Vision Transformer with Deformable Attention

Zhuofan Xia[1*] Xuran Pan[1*] Shiji Song[1] Li Erran Li[2] Gao Huang[1,3†]

[1]Department of Automation, BNRist, Tsinghua University

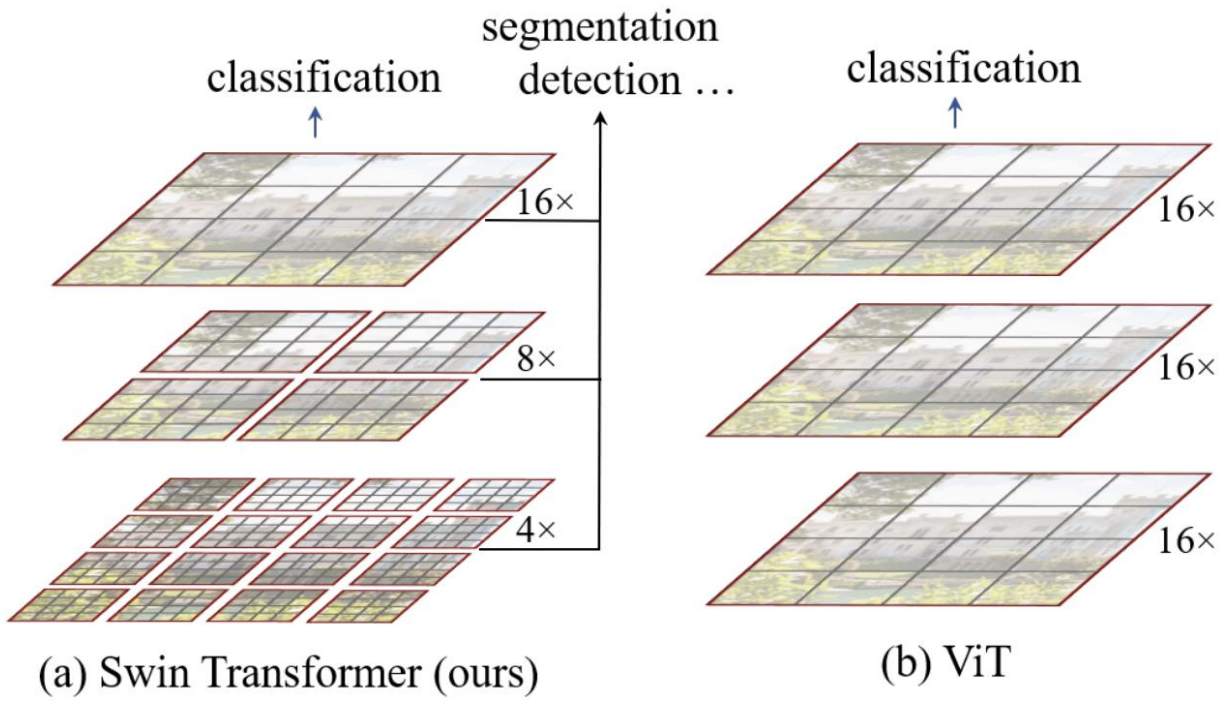
[2]AWS AI, Amazon [3]Beijing Academy of Artificial Intelligence

Brief Introduction

- Baseline：Swin-Transformer
- Problem of previous work:
  - PVT and Swin:
    - The downsampling technique of PVT results in severe information loss
    - The shift-window attention of Swin leads to a much slower growth of receptive fields, which limits the potential of modeling large objects
  - DCN’s non-trivial:
    - DCN just use HxWxC image feature to get 3x3 kernel and bias, computation complexity is about 9HWC
    - If in the same way with DCN, DAT’s complexity will drastically rise to N_qN_kC, where N_q and N_k have the same scale with HW
  - Thus a data-dependent sparse attention is required to flexibly model relevant features
- Main idea:
  - Deformable offset, an important improvement of self-attention
  - A new framework based on Swin-Transformer, using this deformable attention structure to replace part of the shift-window attention

Swin-Transformer

- Using hierarchical Transformer to address multi-scale features
- Split the image into non-overlapping windows and calculate the patch-wise self-attention in each window(W-MSA)
- Propose a shift-window strategy to allow cross-window connection(SW-MSA)



Preliminaries —— Self Attention

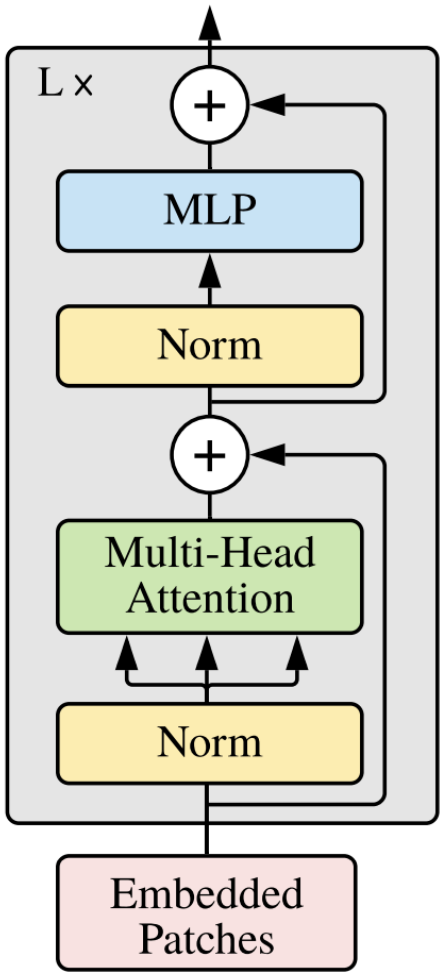
$$q = xW_q,\ k = xW_k,\ v = xW_v, \tag{1}$$

