Learning Correlation Structures for Vision Transformers
Supplementary Material

Manjin Kim[1] Paul Hongsuck Seo[2*] Cordelia Schmid[3] Minsu Cho[1*]
[1]POSTECH [2]Korea University [3]Google Research
http://cvlab.postech.ac.kr/research/StructViT/

Here we provide additional details and experimental results not included in our main paper due to the lack of space.

A. Full Derivation

A.1. Structural Self-Attention

In Sec.3.2 of our main paper, we explain StructSA with a slightly simpler version that uses dot-product correlation. Here we provide the full version of StructSA used in our experiments, which captures fine-grained correlation structures by employing channel-wise correlation.

Channel-wise Correlation. Since the dot product correlation $\boldsymbol{q}_i\boldsymbol{K}^\mathsf{T}$ reduces all the channels of the query and the keys, it might lose rich semantic information. We instead use the Hadamard product [5, 6, 12] to leverage richer channel-wise correlation structures for generating the final attention weights. SQKA extracts structural patterns from the channel-wise correlation applying convolution as

$$\boldsymbol{A}_i = \sigma\left(\texttt{conv}\left(\text{diag}(\boldsymbol{q}_i)\boldsymbol{K}, \mathbf{H}^\text{K}\right)\right) \in \mathbb{R}^{N\times D}, \quad (11)$$

$$\mathbf{H}^\text{K} = [\boldsymbol{H}^\text{K}_1, \cdots, \boldsymbol{H}^\text{K}_D] \in \mathbb{R}^{D\times M\times C}, \quad (12)$$

where $\text{diag}(\cdot)$ is a function that outputs a square diagonal matrix from an input vector and $\mathbf{H}^\text{K}$ represents D convolutional filters of which kernel size and input channel are M and C, respectively. Each score of $\boldsymbol{A}_i$ is computed as

$$\boldsymbol{a}_{i,j} = \sigma_j\left(\text{vec}\left(\text{diag}\left(\boldsymbol{q}_i\right)\boldsymbol{K}_j\right) f(\mathbf{H}^\text{K})^\mathsf{T}\right), \quad (13)$$

$$f(\mathbf{H}^\text{K}) = [\text{vec}(\boldsymbol{H}^\text{K}_1), \cdots, \text{vec}(\boldsymbol{H}^\text{K}_D)] \in \mathbb{R}^{D\times MC}, \quad (14)$$

where $\text{vec}(\cdot)$ is a vectorization function. Compared to $\boldsymbol{U}^\text{K}$ where each column takes a single correlation map to detect a structural pattern, each of $f(\mathbf{H}^\text{K})$, *i.e.*, $\text{vec}(\boldsymbol{H}^\text{K}_d)$, extracts a pattern from the whole C correlation maps using fine-grained channel-wise correlation. One potential drawback of the channel-wise correlation map, $\text{diag}(\boldsymbol{q}_i)\boldsymbol{K}$, would be to increase the memory complexity C-times larger compared to that of dot-product correlation map, *i.e.*, $\mathcal{O}(N^2C)$ vs. $\mathcal{O}(N^2)$. To address the issue, we permute the computation orders of Eq. 13 as

$$\begin{aligned}\boldsymbol{a}_{i,j} &= \sigma_j\left(\sum_{c=1}^{C}\sum_{m=1}^{M}(\boldsymbol{q}_i)_c(\boldsymbol{K}_j)_{m,c}(\mathbf{H}^\text{K})_{:,m,c}\right)\\ &= \sigma_j\left(\sum_{c=1}^{C}(\boldsymbol{q}_i)_c\sum_{m=1}^{M}(\boldsymbol{K}_j)_{m,c}(\mathbf{H}^\text{K})_{:,m,c}\right)\\ &= \sigma_j\left(\boldsymbol{q}_i(\boldsymbol{K}_j * \mathbf{H}^\text{K})\right). \end{aligned} \quad (15)$$

where we first compute $\boldsymbol{K}_j * \mathbf{H}^\text{K}$, which requires memory complexity of $\mathcal{O}(NCD)$, and then multiply it with $\boldsymbol{q}_i$. This enables us to compute $\boldsymbol{a}_{i,j}$ in a memory-efficient way when $N > D$ without explicit computation of channel-wise correlation maps.

