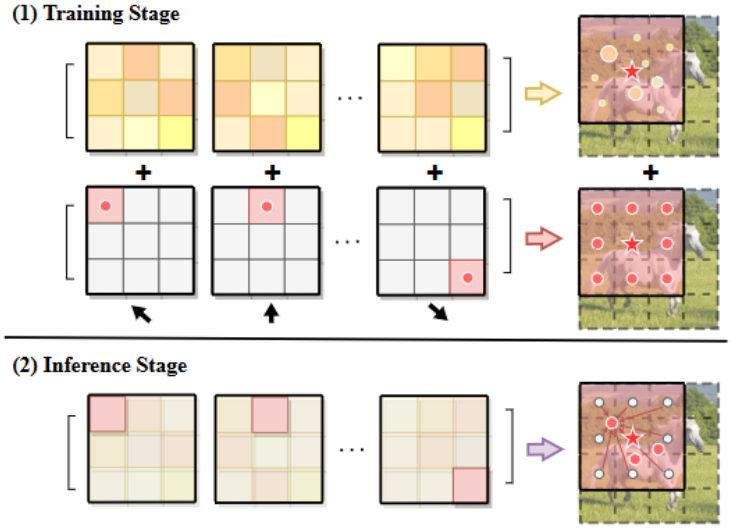
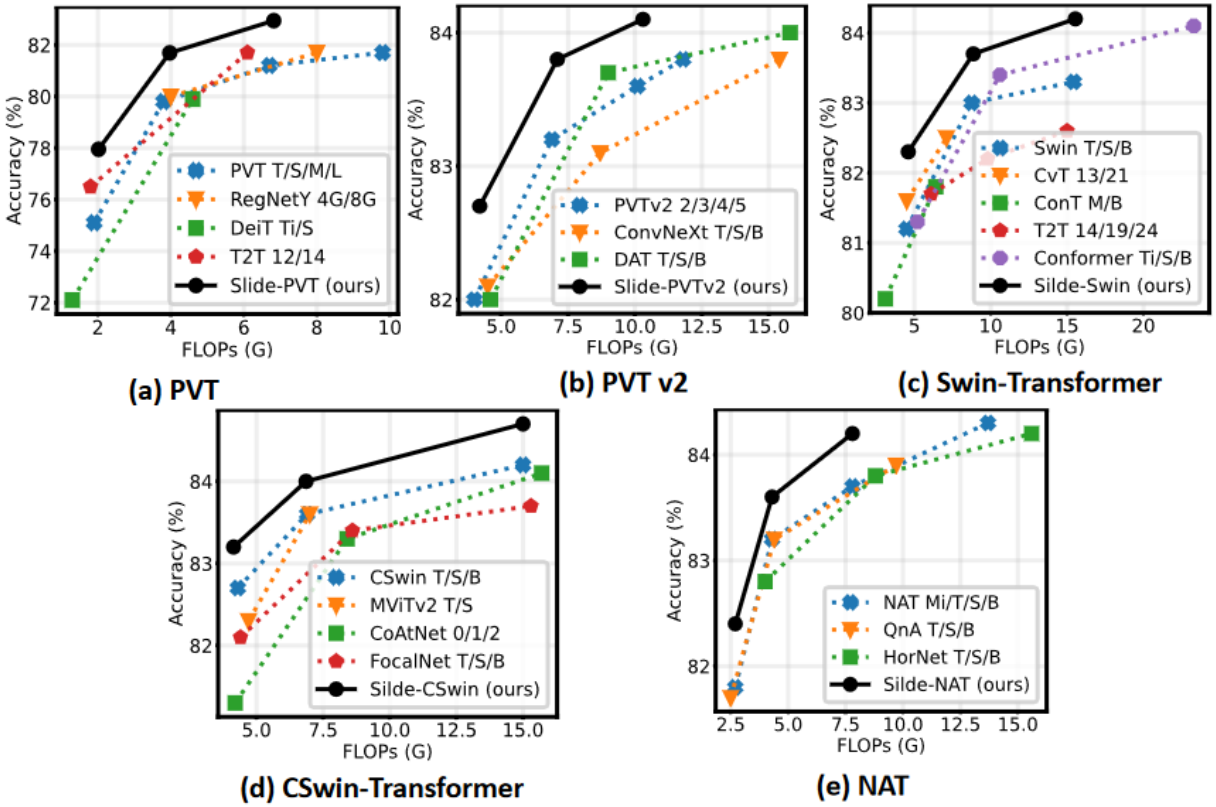


