Appendix

A. Detailed Architecture Specifications

Tab. 3 provides a detailed overview of the architecture specifications for all models, with an assumed input image size of 224×224. The stem of the model is denoted as ”conv $n \times n$, 64-d, BN; conv 2×2, 64-d, LN”, representing two convolution layers with a stride of 2 to obtain a more informative token sequence with a length of $\frac{H}{4} \times \frac{W}{4}$. Here, ”BN” and ”LN” indicate Batch Normalization and Layer Normalization [1], respectively, while ”64-d” denotes the convolution layer with an output dimension of 64. The multi-head mixed convolution module with 4 heads (conv 3×3, conv 5×5, conv 7×7, conv 9×9) is denoted as ”sam. head. 4”, while ”msa. head. 8” represents the multi-head self-attention module with 8 heads. Additionally, ”sam. ep_r. 2” indicates a Scale-Aware Aggregation module with twice as much expanding ratio.

B. Detailed Experimental Settings

B.1. Image classification on ImageNet-1K

We trained all models on the ImageNet-1K dataset [5] for 300 epochs, using an image size of 224×224 . Following Swin [10], we utilized a standardized set of data augmentations [4], including Random Augmentation, Mixup [23], CutMix [22], and Random Erasing [26]. To regularize our models, we applied Label Smoothing [16] and DropPath [7] techniques. The initial learning rate for all models was set to 2×10^{-3} after 5 warm-up epochs, beginning with a rate of 1×10^{-6}. To optimize our models, we employed the AdamW [11] algorithm and a cosine learning rate scheduler [12]. The weight decay was set to 0.05 and the gradient clipping norm to 5.0. For our mini, tiny, small, base, and large models, we used stochastic depth drop rates of 0.1, 0.1, 0.2, 0.3, and 0.5, respectively. For more details, please refer to the Tab. 1 provided.

B.2. Image classification pretrained on ImageNet-22K

We trained the SMT-L model for 90 epochs using a batch size of 4096 and an input resolution of 224×224. The initial learning rate was set to 1×10^{-3} after a warm-up period of 5 epochs. The stochastic depth drop rates were set to 0.1. Following pretraining, we performed fine-tuning on the ImageNet-1K dataset for 30 epochs. The initial learning rate was set to 2×10^{-5}, and we utilized a cosine learning rate scheduler and AdamW optimizer. The stochastic depth drop rate remained at 0.1 during fine-tuning, while both CutMix and Mixup augmentation techniques were disabled.

config	value
optimizer	AdamW
LR	2e-3
weight decay	0.05
optimizer momentum	β_1, β_2=0.9, 0.999
batch size	1024
LR schedule	cosine
minimum learning rate	1e-5
warmup epochs	5
warmup learning rate	1e-6
training epochs	300
augmentation	rand-m9-mstd0.5-inc1
color jitter	0.4
mixup α	0.2
cutmix α	1.0
random erasing	0.25
label smoothing	0.1
gradient clip	5.0
drop path	[0.1, 0,1, 0,2, 0,3, 0.5] (M,T,S,B,L)

Table 1: Image Classification Training Settings

B.3. Object Detection and Instance Segmentation

In transferring SMT to object detection and instance segmentation on COCO [9], we have considered six common frameworks: Mask R-CNN [6], Cascade Mask RCNN [2], RetinaNet [8], Sparse R-CNN [15], ATSS [25], and DINO [24]. For DINO, the model is fine-tuned for 12 epochs, utilizing 4 scale features. For optimization, we adopt the AdamW optimizer with an initial learning rate of 0.0002 and a batch size of 16. When training models of different sizes, we adjust the training settings according to the settings used in image classification. The detailed hyper-parameters used in training models are presented in Tab. 2.

config	value
optimizer	AdamW
LR	0.0002
weight decay	0.05
optimizer momentum	β_1, β_2=0.9, 0.999
batch size	16
LR schedule	steps:[8, 11] (1×), [27, 33] (3×)
warmup iterations (ratio)	500 (0.001)
training epochs	12 (1×), 36 (3×)
scales	(800, 1333) (1×), Multi-scales [10] (3×)
drop path	0.2 (Small), 0.3 (Base)