Channel-wise Context Value Aggregation. We also extend the context aggregators $\boldsymbol{U}^\text{V}$, which are shared by different channels, to be channel-wise aggregators, so that they can learn aggregation weights more adaptive to each channel of the values. Each channel of StructSA output is computed as

$$(\boldsymbol{y}_i)_c = \sum_{j=1}^{N}\sigma_j\left(\text{vec}\left(\text{diag}(\boldsymbol{q}_i)\boldsymbol{K}_j\right) f(\mathbf{H}^\text{K})^\mathsf{T}\right)(\mathbf{H}^\text{V})_{:,:,c}(\boldsymbol{V}_j)_{:,c}, \quad (16)$$

$$\mathbf{H}^\text{V} = [\boldsymbol{H}^\text{V}_1, \cdots, \boldsymbol{H}^\text{V}_D] \in \mathbb{R}^{D\times M\times C}, \quad (17)$$

Compared to $\boldsymbol{U}^\text{V}$ that produces a single kernel shared by every channel, $(\mathbf{H}^\text{V})_{:,:,c}$ generates C different spatial kernels aggregating the context with diverse patterns. We conduct experiments to investigate the effect of utilizing channel-wise correlation and context aggregation in Sec. B. We use this version of StructSA as a basic operation in our main paper.

*Co-corresponding authors.

A.2. Convolutional Self-Attention

For the sake of simplicity in derivation, ConvSA (Eqs.8-10) in Sec.3.3 is described as sharing channel-wise convolution weights across channels for key and value projection, which is not exactly the same as those used in previous ConvSA-based methods [2, 9, 15]. We here provide a full derivation of ConvSA with conventional channel-wise convolution, of which weights are not shared across channels. Given the channel-wise convolution weights $\boldsymbol{H}^{\mathrm{K}}, \boldsymbol{H}^{\mathrm{V}} \in \mathbb{R}^{M\times C}$, c-th channel of each key k_i^{conv} and value v_i^{conv} is computed as

$$(\boldsymbol{k}_i^{\mathrm{conv}})_c = ({\boldsymbol{H}^{\mathrm{K}}}^{\mathsf{T}})_c(\boldsymbol{K}_i)_{:,c} \in \mathbb{R}, \tag{18}$$

$$(\boldsymbol{v}_i^{\mathrm{conv}})_c = ({\boldsymbol{H}^{\mathrm{V}}}^{\mathsf{T}})_c(\boldsymbol{V}_i)_{:,c} \in \mathbb{R}, \tag{19}$$

where $(\boldsymbol{K}_i)_{:,c}, (\boldsymbol{V}_i)_{:,c} \in \mathbb{R}^{M\times 1}$ indicate features in the c-th channel of $(\boldsymbol{K}_i)$ and $(\boldsymbol{V}_i)$, respectively. Plugging Eqs. 18 and 19 into Eq.10, each channel of ConvSA output is computed as

$$\begin{aligned}
(\boldsymbol{y}_i)_c &= \sum_{j=1}^{N} \sigma_j \left(\boldsymbol{q}_i {\boldsymbol{k}_j^{\mathrm{conv}}}^{\mathsf{T}}\right) (\boldsymbol{v}_j^{\mathrm{conv}})_c \\
&= \sum_{j=1}^{N} \sigma_j \left(\sum_{c=1}^{C} (\boldsymbol{q}_i)_c \left((\boldsymbol{H}^{\mathrm{K}^{\mathsf{T}}})_c (\boldsymbol{K}_j)_{:,c}\right)\right) (\boldsymbol{v}_j^{\mathrm{conv}})_c \\
&= \sum_{j=1}^{N} \sigma_j \left(\sum_{c=1}^{C}\sum_{m=1}^{M} (\boldsymbol{q}_i)_c (\boldsymbol{K}_j)_{m,c} (\boldsymbol{H}^{\mathrm{K}})_{m,c}\right) (\boldsymbol{v}_j^{\mathrm{conv}})_c \\
&= \sum_{j=1}^{N} \sigma_j \left(\mathrm{vec}\left(\mathrm{diag}(\boldsymbol{q}_i)\boldsymbol{K}_j\right) \mathrm{vec}\left(\boldsymbol{H}^{\mathrm{K}}\right)\right) ({\boldsymbol{H}^{\mathrm{V}}}^{\mathsf{T}})_c (\boldsymbol{V}_j)_{:,c}.
\end{aligned} \tag{20}$$

This reveals that the channel-wise convolution weights $\boldsymbol{H}^{\mathrm{K}}$ for the key projection, in fact, act as a *pattern detector* that extracts a single structural pattern from the channel-wise correlation, while those of $\boldsymbol{H}^{\mathrm{V}}$ perform as a *channel-wise context aggregator* that generates a spatial kernel weights for every channel. Despite the capability of capturing a channel-wise correlation structure, it still learns a single pattern only from the rich channel-wise correlation, thus being limited in leveraging diverse structural patterns for the attention weight generation compared to StructSA.