Figure 4. **Deformed shifting module with re-parameterization.** (1) At the training stage, we maintain two paths, one with designed kernel weights to perform shifting towards different directions, and the other with learnable parameters to enable more flexibility. (2) At the inference stage, we merge these two convolution operations into a single path with re-parameterization, which improves the model capacity while maintaining the inference efficiency.

the corresponding output can be formulated as:

Slide-Transformer



Method	Params	Flops	Top-1
PVT-T [32]	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 [33]	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 [21]	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 [9]	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 [9]	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)

Figure 5. Comparisons of FLOPS and parameters against accuracy on ImageNet classification task. Models in (a)(b) adapt from PVT and PVTv2 with global attention; Models in (c)(d) adapt from Swin-Transformer and CSwin-Transformer with window attention; Models in (e) adapt from NAT with local attention. See the full comparison table in Appendix.

Slide-Transformer

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 [21]				4.5G	29M	81.3	-1.0

Table 5. Ablation study on applying slide attention on different stages. All the models are based on the Swin-Tiny structure.

stage	output	Slide-Swin-T		Slide-Swin-S		Slide-Swin-B	
		Slide Attention	Swin Block	**Slide Attention**	Swin Block	**Slide Attention**	Swin Block
res1	56 × 56	concat 4 × 4, 96, LN		concat 4 × 4, 96, LN		concat 4 × 4, 128, LN	
		win 3×3, dim 96, head 3 ×2	None	win 3×3, dim 96, head 3 ×2	None	win 3×3, dim 128, head 3 ×2	None
res2	28 × 28	concat 4 × 4, 192, LN		concat 4 × 4, 192, LN		concat 4 × 4, 256, LN	
		win 3×3, dim 192, head 6 ×2	None	win 3×3, dim 192, head 6 ×2	None	win 3×3, dim 256, head 6 ×2	None
res3	14 × 14	concat 4 × 4, 384, LN		concat 4 × 4, 384, LN		concat 4 × 4, 512, LN	
		None	win 7×7, dim 384, head 12 ×6	None	win 7×7, dim 384, head 12 ×18	None	win 7×7, dim 512, head 12 ×18
res4	7 × 7	concat 4 × 4, 768, LN		concat 4 × 4, 768, LN		concat 4 × 4, 1024, LN	
		None	win 7×7, dim 768, head 24 ×2	None	win 7×7, dim 768, head 24 ×2	None	win 7×7, dim 1024, head 24 ×2

stage	output	Slide-PVT-T		Slide-PVT-S		Slide-PVT-M	
		Slide Attention	PVT Block	**Slide Attention**	PVT Block	**Slide Attention**	PVT Block
res1	56 × 56	Conv1×1, stride=4, 64, LN					
		win 3×3, dim 64, head 1 ×2	None	win 3×3, dim 64, head 1 ×3	None	win 3×3, dim 64, head 1 ×3	None
res2	28 × 28	Conv1×1, stride=2, 128, LN					
		win 3×3, dim 128, head 2 ×2	None	win 3×3, dim 128, head 2 ×3	None	win 3×3, dim 128, head 2 ×3	None
res3	14 × 14	Conv1×1, stride=2, 320, LN					
		None	win 7×7, dim 256, head 5 ×2	None	win 7×7, dim 256, head 5 ×6	None	win 7×7, dim 256, head 5 ×18
res4	7 × 7	Conv1×1, stride=2, 512, LN					
		None	win 7×7, dim 512, head 8 ×2	None	win 7×7, dim 512, head 8 ×3	None	win 7×7, dim 512, head 8 ×3

stage	output	Slide-NAT-Mini		Slide-NAT-Tiny		Slide-NAT-Small	
		Slide Attention	Swin Block	**Slide Attention**	Swin Block	**Slide Attention**	Swin Block
res1	56 × 56	2 * Conv3×3, stride=2, 64, LN				2 * Conv3×3, stride=2, 96, LN	
		win 3×3, dim 64, head 2 ×3	None	win 3×3, dim 64, head 2 ×3	None	win 3×3, dim 96, head 3 ×3	None
res2	28 × 28	Conv3×3, stride=2, 128, LN				Conv3×3, stride=2, 192, LN	
		win 3×3, dim 128, head 4 ×4	None	win 3×3, dim 128, head 4 ×4	None	win 3×3, dim 192, head 6 ×4	None
res3	14 × 14	Conv3×3, stride=2, 256, LN				Conv3×3, stride=2, 384, LN	
		None	win 7×7, dim 256, head 8 ×6	win 3×3, dim 256, head 8 ×10	win 7×7, dim 256, head 8 ×8	win 3×3, dim 384, head 12 ×10	win 7×7, dim 384, head 12 ×8
res4	7 × 7	Conv3×3, stride=2, 512, LN				Conv3×3, stride=2, 768, LN	
		None	win 7×7, dim 512, head 16 ×5	None	win 7×7, dim 512, head 16 ×5	None	win 7×7, dim 768, head 24 ×5