$$z^{(m)} = \sigma(q^{(m)}k^{(m)\top}/\sqrt{d})v^{(m)}, m=1,\ldots,M, \tag{2}$$

$$z = \text{Concat}\left(z^{(1)},\ldots,z^{(M)}\right)W_o, \tag{3}$$

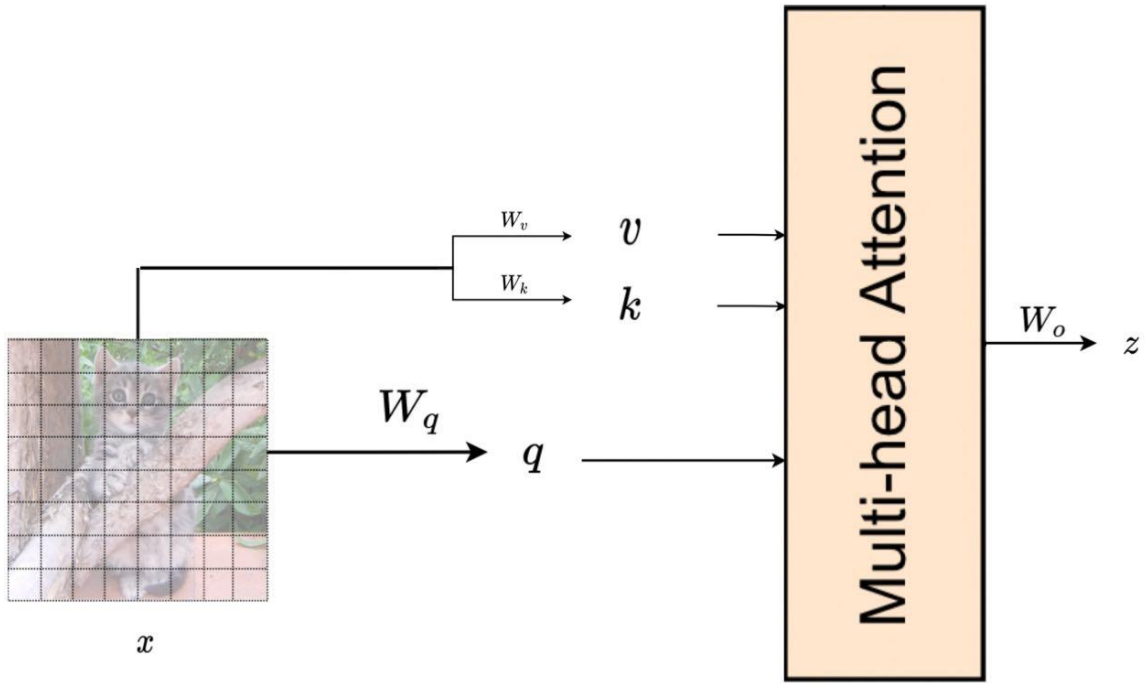
$$z'_l = \text{MHSA}\left(\text{LN}(z_{l-1})\right) + z_{l-1}, \tag{4}$$

$$z_l = \text{MLP}\left(\text{LN}(z'_l)\right) + z'_l, \tag{5}$$

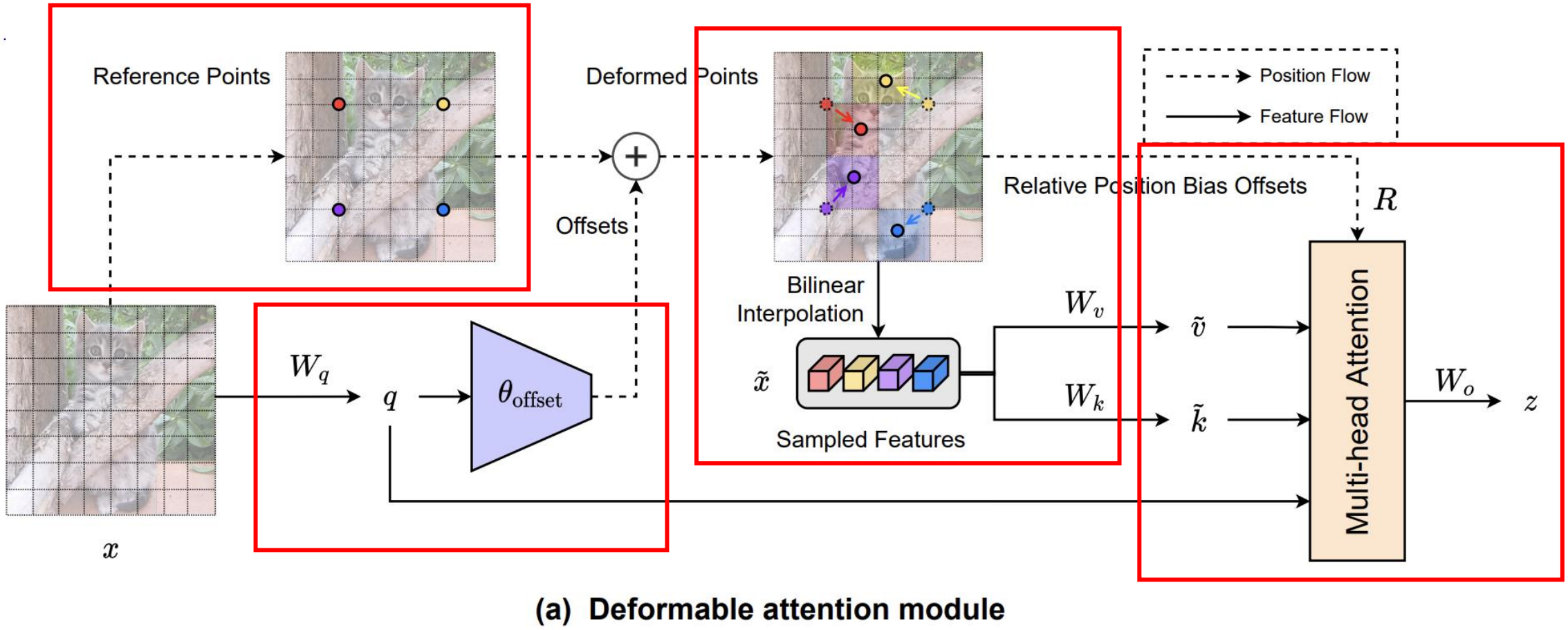


Preliminaries —— Self Attention

- W_q , W_k , W_V have the same shape of C x C
- Feature x has the shape of HW x C
- So feature q, feature k and feature v have the same shape HW x C