B. Additional Ablation Experiments

Here we provide additional ablation experiments to validate design components in StructSA. We follow the same training and testing protocols in Sec.4.3 of our main paper.

Channel-wise Correlation and Aggregation. Table 6a summarizes the effectiveness of the channel-wise correlation and aggregation. Compared to the dot-product correlation, the channel-wise correlation improves the top-1 accuracy by 0.3%p and 1.2%p on ImageNet-1K and Something-Something V1 datasets, respectively, validating that fine-grained structures from the channel-wise correlation are beneficial to the attention weight generation. As we use channel-wise context aggregator, we obtain additional improvements by 0.1%p and 0.5%p on both datasets.

channel-wise correlation	channel-wise aggregation	ImageNet-1K top-1	ImageNet-1K top-5	Something V1 top-1	Something V1 top-5
		80.7	95.1	48.7	77.2
✓		81.0	95.3	49.9	77.7
✓	✓	**81.1**	**95.4**	**50.4**	**78.2**

(a) **Channel-wise correlation and aggregation**.

M $\boldsymbol{U}^{\mathrm{K}}$	M $\boldsymbol{U}^{\mathrm{V}}$	ImageNet-1K top-1	ImageNet-1K top-5	Something V1 top-1	Something V1 top-5
-	-	80.5	95.0	48.3	76.6
3×3 ($\times3$)	3×3 ($\times3$)	<u>81.1</u>	<u>95.4</u>	50.4	**78.2**
1×1 ($\times1$)	3×3 ($\times3$)	80.8	95.1	48.7	77.1
5×5 ($\times5$)	3×3 ($\times3$)	81.0	95.3	**50.6**	78.1
7×7 ($\times7$)	3×3 ($\times3$)	80.9	95.1	50.3	78.1
3×3 ($\times3$)	1×1 ($\times1$)	80.9	95.2	49.6	77.5
3×3 ($\times3$)	5×5 ($\times5$)	**81.2**	**95.6**	50.3	**78.2**
3×3 ($\times3$)	7×7 ($\times7$)	81.0	95.4	50.3	77.8
1×1 ($\times1$)	1×1 ($\times1$)	80.6	95.0	48.5	76.9
5×5 ($\times5$)	5×5 ($\times5$)	<u>81.1</u>	<u>95.4</u>	<u>50.5</u>	**78.2**
7×7 ($\times7$)	7×7 ($\times7$)	81.0	95.2	<u>50.5</u>	78.1

(b) **Kernel size M.**

Table 6. **Ablation studies on ImageNet-1K and Something-Something V1**. Top-1 and top-5 accuracies (%) are shown. In Table 6a, we set $D = 4$, $M = 3\times3\times3$. In Table 6b, we set $D = 4$ as default. **Bold-faced** and <u>underlined</u> numbers indicate the first and second highest scores, respectively.

method (DeiT-S)	IN-1K param	IN-1K FLOPs	IN-1K top-1	IN-1K top-5	SS-V1 param	SS-V1 FLOPs	SS-V1 top-1	SS-V1 top-5
ConvSA	22.1M	4.6G	80.8	95.2	22.8M	57.3G	49.7	77.6
ConvSA + channel↑	27.9M	5.8G	80.9	95.2	34.3M	80.4G	49.9	77.9
ConvSA + layer↑	29.3M	6.1G	81.0	**95.4**	31.8M	80.8G	50.1	77.8
StructSA	22.4M	5.7G	**81.1**	**95.4**	23.1M	80.4G	**50.4**	**78.2**

Table 7. **Comparison to ConvSA variants with similar FLOPs.**

Different Combinations of $\boldsymbol{U}^{\mathrm{K}}$ and $\boldsymbol{U}^{\mathrm{V}}$. In Table 6b, we investigate different combinations of $\boldsymbol{U}^{\mathrm{K}}$ and $\boldsymbol{U}^{\mathrm{V}}$ varying the size of the kernel size M. As discussed in Sec.4.3, using the large kernel size M on both $\boldsymbol{U}^{\mathrm{K}}$ and $\boldsymbol{U}^{\mathrm{V}}$ improves the performance, demonstrating the effectiveness of SQKA and contextual aggregation. The performance saturates as M gets larger than $5\times5\times5$. We set the kernel size M of $\boldsymbol{U}^{\mathrm{K}}$ and $\boldsymbol{U}^{\mathrm{V}}$ to $3\times3\times3$ as default considering computation-accuracy trade-off.