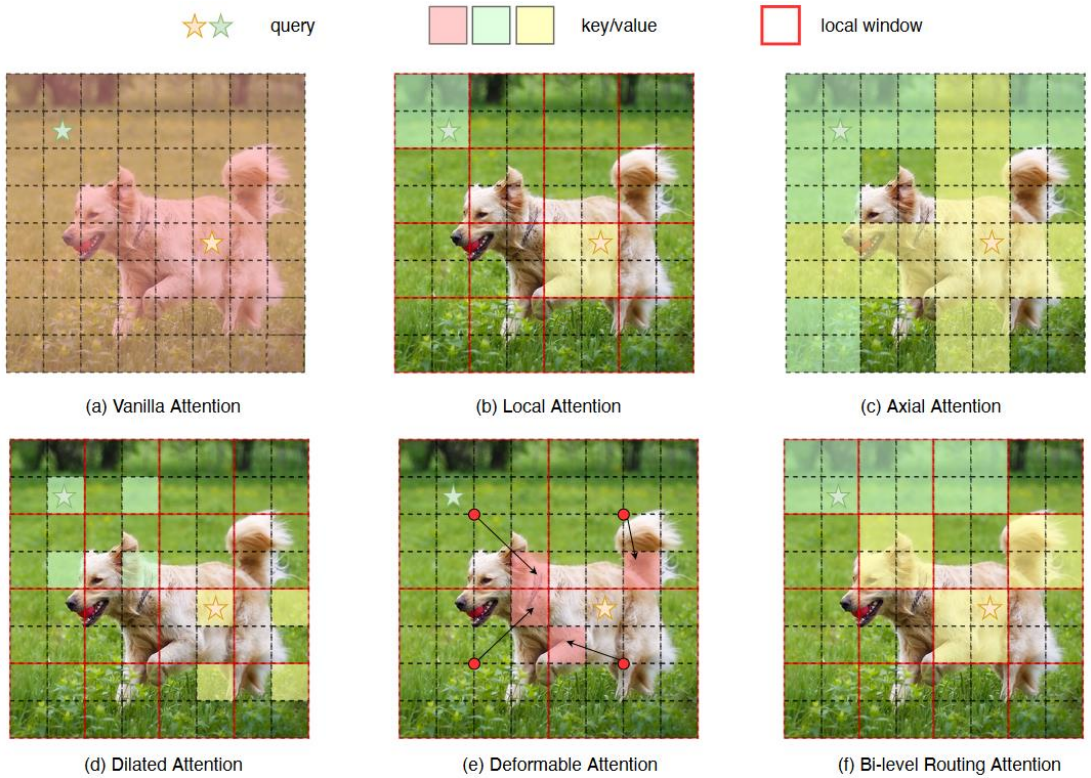
Table 15. Architectures of Slide-NAT models.

- 作者介绍
- 研究背景
- NAT
- Slide-Transformer
- BiFormer
- 后话

BiFormer

BiFormer: Vision Transformer with Bi-Level Routing Attention

Lei Zhu[1] Xinjiang Wang[2] Zhanghan Ke[1] Wayne Zhang[2] Rynson Lau[1†]
[1] City University of Hong Kong [2] SenseTime Research
{lzhu68-c,zhanghake2-c}@my.cityu.edu.hk, {wangxinjiang,wayne.zhang}@sensetime.com
Rynson.Lau@cityu.edu.hk