Table 2: Object Detection and Instance Segmentation Training Settings

B.4. Semantic Segmentation

For ADE20K, we utilized the AdamW optimizer with an initial learning rate of 0.00006, a weight decay of 0.01,

	downsp. rate (output size)	Layer Name	SAM-M	SAM-T	SAM-S	SAM-B	SAM-L
stage 1	4× (56×56)	SAM Block	conv 3×3, 64-d, BN conv 2×2, 64-d, LN	conv 3×3, 64-d, BN conv 2×2, 64-d, LN	conv 7×7, 64-d, BN conv 2×2, 64-d, LN	conv 7×7, 64-d, BN conv 2×2, 64-d, LN	conv 7×7, 96-d, BN conv 2×2, 96-d, LN
			[dim 64 sam.head. 4 sam.ep_r. 2] × 1	[dim 64 sam.head. 4 sam.ep_r. 2] × 2	[dim 64 sam.head. 4 sam.ep_r. 2] × 3	[dim 64 sam.head. 4 sam.ep_r. 2] × 4	[dim 96 sam.head. 4 sam.ep_r. 2] × 4
stage 2	8× (28×28)	SAM Block	conv 3×3, 128-d, LN	conv 3×3, 128-d, LN	conv 3×3, 128-d, LN	conv 3×3, 128-d, LN	conv 3×3, 192-d, LN
			[dim 128 sam.head. 4 sam.ep_r. 2] × 1	[dim 128 sam.head. 4 sam.ep_r. 2] × 2	[dim 128 sam.head. 4 sam.ep_r. 2] × 4	[dim 128 sam.head. 4 sam.ep_r. 2] × 6	[dim 192 sam.head. 4 sam.ep_r. 2] × 6
stage 3	16× (14×14)	Mix Block	conv 3×3, 256-d , LN	conv 3×3, 256-d , LN	conv 3×3, 256-d , LN	conv 3×3, 256-d , LN	conv 3×3, 384-d , LN
			[dim 256 sam.head. 4 sam.ep_r. 2 msa.head. 8] × 4	[dim 256 sam.head. 4 sam.ep_r. 2 msa.head. 8] × 8	[dim 256 sam.head. 4 sam.ep_r. 2 msa.head. 8] × 18	[dim 256 sam.head. 4 sam.ep_r. 2 msa.head. 8] × 28	[dim 384 sam.head. 4 sam.ep_r. 2 msa.head. 8] × 28
stage 4	32× (7×7)	MSA Block	conv 3×3, 512-d , LN	conv 3×3, 512-d , LN	conv 3×3, 512-d , LN	conv 3×3, 512-d , LN	conv 3×3, 768-d , LN
			[dim 512 msa.head 16] × 1	[dim 512 msa.head 16] × 1	[dim 512 msa.head 16] × 1	[dim 512 msa.head 16] × 2	[dim 768 msa.head 16] × 3

Table 3: Detailed architecture specifications at four stages for SMT.

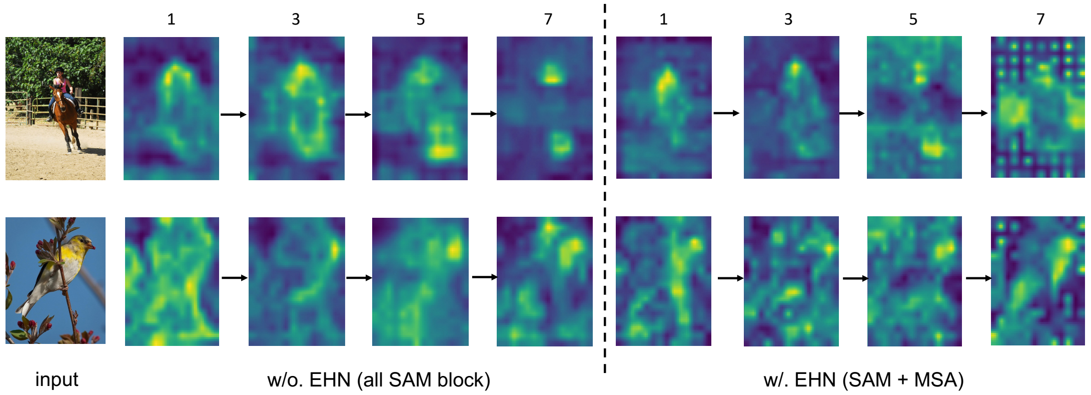

Figure 1: Visualization of modulation values at the penultimate stage for two variants of SMT. **(Left: w/o. EHN)** Stacking of SAM blocks exclusively in the penultimate stage. **(Right: w/. EHN)** The utilization of an evolutionary hybrid stacking strategy, wherein one SAM block and one MSA are successively stacked.

