SHViT: Single-Head Vision Transformer with Memory Efficient Macro Design —— Supplementary Material ——

This supplementary material presents additional comparison results, memory analysis, detection results, and experimental settings.

A. Comparison with Tiny Variants of Large-scale Models

We compare our model against tiny variants of established models in Tab. 1. Our model, when applied to higher resolutions, outperforms state-of-the-art models in terms of parameter and throughput. Compared to Swin-T [11], our SHViT-S4$_{r384}$ is 0.3% inferior in accuracy but is 2.3× / 9.5× faster on the A100 GPU / Intel CPU.

In Fig. 1, we also provide further results of Section 3.2. It demonstrates improved speed performance when increasing the resolution not only on mobile devices but also on other inference platforms compared to the recent models [8, 20]. This result showcases that our model can be a competitive alternative in real-world applications. Further analysis of these performance enhancements will be detailed in the following section.

B. Memory Efficiency Analysis

Our model has a larger number of parameters compared to lightweight models. For instance, SHViT-S3 has 2.7× more parameters compared to EfficientNet-B0 [17]. However, an important consideration for deploying the model on resource-constrained devices is the memory access cost of the feature maps. On an I/O bound devices, the number of memory access for a given layer is as follows

$$2 \times b \times h \times w \times c + k^2 \times c^2 \quad (1)$$

Particularly, when increasing batch size to enhance throughput, or for applications that require high-resolution input, the impact of the first term in the above equation becomes significantly more critical. Our proposed macro and micro designs considerably reduce memory usage by eliminating redundancies in the first term's $h \times w$ and c components, respectively. In Tab. 2, our model, despite having more parameters than EfficientNet-B0, consumes less test memory. Notably, the disparity in memory usage grows with increasing batch sizes.

C. Further Results on COCO Detection

We also present results on COCO object detection benchmark [9] with DEtection TRansformer (DETR) [2,23] framework in Tab. 3. The encoder of DETR consists of self-attention and FFN, and the decoder consists of self-attention, cross-attention, and FFN. To demonstrate the efficacy of our single-head attention module not only as a feature extractor but also as a detection head, we apply single-head design to the encdoer's self-attention and decoder's cross-attention layers. These two layers involve significant computational costs, thus employing a single-head design can greatly enhance the model speed. However, in the detection head, each of the attention weights localizes different extremities [14], making it challenging to simply com-

Model	Params (M)	FLOPs (G)	Throughput (image/s) GPU	CPU$_{\text{ONNX}}$	Top-1 (%)
CaiT-XXS36 [19]	17	3.8	1394	24	79.1
Twins-PCPVT-S [4]	24	3.8	3800	53	81.2
Swin-T [11]	28	4.5	2868	33	81.3
TNT-S [7]	24	5.2	1554	37	81.5
CoAtNet-0 [5]	25	4.2	2448	53	81.6
DeiT-B [18]	87	17.6	3227	21	81.8
XCiT-S12 [1]	26	4.8	3110	-	82.0
PVTv2-B2 [21]	25	4.0	2924	14	82.0
FocalNet-T [22]	28	4.4	2808	68	82.1
ConvNeXt-T [12]	29	4.5	3325	49	82.1
SHViT-S4	17	1.0	14283	509	79.4
SHViT-S4$_{r384}$	17	2.2	6702	315	81.0
SHViT-S4$_{r512}$	17	4.0	3957	198	82.0

Table 1. Comparison with the tiny variants of state-of-the-art large-scale models on ImageNet-1K classification. 'r384' means fine-tuned at 384×384 resolution. Models which could not be reliably converted to ONNX format are annotated by '—'.

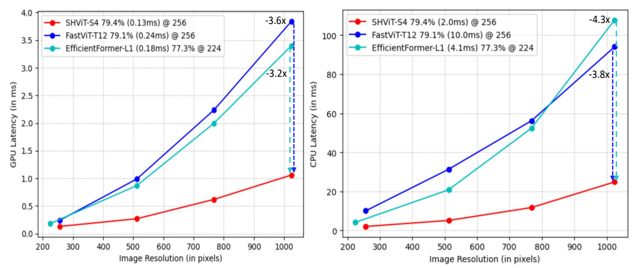

Figure 1. GPU, CPU latency comparison of a SHViT-S4 with recent state-of-the-art FastViT [20] and EfficientFormer [8]; measured on A100 GPU, Intel CPU for various image resolutions.

Model	Top-1 (%)	Params (M)	Inference Memory (MB) / Throughput (images/s) bs1	bs32	bs256	bs1024
SHViT-S3	77.4	14.2	1855 / 147	1963 / 4691	2613 / 20522	5525 / 22309
EfficientNet-B0	77.1	5.3	1931 / 175	2015 / 5427	3861 / 8433	10493 / 8706

Table 2. Memory Consumption Comparison with EfficientNet-B0 [17]. 'bs32' indicates that test time memory consumption and throughput are measured at batch size of 32.

Method	Params	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Deformable DETR w/ single-head	37.1M	31.4 (24% ↑)	43.1	62.7	46.6	26.3	46.6	57.2
Deformable DETR	40.0M	25.4	43.8	62.6	47.7	26.4	47.1	58.0

Table 3. Effectiveness of our Single-Head Attention module with Deformable DETR [23] framework. Our method improves test speed by 24% without significant performance degradation.

bine them into a single-head design. Furthermore, we find that the multi-head design in both the initial layer and latter layers of the encoder/decoder is vital. Thus, we employ single-head attention modules in the 2nd, 3rd, and 4th layers of each encoder/decoder. To minimize performance degradation, we also increase the head dimension in the single-head module from 32 to 64. We train our model using the training recipe of Deformable DETR [2, 23]. As shown in Tab. 3, single-head module demonstrates reasonable performance as a detector head and is a competitive alternative for applications where inference speed is crucial.