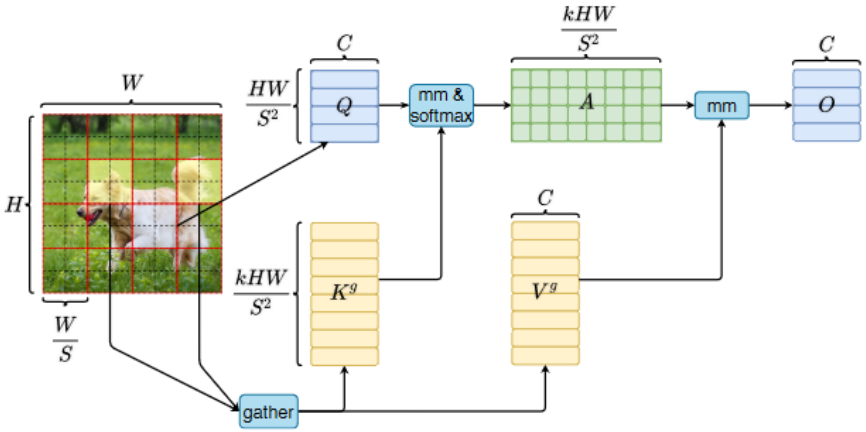


BiFormer

Algorithm 1 Pseudocode of BRA in a PyTorch-like style.

```
# input: features (H, W, C). Assume H==W.
# output: features (H, W, C).
# S: square root of number of regions.
# k: number of regions to attend.

# patchify input (H, W, C) -> (S^2, HW/S^2, C)
x = patchify(input, patch_size=H//S)

# linear projection of query, key, value
query, key, value = linear_qkv(x).chunk(3, dim=-1)

# regional query and key (S^2, C)
query_r, key_r = query.mean(dim=1), key.mean(dim=1)

# adjacency matrix for regional graph (S^2, S^2)
A_r = mm(query_r, key_r.transpose(-1, -2))

# compute index matrix of routed regions (S^2, K)
I_r = topk(A_r, k).index

# gather key-value pairs
key_g = gather(key, I_r) # (S^2, kHW/S^2, C)
value_g = gather(value, I_r) # (S^2, kHW/S^2, C)

# token-to-token attention
A = bmm(query, key_g.transpose(-2, -1))
A = softmax(A, dim=-1)
output = bmm(A, value_g) + dwconv(value)

# recover to (H, W, C) shape
output = unpatchify(output, patch_size=H//S)
```

bmm: batch matrix multiplication; mm: matrix multiplication. dwconv: depthwise convolution.



$$\mathbf{Q} = \mathbf{X}^r\mathbf{W}^q, \quad \mathbf{K} = \mathbf{X}^r\mathbf{W}^k, \quad \mathbf{V} = \mathbf{X}^r\mathbf{W}^v,$$

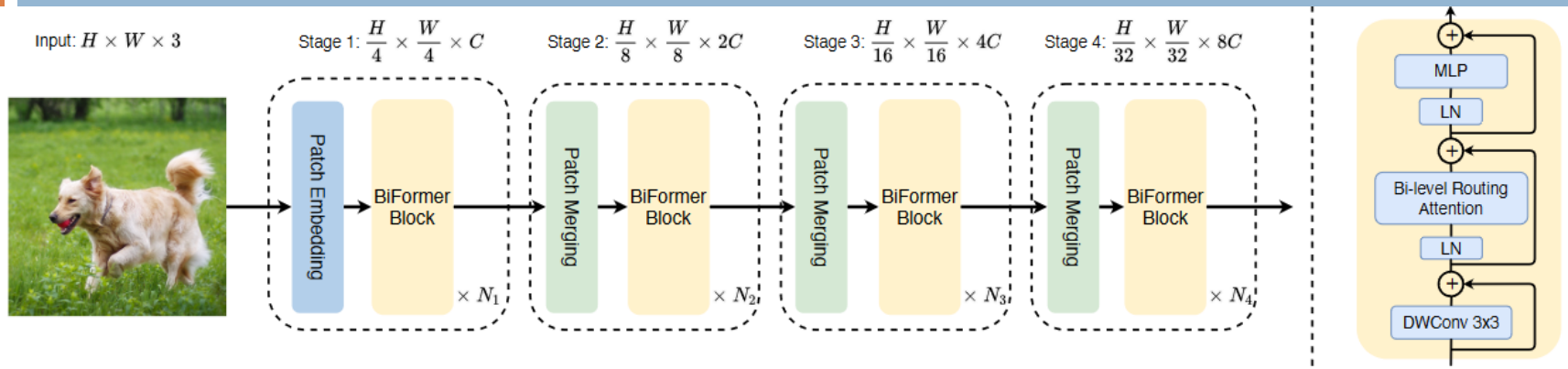
$$\mathbf{A}^r \in \mathbb{R}^{S^2 \times S^2} \qquad \mathbf{A}^r = \mathbf{Q}^r(\mathbf{K}^r)^T$$

$$\mathbf{I}^r = \text{topkIndex}(\mathbf{A}^r)$$

$$\mathbf{K}^g = \text{gather}(\mathbf{K}, \mathbf{I}^r), \quad \mathbf{V}^g = \text{gather}(\mathbf{V}, \mathbf{I}^r),$$

$$\mathbf{O} = \text{Attention}(\mathbf{Q}, \mathbf{K}^g, \mathbf{V}^g) + \text{LCE}(\mathbf{V}).$$