and a batch size of 16 for all models trained for 160K iterations. In terms of testing, we reported the results using both single-scale (SS) and multi-scale (MS) testing in the main comparisons. For multi-scale testing, we experimented with resolutions ranging from 0.5 to 1.75 times that of the training resolution. To set the path drop rates in different models, we used the same hyper-parameters as those used for object detection and instance segmentation.

C. More Experiments

C.1. More Variants of SMT

This section demonstrates how we scaled our SMT to create both smaller (SMT-M) and larger (SMT-L) models. Their detailed architecture settings are provided in Tab. 3, along with previous variants. We then evaluated their performance on the ImageNet-1K dataset.

As shown in Tab. 4, SMT-M achieves competitive results with a top-1 accuracy of 78.4%, despite having only 6.5M parameters and 1.3 GFLOPs of computation. On the other side, SMT-L shows an example to scale our SMT to larger models, which outperforms other state-of-the-art networks with similar parameters and computation costs, achieving a top-1 accuracy of 84.6%. These results confirm the strong scalability of the SMT architecture, which can be applied to create models of varying sizes, demonstrating its immense potential.

D. Additional Network Analysis

In Fig. 1, we present the learned scale-aware modulation (SAM) value maps in two variants of SMT-T: evolutionary SMT, which employs an evolutionary hybrid stacking strategy, and general SMT, which only employs SAM in the

method	image size	#param.	FLOPs	ImageNet top-1 acc.
RegNetY-4G [13]	224^2	21M	4.0G	80.0
RegNetY-8G [13]	224^2	39M	8.0G	81.7
RegNetY-16G [13]	224^2	84M	16.0G	82.9
EffNet-B3 [17]	300^2	12M	1.8G	81.6
EffNet-B4 [17]	380^2	39M	4.2G	82.9
EffNet-B5 [17]	456^2	30M	9.9G	83.6
EffNet-B6 [17]	528^2	43M	19.0G	84.0
PVT-T [19]	224^2	13M	1.8G	75.1
PVT-S [19]	224^2	25M	3.8G	79.8
PVT-M [19]	224^2	44M	6.7G	81.2
PVT-L [19]	224^2	61M	9.8G	81.7
Swin-T [10]	224^2	29M	4.5G	81.3
Swin-S [10]	224^2	49.6M	8.7G	83.0
Swin-B [10]	224^2	87.8M	15.4G	83.4
Twins-S [3]	224^2	24M	2.9G	81.7
Twins-B [3]	224^2	56M	8.6G	83.2
Focal-T [21]	224^2	29M	4.9G	82.2
Focal-B [21]	224^2	89.8M	16.4G	83.8
Shunted-T [14]	224^2	11.5M	2.1G	79.8
Shunted-S [14]	224^2	22.4M	4.9G	82.9
Shunted-B [14]	224^2	39.6M	8.1G	84.0
FocalNet-T [20]	224^2	28.6M	4.5G	82.3
FocalNet-S [20]	224^2	50.3M	8.7G	83.5
FocalNet-B [20]	224^2	88.7M	15.4G	83.9
MaxViT-T [18]	224^2	31M	5.6G	83.6
MaxViT-S [18]	224^2	69M	11.7G	84.5
MaxViT-B [18]	224^2	120M	23.4G	84.9
SMT-M	224^2	6.5M	1.3G	**78.4**
SMT-T	224^2	11.5M	2.4G	**82.2**
SMT-S	224^2	20.5M	4.7G	**83.7**
SMT-B	224^2	32.0M	7.7G	**84.3**
SMT-L	224^2	80.5M	17.7G	**84.6**

Table 4: Comparison of different backbones on ImageNet-1K classification.