Comparison to ConvSA. Table 7 compares our StructSA to its ConvSA counterpart with a matching capacity, *i.e.*, a similar number of parameters; we match their capacities by varying the number of channels or layers of the ConvSA backbone (DeiT-S). Our method achieves

better accuracy-compute trade-off on ImageNet-1K and Something-Something V1 datasets. For example, StructSA outperforms the ConvSA variants with more parameters and compute on Something-Something V1 where learning motion dynamics may be more important for classification.

C. Results on Dense Prediction Tasks

We evaluate the generalizability of StructViT on various dense prediction tasks: object detection and instance segmentation on COCO 2017 [10] as well as semantic segmentation on ADE20K [18]. For object detection and instance segmentation, we use the Mask R-CNN [4] with Hourglass UniFormer-$\{$S, B$\}_{h14}$ [8] as the backbone and then replace all SA blocks with our StructSA blocks. We train the models for 12 epochs following the $1\times$ schedule in [8]. Similarly, for semantic segmentation, we integrate StructSA blocks into Semantic FPN [7] with Hourglass UniFormer-$\{$S,B$\}_{h32}$ backbone and train the models for 80K iterations following the protocols in [13].

Tables 8 and 9 show consistent performance improvements across all benchmarks, affirming the effectiveness of StructSA. Specifically, in Table 8, StructViT-S-4-1$_{h14}$ outperforms the baseline UniFormer-S$_{h14}$ on both detection and segmentation tasks by 1.0 box mAP and 0.8 mask mAP, respectively. Furthermore, semantic segmentation results in Table 9 also shows the significant increase of mIOU over the baseline by 0.7 %p and 0.8 %p at both small and base scales, respectively. These consistent improvements effectively demonstrate the generalizability of StructSA across various backbone scales and downstream tasks.

D. Attention Map Visualization

We visualize attention maps of SA, ConvSA, and StructSA to provide an in-depth comparison across the methods. Different from SA, which uses individual query-key correlation as an attention weight for a single value feature (Fig. 4b), ConvSA and StructSA aggregate a local chunk of value features by generating dynamic kernels for each location. ConvSA generates the dynamic kernels $\boldsymbol{\kappa}_{i,j}^{\text{conv}}$, where spatial patterns are identical for all locations except for their scales (Fig. 4c). In contrast, StructSA constructs the dynamic kernels $\boldsymbol{\kappa}_{i,j}^{\text{struct}}$ in diverse aggregation patterns (Fig. 4e) by combining D correlation pattern scores and context aggregation patterns as explained in Sec.3.2. This property of StructSA enables the model to effectively leverage geometric structures for visual representation learning. To better observe the effect, we visualize the final attention maps of StructSA in Fig. 4f by spatially merging the overlapped kernels $\boldsymbol{\kappa}_{i,j}^{\text{struct}}$ following the equation:

$$c_{i,j}^{\text{struct}} = \sum_{m=0}^{M} (\boldsymbol{\kappa}_{i,j-\lfloor M/2 \rfloor+m}^{\text{struct}})_m. \tag{21}$$