D. More Details on Redundancy Experiments

In this section, we provide implementation details of section 2.2.

head similarity analysis. For each layer i, the average cosine similarity value is computed as:

$$HeadSim_i = \frac{1}{N_h(N_h-1)} \sum_{j \neq k} cos(head_j, head_k) \quad (2)$$

where N_h is the number of heads. Then, the value is averaged for all batches.

head ablation study. In order to perform head ablation experiments, we modify the formula for Multi-Head Self-Attention (MHSA):

$$\text{MHSA} = \text{Concat}(\delta_1 \text{head}_1, ..., \delta_N \text{head}_N)W^O, \quad (3)$$

$$\text{head}_i = \text{Attention}(\mathbf{X}_i W_i^Q, \mathbf{X}_i W_i^K, \mathbf{X}_i W_i^V), \quad (4)$$

$$\text{Attention}(\mathbf{q}, \mathbf{k}, \mathbf{v}) = \text{Softmax}(\mathbf{q}\mathbf{k}^\intercal / \sqrt{d_{head}})\mathbf{v}, \quad (5)$$

where the δ are mask vaiables with values in $\{0, 1\}$. When all δ are equal to 1, the above layer is equivalent to the MHSA layer. In order to ablate head i, we simply set $\delta_i = 0$. We conduct experiments by selectively removing one or more attention heads from a given architecture during test time and assessing the resulting impact on accuracy. *And we report the best accuracy for each layer in the model, i.e. the accuracy achieved by reducing the entire layer to the single most important head.*

We further investigate head redundancy in DeiT-S-Distill [18], a vision transformer distilled with knowledge from ConvNets. In the distilled model, we can also observe a significant computational redundancy among many heads in the latter stages. Additionally, in the early stages, where

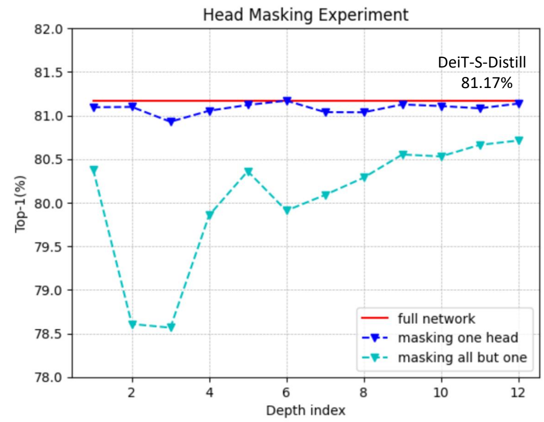

Figure 2. Head ablation study on DeiT-Small-Distill [18].

many heads operate similarly to convolution, there is a relatively substantial decline in performance.

E. Further Discussions on Related Works

About Macro Design. Our patch embedding scheme is similar to that of [6, 10], but the derivation process takes place from a completely different perspective. While [6] indirectly determines the patch embedding size through experiment grafting ResNet and DeiT, our work, on the other hand, investigate redundancy from the beginning, analyzing it separately in terms of spatial and channel. This allows us to address not only the spatial redundancy in traditional patch embedding but also propose a SHSA module, in contrast to [6] which employs MHSA (at mobile, SHViT-S4 80.2%/**1.6**ms *vs.* LeViT-192 80.0%/**28.0**ms). *To the best of our knowledge, none of existing works have analyzed the effects (speed, memory efficiency) of resolving spatial redundancy in diverse environments (devices, tasks).*

About Partial Design in SHSA. Partial channel design has also been employed in previous research [3, 13]. However, our work is distinct in both motivation and effectiveness. While prior work primarily focused on FLOPs (or throughput) and so employs convolutions (either depthwise or vanilla) on partial channels, this paper addresses multi-head redundancy by employing attention with single-head on partial channels. Furthermore, our SHSA, with preceding convolution, memory-efficiently leverages two complementary features in parallel within a single token mixer [15, 16].

References

[1] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *Advances in neural information processing systems*, 34:20014–20027, 2021. 1

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1, 2

[3] Jierun Chen, Shiu-hong Kao, Hao He, Weipeng Zhuo, Song Wen, Chul-Ho Lee, and S.-H. Gary Chan. Run, don't walk: Chasing higher flops for faster neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12021–12031, June 2023. 2

[4] 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. 1

[5] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in neural information processing systems*, 34:3965–3977, 2021. 1

[6] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12259–12269, 2021. 2

[7] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34:15908–15919, 2021. 1

[8] Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 35:12934–12949, 2022. 1

[9] 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. 1

[10] Xinyu Liu, Houwen Peng, Ningxin Zheng, Yuqing Yang, Han Hu, and Yixuan Yuan. Efficientvit: Memory efficient vision transformer with cascaded group attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14420–14430, 2023. 2

[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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 1

[12] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 1

[13] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018. 2

[14] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3651–3660, 2021. 1

[15] Namuk Park and Songkuk Kim. How do vision transformers work? In *International Conference on Learning Representations*, 2022. 2

[16] Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, and Shuicheng YAN. Inception transformer. In *Advances in Neural Information Processing Systems*, 2022. 2

[17] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 1

[18] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 1, 2

[19] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 32–42, October 2021. 1

[20] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. Fastvit: A fast hybrid vision transformer using structural reparameterization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 1

[21] 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. 1

[22] Jianwei Yang, Chunyuan Li, Xiyang Dai, and Jianfeng Gao. Focal modulation networks. *Advances in Neural Information Processing Systems*, 35:4203–4217, 2022. 1

[23] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1, 2