penultimate stage. In evolutionary SMT-T, comprising a total of 8 layers in the penultimate stage, we select the layers ($[1, 3, 5, 7]$) containing SAM block and compare them with the corresponding layers in general SMT. Through visualization, we can observe some noteworthy patterns. In general SMT, the model primarily concentrates on local details in the shallow layers and on semantic information in the deeper layers. However, in evolutionary SMT, the focus region does not significantly shift as the network depth increases. Furthermore, it captures local details more effectively than general SMT in the shallow layers, while preserving detailed and semantic information about the target object at deeper layers. These results indicate that our evolutionary hybrid stacking strategy facilitates SAM blocks in capturing multi-granularity features while allowing multi-head self-attention (MSA) blocks to concentrate on capturing global semantic information. Accordingly, each block within each layer is more aptly tailored to its computational characteristics, leading to enhanced performance in diverse visual tasks.

E. Additional Visual Examples

We present supplementary visualization of modulation value maps within our SMT. Specifically, we randomly select validation images from the ImageNet-1K dataset and generate visual maps for modulation at different stages, as illustrated in Fig 2. The visualizations reveal that the scale-aware modulation is critical in strengthening semantically relevant low-frequency signals and accurately localizing the most discriminative regions within images. By exploiting this robust object localization capability, we can allocate more effort towards modulating these regions, resulting in more precise predictions. We firmly believe that both our multi-head mixed convolution module and scale-aware aggregation module have the potential to further enhance the modulation mechanism.

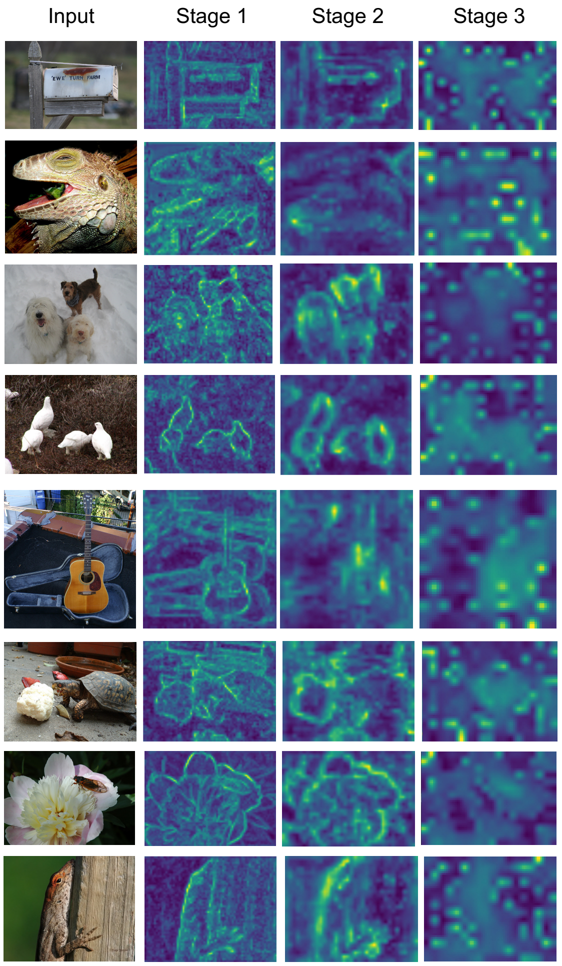

Figure 2: Visualization of modulation value maps at the top three stages.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *Advances in neural information processing systems*, 2016.

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[3] 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.

[4] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] 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.

[7] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 646–661. Springer, 2016.

[8] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[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 *European conference on computer vision*, pages 740–755. Springer, 2014.

[10] 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.

[11] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

[12] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*.

[13] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10428–10436, 2020.

[14] Sucheng Ren, Daquan Zhou, Shengfeng He, Jiashi Feng, and Xinchao Wang. Shunted self-attention via multi-scale token aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10853–10862, 2022.

[15] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14454–14463, 2021.

[16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[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.

[18] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *European conference on computer vision*, pages 459–479. Springer, 2022.

[19] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021.

[20] Jianwei Yang, Chunyuan Li, Xiyang Dai, and Jianfeng Gao. Focal modulation networks. In *Advances in Neural Information Processing Systems*.

[21] 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. *Advances in Neural Information Processing Systems*, 2021.

[22] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[23] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[24] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *The Eleventh International Conference on Learning Representations*, 2022.

[25] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9759–9768, 2020.

[26] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.