method	#param (M)	Mask R-CNN 1×					
		AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
R50 [3]	44	38.0	58.6	41.4	34.4	55.1	36.7
PVT-M [13]	44	40.4	62.9	43.8	37.8	60.1	40.3
Focal-T [17]	49	44.8	67.7	49.2	41.0	64.7	44.2
PVTv2-B2 [14]	45	45.3	67.1	49.6	41.2	64.2	44.4
UniFormer-S$_{h14}$ [8]	41	45.6	68.1	49.7	41.6	64.8	45.0
StructViT-S-4-1$_{h14}$	42	**46.6**	**69.2**	**51.5**	**42.8**	**65.5**	**46.1**
R101 [3]	63	40.4	61.1	44.2	36.4	57.7	38.8
X101-32 [16]	63	41.9	62.5	45.9	37.5	59.4	40.2
PVT-M [13]	64	42.0	64.4	45.6	39.0	61.6	42.1
PVT-L [13]	81	42.9	65.0	46.6	39.5	61.9	42.5
Twins-B [1]	76	45.2	67.6	49.3	41.5	64.5	44.8
Swin-S [11]	69	44.8	66.6	48.9	40.9	63.8	44.2
Swin-B [11]	107	46.9	-	-	42.3	-	-
Focal-S [17]	71	47.4	69.8	51.9	42.8	66.6	46.1
Focal-B [17]	110	47.8	-	-	43.2	-	-
PVTv2-B5 [14]	101	47.4	68.6	51.9	42.5	65.7	6.0
UniFormer-B$_{h14}$ [8]	69	47.4	69.7	52.1	43.1	66.0	46.5
StructViT-B-4-1$_{h14}$	70	**48.2**	**70.8**	**53.0**	**43.7**	**66.7**	**46.9**

Table 8. **Results of object detection, instance segmentation on COCO val2017**. AP^b and AP^m indicates box mAP and mask mAP, respectively. We measure FLOPs at 800×1280 resolution.

method	Semantic FPN 80K		
	#param (M)	FLOPs (G)	mIoU (%)
Res101 [3]	48	260	38.8
PVT-M [13]	48	219	41.6
PVT-L [13]	65	283	42.1
Swin-S [11]	53	274	45.2
Twins-B [1]	60	261	45.3
TwinsP-L [1]	65	283	46.4
UniFormer-S$_{h32}$ [8]	25	199	46.2
UniFormer-S [8]	25	247	46.6
StructViT-S-4-1$_{h32}$	26	271	**46.9**
X101-32x4d [16]	86	-	40.2
Swin-B [11]	91	422	46.0
Twins-L [1]	104	404	46.7
UniFormer-B$_{h32}$ [8]	54	350	47.7
UniFormer-B [8]	54	471	48.0
StructViT-B-4-1$_{h32}$	54	529	**48.5**

Table 9. **Results of semantic segmentation on ADE20K.** We measure FLOPs using 512×2048 resolution images.

$c_{i,j}^{\text{struct}}$ indicates the final attention score multiplied to the value $\boldsymbol{v}_j$ to generate the output. The examples in Fig. 4f show that StructSA contextualizes the entire features in a structure-aware manner considering objects' layouts or shapes; for instance, StructSA aggregates global contexts distinguishing different parts of an orange (Fig. 4f, 2nd row) or an ostrich (Fig. 4f, 3rd row). The qualitative analysis demonstrates that StructSA outperforms ConvSA in leveraging correlation structures for visual representation learning. This suggests that StructSA may be particularly useful for computer vision tasks that require an understanding of relational structures and layouts of visual elements.