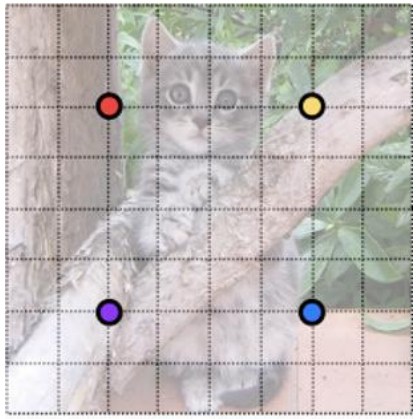


Method



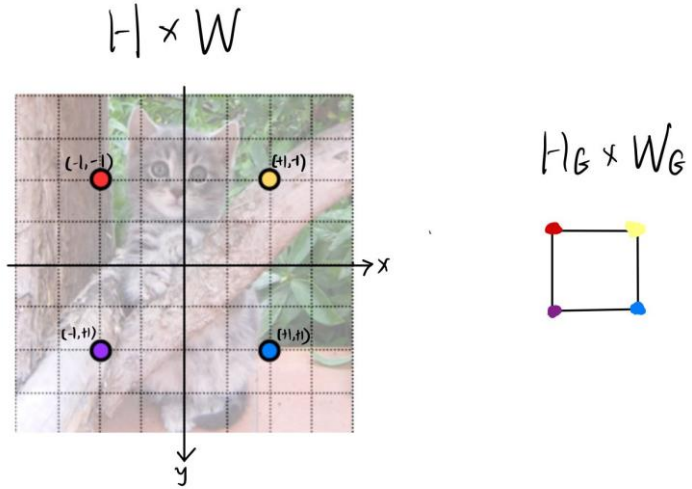
(a) Deformable attention module

Reference Points



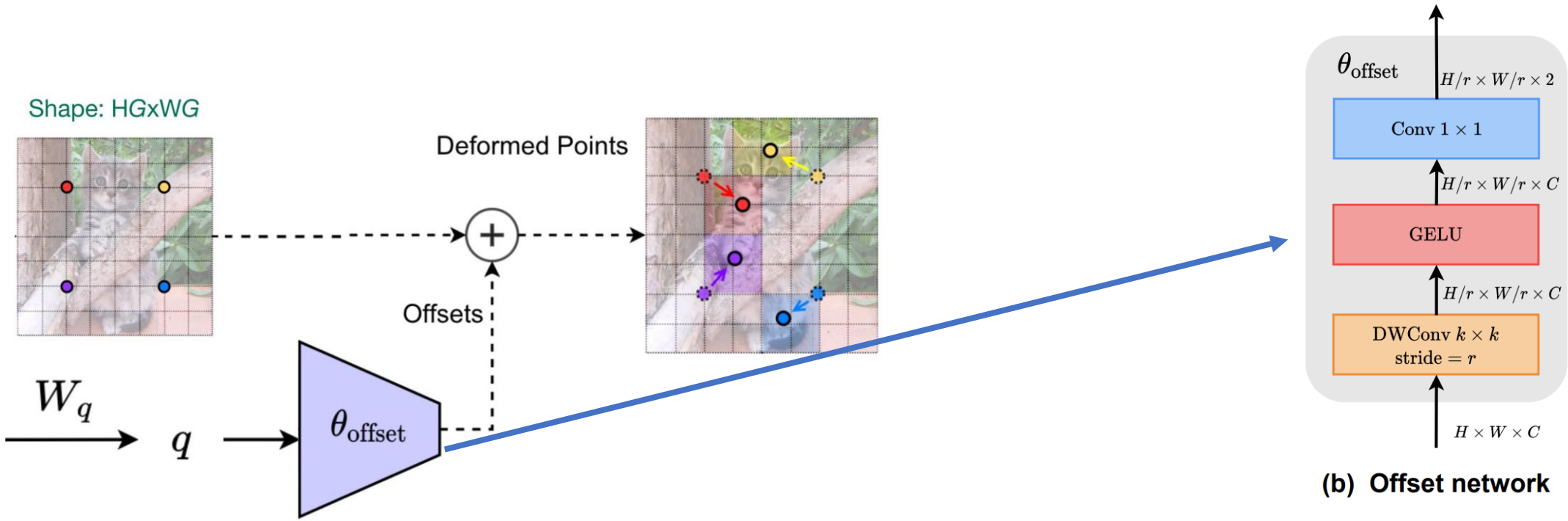
- Shape: $p \in \mathbb{R}^{H_G \times W_G \times 2}$
- $H_G = H/r, W_G = W/r$, r is downsampling factor
- The values of reference points are linearly spaced 2D coordinates $\{(0,0), \ldots, (H_G - 1, W_G - 1)\}$
- Then the coordinates is normalized to (-1, +1), where