BiFormer



We instantiate BiFormer with 3 different model sizes by scaling the network width (*i.e.*, the number of base channels C) and depth (*i.e.*, the number of BiFormer blocks used in each stage, $N_i, i = 1, 2, 3, 4$), as listed in Table 1. They share other configurations. We set each attention head to 32 channels, and MLP expansion ratio e=3. For BRA, we use $topk = 1, 4, 16, S^2$[3] for the 4 stages, and region partition factor $S = 7/8/16$ for classification/semantic segmentation/object detection task, due to different input resolutions.

Models	#Channels.	#Blocks	Params	FLOPs
BiFormer-T	64	[2, 2, 8, 2]	13M	2.2G
BiFormer-S	64	[4, 4, 18, 4]	26M	4.5G
BiFomrer-B	96	[4, 4, 18, 4]	57M	9.8G

S	k	#tokens to attend	Acc	im/s (FP32)
7	1,4,16,49	64,64,64,49	**82.7**	522.3
7	1,2,8,32	64,32,32,32	82.4	563.2
7	2,8,32,49	128,128,128,49	82.6	419.9
8,4,2,1	2,2,2,1	98,98,98,49	82.3	**606.2**

Table 7. Ablation study on top-k and partition factor S.

BiFormer

Model	FLOPs (G)	Params (M)	Top-1 Acc. (%)
ResNet-18 [19]	1.8	11.7	69.8
RegNetY-1.6G [34]	1.6	11.2	78.0
PVTv2-b1 [45]	2.1	13.1	78.7
Shunted-T [37]	2.1	11.5	79.8
QuadTree-B-b1 [38]	2.3	13.6	80.0
BiFormer-T	2.2	13.1	**81.4**
Swin-T [29]	4.5	29	81.3
CSWin-T [14]	4.5	23	82.7
DAT-T [48]	4.6	29	82.0
CrossFormer-S [46]	5.3	31	82.5
RegionViT-S [2]	5.3	31	82.6
QuadTree-B-b2 [38]	4.5	24	82.7
MaxViT-T [41]	5.6	31	83.6
ScalableViT-S [50]	4.2	32	83.1
Uniformer-S*	4.2	24	83.4
Wave-ViT-S* [51]	4.7	23	83.9
BiFormer-S	4.5	26	**83.8**
BiFormer-S*	4.5	26	**84.3**
Swin-B [29]	15.4	88	83.5
CSWin-B [14]	15.0	78	84.2
CrossFormer-L [46]	16.1	92	84.0
ScalableViT-B [50]	8.6	81	84.1
Uniformer-B* [25]	8.3	50	85.1
Wave-ViT-B* [51]	7.2	34	84.8
BiFormer-B	9.8	57	**84.3**
BiFormer-B*	9.8	58	**85.4**