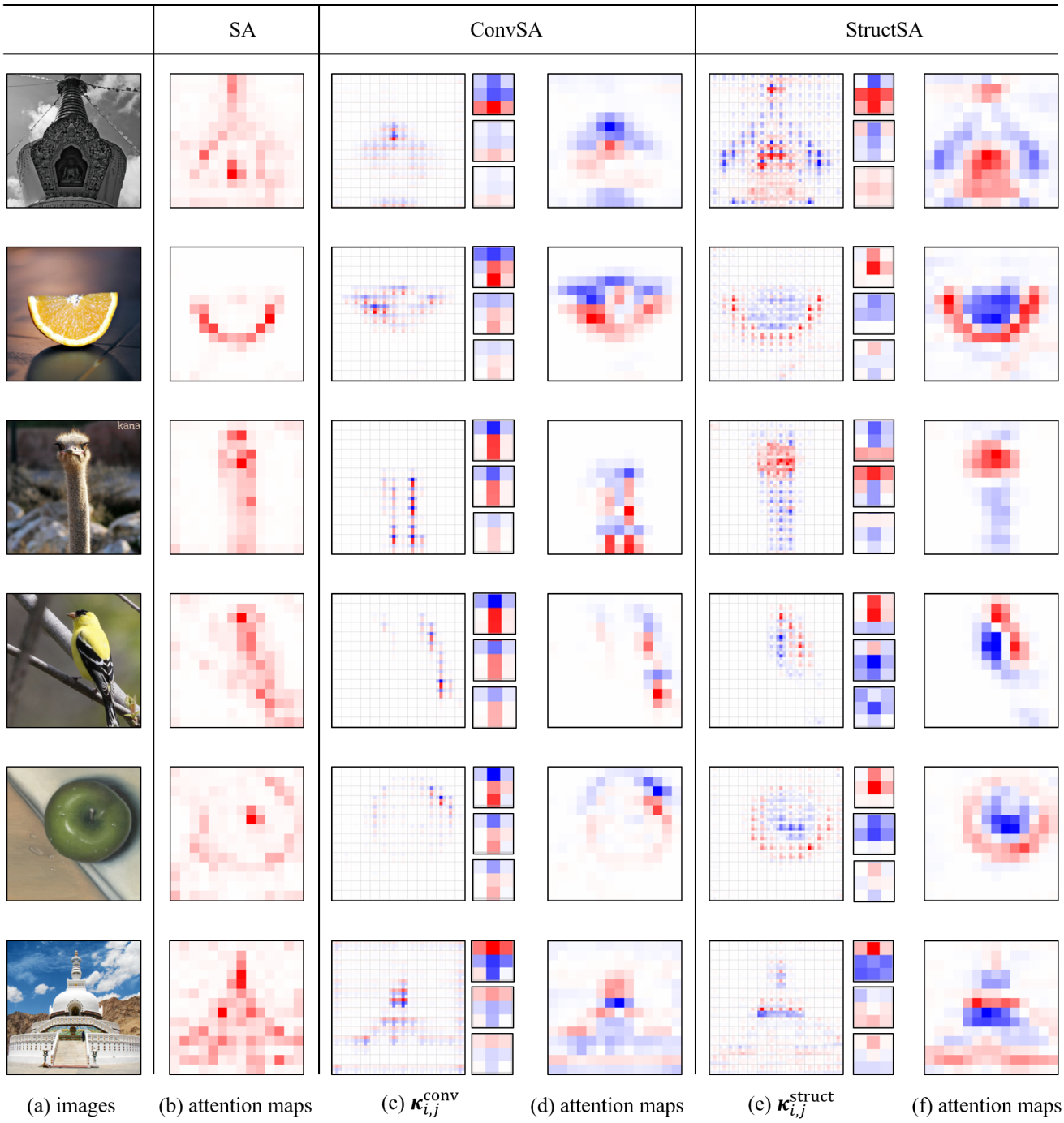

Figure 4. **Attention map visualization of SA, ConvSA, and StructSA on ImageNet-1K.** The query location i is set to the center of the image and the kernel size $M = 3 \times 3$. Given (a) input images, we illustrate (b) attention maps of SA, (c) dynamic kernels $\boldsymbol{\kappa}_{i,j}^{\text{conv}}$, (d) final attention maps of ConvSA, *i.e.*, aggregated weights of $\boldsymbol{\kappa}_{i,j}^{\text{conv}}$, (e) dynamic kernels $\boldsymbol{\kappa}_{i,j}^{\text{struct}}$, and (f) final attention maps of StructSA, *i.e.*, aggregated weights of $\boldsymbol{\kappa}_{i,j}^{\text{struct}}$, respectively. Note that in (c) and (e), each location j has an aggregation map of the kernel size $M = 3 \times 3$ and thus we show enlarged images for three different sampled locations j.

References

[1] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in neural information processing*

systems, 34:9355–9366, 2021. 3

[2] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *arXiv preprint arXiv:2104.11227*, 2021. 2

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 3

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3

[5] Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling. *arXiv preprint arXiv:1610.04325*, 2016. 1

[6] Manjin Kim, Heeseung Kwon, Chunyu Wang, Suha Kwak, and Minsu Cho. Relational self-attention: What's missing in attention for video understanding. *NeurIPS*, 34:8046–8059, 2021. 1

[7] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6399–6408, 2019. 3

[8] Kunchang Li, Yali Wang, Junhao Zhang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unifying convolution and self-attention for visual recognition. *arXiv preprint arXiv:2201.09450*, 2022. 3

[9] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *CVPR*, pages 4804–4814, 2022. 2

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 3

[11] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 3

[12] Hamed Pirsiavash, Deva Ramanan, and Charless Fowlkes. Bilinear classifiers for visual recognition. *NeurIPS*, 22, 2009. 1

[13] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, pages 568–578, 2021. 3

[14] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 3

[15] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *ICCV*, pages 22–31, 2021. 2

[16] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 3

[17] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *arXiv preprint arXiv:2107.00641*, 2021. 3

[18] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019. 3