(-1, -1) indicates top left and (+1, +1) indicates bottom right



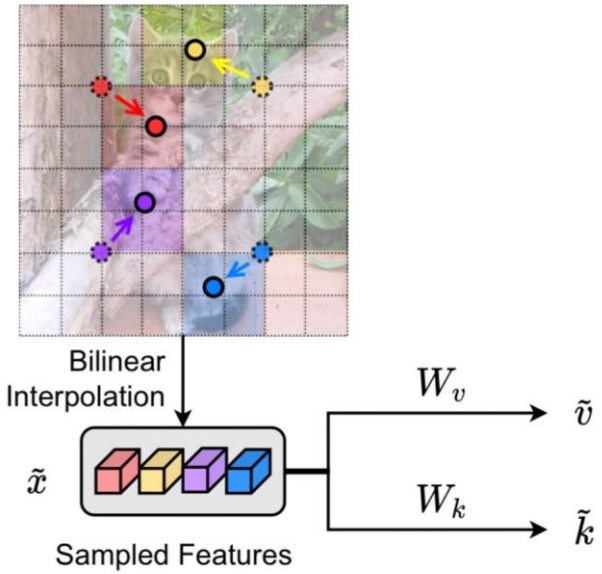
Offsets

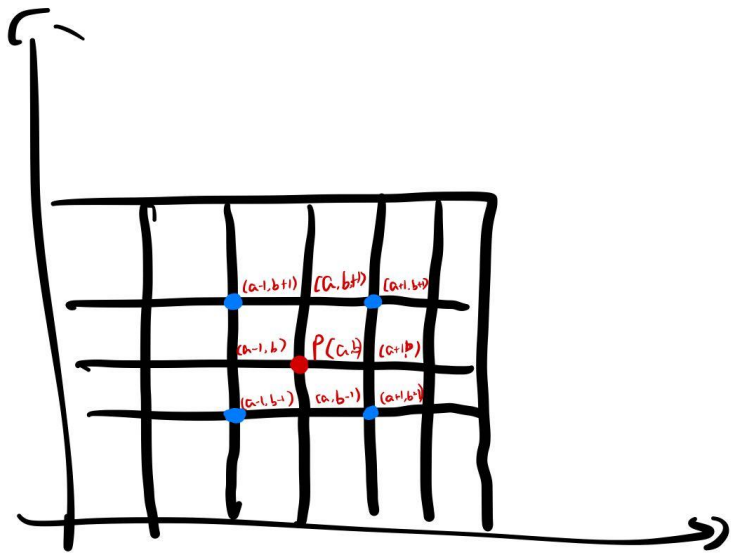
- Using offset network to get the offsets, input is tensor q, output is offset value
- Input shape is HxWxC, output shape is H/r x W/r x 2 (H_G x W_G x 2 , same shape as p)
- To avoid too large offset use predefined ***S*** and formulation $\Delta p \leftarrow s \tanh(\Delta p)$
- Offset value can be directly added to the reference point to get the deformed points



Local Bilinear Interpolation

- $\phi\left(z;(p_x,p_y)\right)=\sum_{(r_x,r_y)} g(p_x,r_x)g(p_y,r_y)z[r_y,r_x,:],\ \text{where } g(a,b)=\max(0,1-|a-b|)$
- $q=xW_q,\ \tilde{k}=\tilde{x}W_k,\ \tilde{v}=\tilde{x}W_v,\ \text{with } \Delta p=\theta_{\text{offset}}(q),\ \tilde{x}=\phi(x;p+\Delta p).$
- So $\tilde{x}$'s shape is H_G W_G x C





Other Details

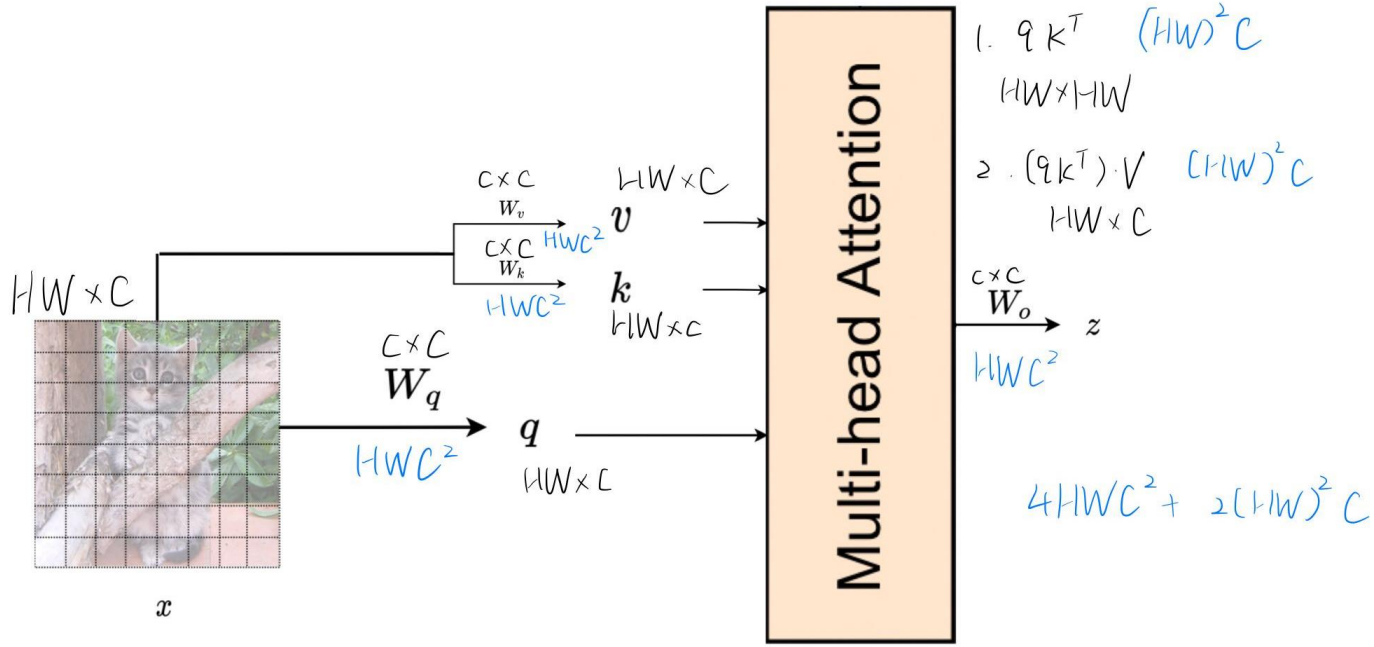
- Relative Position Bias Offset
  - a feature map with shape H×W, its relative coordinate displacements lie in the range of [−H, H] and [−W, W] at two dimensions.
  - In Swin Transformer and DAT, the relative position bias table is $\hat{B} \in \mathbb{R}^{(2H-1)\times(2W-1)}$
  - $\phi(\hat{B}; R) \in \mathbb{R}^{HW \times H_G W_G}$
  - $z^{(m)} = \sigma\left(q^{(m)}\tilde{k}^{(m)\top}/\sqrt{d} + \phi(\hat{B}; R)\right)\tilde{v}^{(m)}$
- Offset groups
  - To promote the diversity of the deformed points, split the feature channel into G groups
  - Features from each group use the shared sub-network to generate the corresponding offsets respectively
  - The head number M for the attention module is set to be multiple times of the size of offset groups G