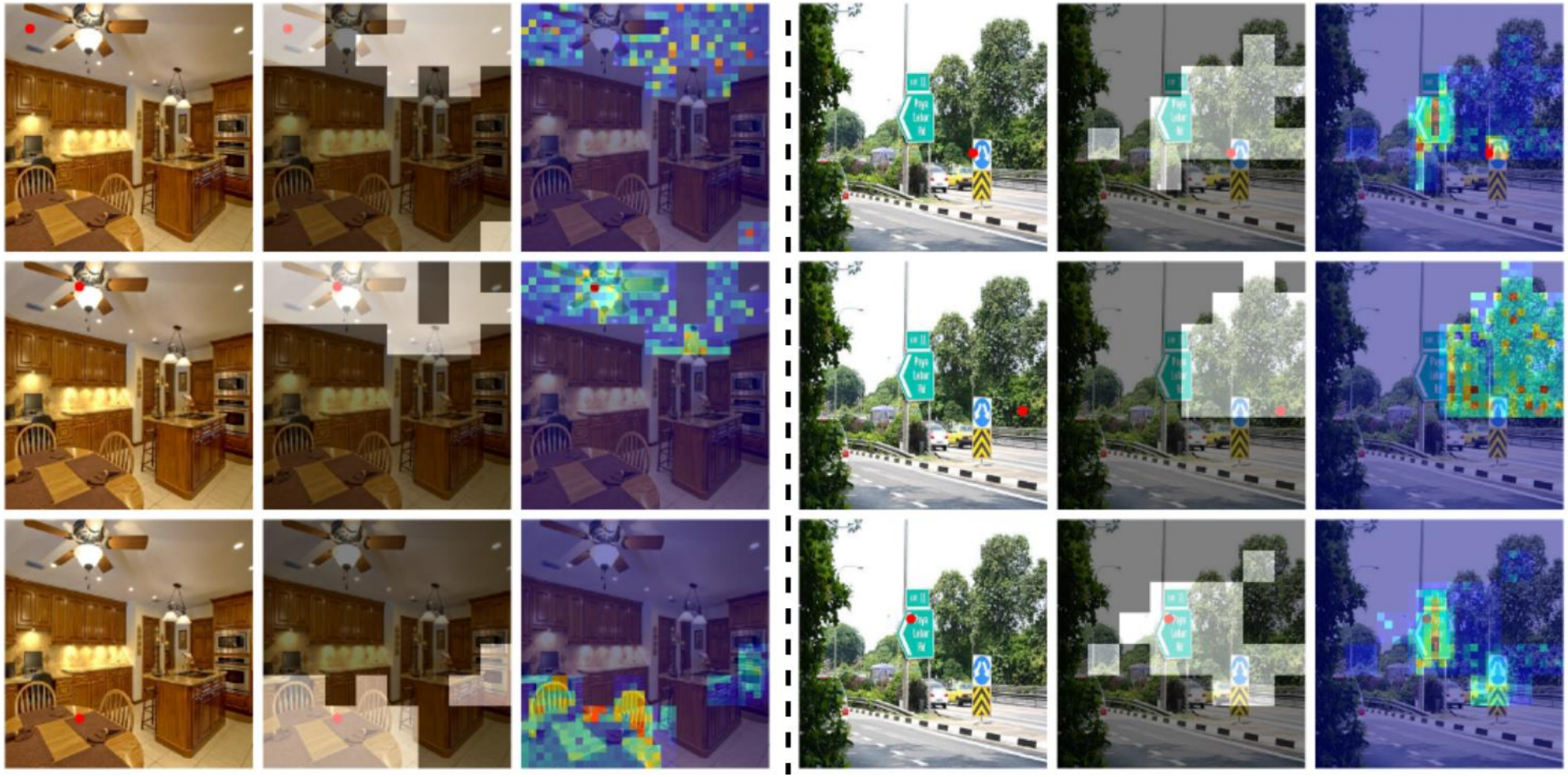
Sparse Attention	IN1K Top1(%)	ADE20K mIoU(%)
Sliding window [35]	81.4	-
Shifted window [29]	81.3	41.5
Spatially Sep [7]	81.5	42.9
Sequential Axial [20]	81.5	39.8
Criss-Cross [22]	81.7	43.0
Cross-shaped window [14]	82.2	43.4
Deformable [48]	82.0	42.6
Block-Grid [41]	81.8	42.8
Bi-level Routing	**82.7**	**44.8**

Table 5. Ablation study on different attention mechanisms. All models follow the architecture design of the Swin-T model.

Architecture design	Params (M)	FLOPs (G)	IN1K Top1 (%)
Baseline (Swin-T layout)	29	4.6	82.7
+Overlapped patch emb.	31	4.9	82.8 (+0.1)
+Deeper layout	25	4.5	83.5 (+0.7)
+Convolution pos. enc.	26	4.5	83.8 (+0.3)
+Token Labling	29	4.9	84.3 (+0.5)

Table 6. Ablation path from Swin-T [29] layout architecture to BiFormer-S. Note that the modifications are applied sequentially.

BiFormer



- 作者介绍
- 研究背景
- NAT
- Slide-Transformer
- BiFormer
- 后话

总结反思

- 以上三篇文章作为最新的非全局注意力的设计方法，在取得了较高精度的同时，实现了操作精简、加速、获得更灵活的局部感受野等等突破
- 局部注意力相比于全局注意力，可以引入local先验和一定程度上的平移不变性，有利于在较小规模数据集上的train-from-scratch，但是当模型的规模增大时，太过于局部的注意力往往会影响到模型的scalability

谢谢！