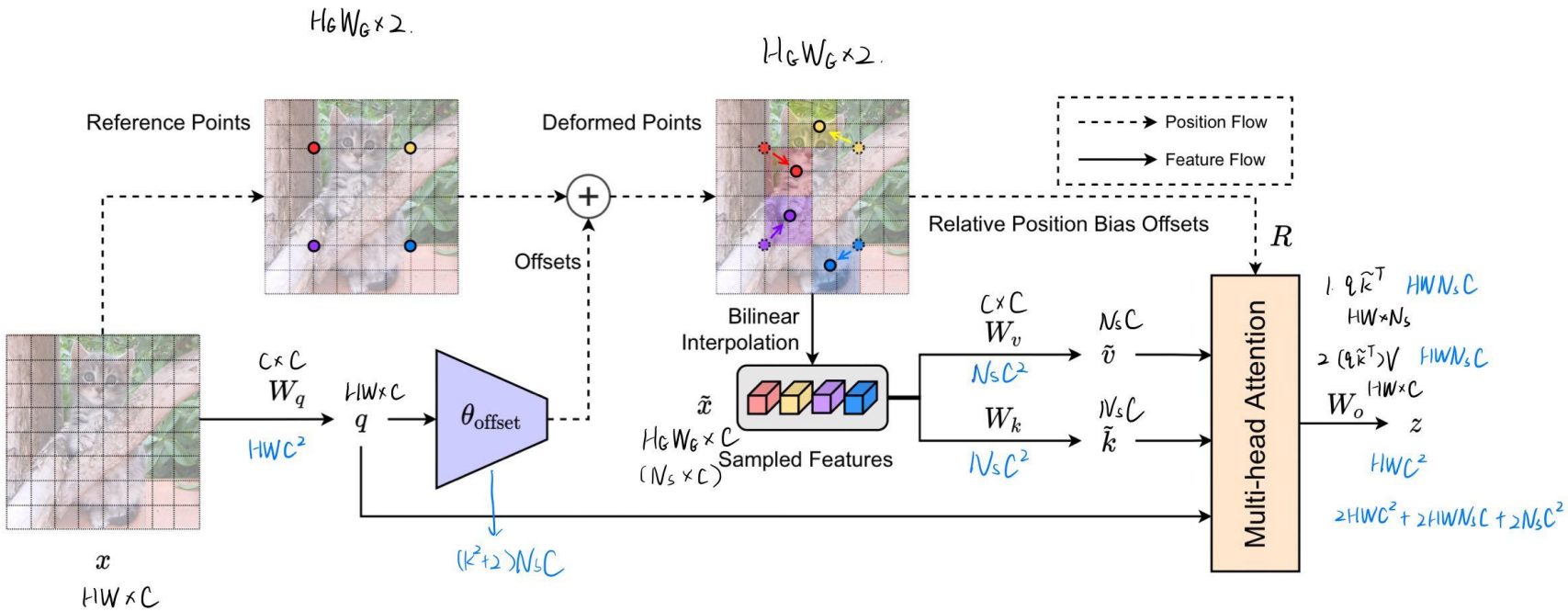
Computational complexity

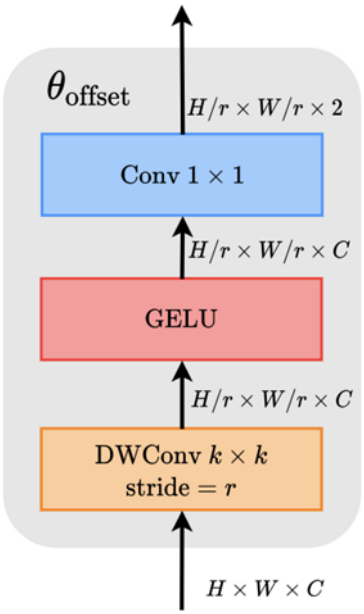
- MSA（multi head self attention）
  - $\Omega(\mathbf{MSA}) = 4hwC^2 + 2(hw)^2C,$

- W-MSA （window-based multi head self attention）
  - $\Omega(\mathbf{W\text{-}MSA}) = 4hwC^2 + 2M^2hwC,$

- DMHA (Deformable multi head self attention)
  - $\Omega(\mathbf{DMHA}) = \underbrace{2HWN_sC + 2HWC^2 + 2N_sC^2}_{\text{vanilla self-attention module}} + \underbrace{(k^2+2)N_sC}_{\text{offset network}}, \quad N_s = H_GW_G = HW/r^2$



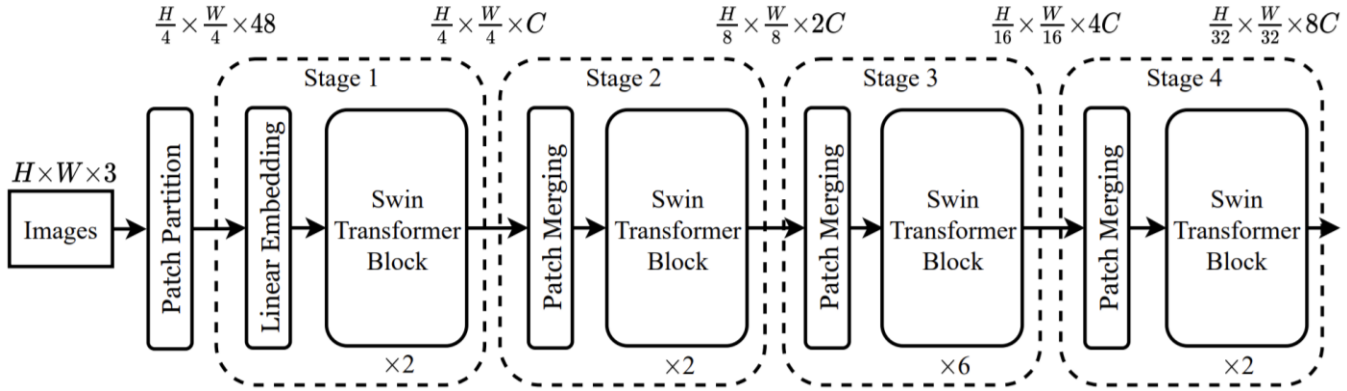




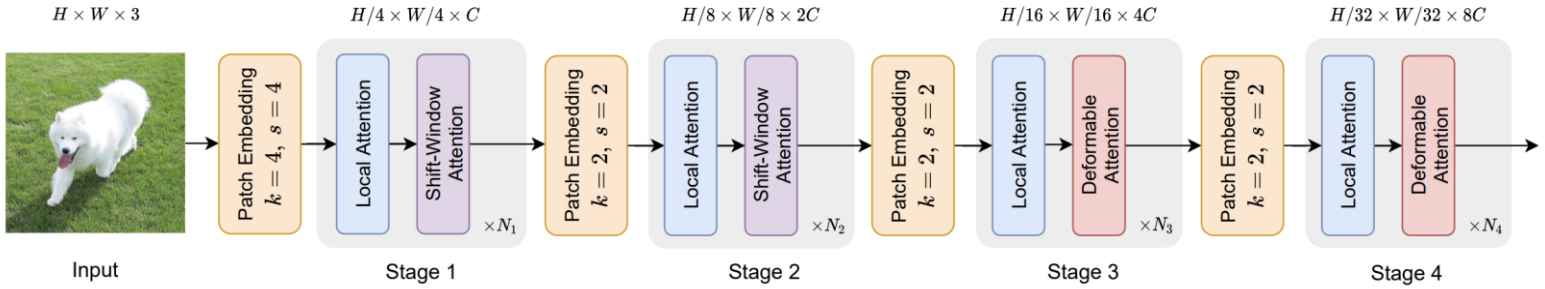
(b) Offset network

Architecture

- 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 =$\{2, 2, 18, 2\}$



(a) Swin-Transformer's Architecture



(b) DAT's Architecture

	DAT Architectures		
	DAT-T	DAT-S	DAT-B
Stage 1 (56×56)	$N_1 = 1, C = 96$ window size: 7 heads: 3	$N_1 = 1, C = 96$ window size: 7 heads: 3	$N_1 = 1, C = 128$ window size: 7 heads: 4
Stage 2 (28×28)	$N_2 = 1, C = 192$ window size: 7 heads: 6	$N_2 = 1, C = 192$ window size: 7 heads: 6	$N_2 = 1, C = 256$ window size: 7 heads: 8
Stage 3 (14×14)	$N_3 = 3, C = 384$ window size: 7 heads: 12 groups: 3	$N_3 = 9, C = 384$ window size: 7 heads: 12 groups: 3	$N_3 = 9, C = 512$ window size: 7 heads: 16 groups: 4
Stage 4 (7×7)	$N_4 = 1, C = 768$ window size: 7 heads: 24 groups: 6	$N_4 = 1, C = 768$ window size: 7 heads: 24 groups: 6	$N_4 = 1, C = 1024$ window size: 7 heads: 32 groups: 8

Table 1. Model architecture specifications. $\boldsymbol{N_i}$: Number of block at stage i. $\boldsymbol{C}$: Channel dimension. **window size**: Region size in local attention module. **heads**: Number of heads in DMHA. **groups**: Offset groups in DMHA.

Experiments on classification datasets

ImageNet-1K Classification				
Method	Resolution	FLOPs	#Param	Top-1 Acc.
DeiT-S [33]	224^2	4.6G	22M	79.8
PVT-S [36]	224^2	3.8G	25M	79.8
GLiT-S [5]	224^2	4.4G	25M	80.5
DPT-S [7]	224^2	4.0G	26M	81.0
Swin-T [26]	224^2	4.5G	29M	81.3
DAT-T	224^2	4.6G	29M	**82.0** (+0.7)
PVT-M [36]	224^2	6.9G	46M	81.2
PVT-L [36]	224^2	9.8G	61M	81.7
DPT-M [7]	224^2	6.9G	46M	81.9
Swin-S [26]	224^2	8.8G	50M	83.0
DAT-S	224^2	9.0G	50M	**83.7** (+0.7)
DeiT-B [33]	224^2	17.5G	86M	81.8
GLiT-B [5]	224^2	17.0G	96M	82.3
Swin-B [26]	224^2	15.5G	88M	83.5
DAT-B	224^2	15.8G	88M	**84.0** (+0.5)
DeiT-B [33]	384^2	55.4G	86M	83.1
Swin-B [26]	384^2	47.2G	88M	84.5
DAT-B	384^2	49.8G	88M	**84.8** (+0.3)

Table 2. Comparisons of DAT with other vision transformer backbones on FLOPs, parameters, accuracy on the ImageNet-1K classification task.

DAT

(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

Table 1. Comparison of different backbones on ImageNet-1K classification. Throughput is measured using the GitHub repository of [68] and a V100 GPU, following [63].

Swin

Experiments on detection datasets

RetinaNet Object Detection on COCO									
Method	FLOPs	#Param	Sch.	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
PVT-S	286G	34M	1x	40.4	61.3	43.0	25.0	42.9	55.7
Swin-T	248G	38M	1x	41.7	63.1	44.3	27.0	45.3	54.7
DAT-T	253G	38M	1x	42.8	64.4	45.2	28.0	45.8	57.8
PVT-S	286G	34M	3x	42.3	63.1	44.8	26.7	45.1	57.2
Swin-T	248G	38M	3x	44.8	66.1	48.0	29.2	48.6	58.6
DAT-T	253G	38M	3x	45.6	67.2	48.5	31.3	49.1	60.8
Swin-S	339G	60M	1x	44.5	66.1	47.4	29.8	48.5	59.1
DAT-S	359G	60M	1x	45.7	67.7	48.5	30.5	49.3	61.3
Swin-S	339G	60M	3x	47.3	68.6	50.8	31.9	51.8	62.1
DAT-S	359G	60M	3x	47.9	69.6	51.2	32.3	51.8	63.4

Table 3. Results on COCO object detection with RetinaNet [24] The table displays the number of parameters, computational cos (FLOPs), mAP at different mIoU thresholds and different objec sizes. The FLOPs are computed over backbone, FPN and detectior head with RGB input image at the resolution of 1280×800.

(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
Swin-T	267G	48M	1x	43.7	66.6	47.7	28.5	47.0	57.3	39.8	63.3	42.7	24.2	43.1	54.6
DAT-T	272G	48M	1x	44.4	67.6	48.5	28.3	47.5	58.5	40.4	64.2	43.1	23.9	43.8	55.5
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
DAT-T	272G	48M	3x	47.1	69.2	51.6	32.0	50.3	61.0	42.4	66.1	45.5	27.2	45.8	57.1
Swin-S	359G	69M	1x	45.7	67.9	50.4	29.5	48.9	60.0	41.1	64.9	44.2	25.1	44.3	56.6
DAT-S	378G	69M	1x	47.1	69.9	51.5	30.5	50.1	62.1	42.5	66.7	45.4	25.5	45.8	58.5
Swin-S	359G	69M	3x	48.5	70.2	53.5	33.4	52.1	63.3	43.3	67.3	46.6	28.1	46.7	58.6
DAT-S	378G	69M	3x	49.0	70.9	53.8	32.7	52.6	64.0	44.0	68.0	47.5	27.8	47.7	59.5
(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	1x	48.1	67.1	52.2	30.4	51.5	63.1	41.7	64.4	45.0	24.0	45.2	56.9
DAT-T	750G	86M	1x	49.1	68.2	52.9	31.2	52.4	65.1	42.5	65.4	45.8	25.2	45.9	58.6
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
DAT-T	750G	86M	3x	51.3	70.1	55.8	34.1	54.6	66.9	44.5	67.5	48.1	27.9	47.9	60.3
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
DAT-S	857G	107M	3x	52.7	71.7	57.2	37.3	56.3	68.0	45.5	69.1	49.3	30.2	49.2	60.9
Swin-B	982G	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
DAT-B	1003G	145M	3x	53.0	71.9	57.6	36.0	56.8	69.1	45.8	69.3	49.5	29.2	49.5	61.9

Table 4. Results on COCO object detection and instance segmentation. The table displays the number of parameters, computational cost (FLOPs), mAP at different IoU thresholds and mAP for objects in different sizes. The FLOPs are computed over backbone, FPN and detection head with RGB input image at the resolution of 1280×800.

Experiments on semantic segmentation datasets

Semantic Segmentation on ADE20K						
Backbone	Method	FLOPs	#Params	mIoU	mAcc	mIoU†
PVT-S	S-FPN	225G	28M	41.95	53.02	41.95
DAT-T	S-FPN	198G	32M	**42.56**	54.72	44.22
PVT-M	S-FPN	315G	48M	42.91	53.80	43.34
DAT-S	S-FPN	320G	53M	**46.08**	58.17	48.46
PVT-L	S-FPN	420G	65M	43.49	54.62	43.92
DAT-B	S-FPN	481G	92M	**47.02**	59.47	49.01
Swin-T	UperNet	945G	60M	44.51	55.61	45.81
DAT-T	UperNet	957G	60M	**45.54**	57.95	46.44
Swin-S	UperNet	1038G	81M	47.64	58.78	49.47
DAT-S	UperNet	1079G	81M	**48.31**	60.44	49.84
Swin-B	UperNet	1188G	121M	48.13	59.13	49.72
DAT-B	UperNet	1212G	121M	**49.38**	61.82	50.55

Table 5. Results of semantic segmentation. The FLOPs are computed over encoders and decoders with RGB input image at the resolution of 512×2048. † denotes the metrics are reported under a multi-scale test setting with flip augmentation. S-FPN is short for SemanticFPN [22] model. The results of PVT and Swin Transformer are copied from their Github repositories, which are higher than the versions in their original papers.

ADE20K		val	test			
Method	Backbone	mIoU	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

Table 3. Results of semantic segmentation on the ADE20K val and test set. † indicates additional deconvolution layers are used to produce hierarchical feature maps. ‡ indicates that the model is pre-trained on ImageNet-22K.

Ablation Study

Attn.	Offsets	Pos. Emebd	FLOPs	#Param	Acc.	Diff.
S	✗	✗	4.57G	28.29M	81.4	-0.6
S	✗	**Relative**	4.57G	28.32M	81.7	-0.3
S	✓	✗	4.58G	28.29M	81.7	-0.3
S	✓	Fixed	4.58G	29.73M	81.8	-0.2
S	✓	DWConv	4.59G	28.31M	81.8	-0.2
P	✓	**Relative**	4.48G	30.68M	81.7	-0.3
S	✓	**Relative**	4.59G	28.32M	82.0	DAT

Table 6. Ablation study on different ways to exploiting geometric information. **P** represents the first two stages use SRA attention in [36], and **S** represents shift-window attention in [26]. ✓ in offsets means performing spatial sampling in deformable attention module while ✗ means not.

Stages w/ Deformable Attention				FLOPs	#Param	Acc.
Stage 1	Stage 2	Stage 3	Stage 4			
✓	✓	✓	✓	4.64G	28.39M	81.7
	✓	✓	✓	4.60G	28.34M	81.9
		✓	✓	4.59G	28.32M	82.0
			✓	4.51G	28.29M	81.4
Swin-T [26]				4.51G	28.29M	81.3

Table 7. Ablation study on applying deformable attention on different stages. ✓ means this stage is made up of successive local attention and deformable attention Transformer blocks. Note that our model takes the relative position indices of all local and shift-window attention and the reference grid points of all deformable attention into parameter counting, which may lead to a higher number of parameters.

Visualization

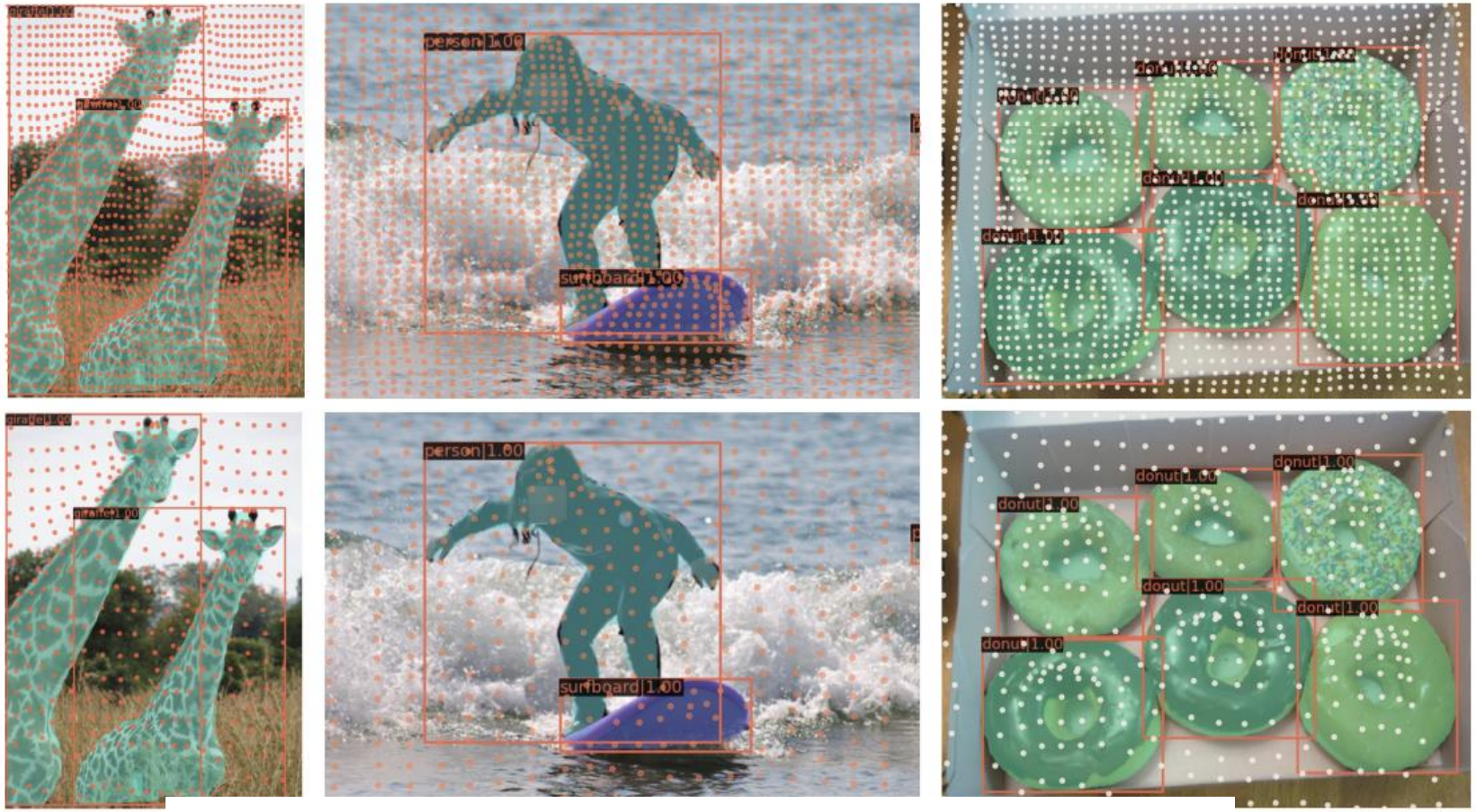


Figure 4. Visualizations on COCO [25] of learned sampling locations in deformable attention at Stage 3 (first row) and Stage 4 (second row) of DAT. The orange and yellow points show one group of deformed points. The detection bounding boxes and segmentation masks are also presented to indicate the targets.