RMT: Retentive Networks Meet Vision Transformers

Supplementary Material

A. Architecture Details

Our architectures are illustrated in the Tab. 1. For convolution stem, we apply five 3×3 convolutions to embed the image into 56×56 tokens. GELU and batch normalization are used after each convolution except the last one, which is only followed by batch normalization. 3×3 convolutions with stride 2 are used between stages to reduce the feature map's resolution. 3×3 depth-wise convolutions are adopted in CPE. Moreover, 5×5 depth-wise convolutions are adopted in LCE. RMT-DeiT-S, RMT-Swin-T, and RMT-Swin-S are models that we used in our ablation experiments. Their structures closely align with the structure of DeiT [20] and Swin-Transformer [16] without using techniques like convolution stem, CPE, and others.

B. Experimental Settings

ImageNet Image Classification. We adopt the same training strategy with DeiT [20] with the only supervision is the classification loss. In particular, our models are trained from scratch for 300 epochs. We use the AdamW optimizer with a cosine decay learning rate scheduler and 5 epochs of linear warm-up. The initial learning rate, weight decay, and batch size are set to 0.001, 0.05, and 1024, respectively. Our augmentation settings are RandAugment [4] (randm9-mstd0.5-inc1), Mixup [26] (prob=0.8), CutMix [25] (probe=1.0), Random Erasing [27] (prob=0.25) and Exponential Moving Average (EMA) [17]. The maximum rates of increasing stochastic depth [10] are set to 0.1/0.15/0.4/0.5 for RMT-T/S/B/L, respectively. For a more comprehensive comparison, we train two versions of the model. The first version uses only classification loss as the supervision, while the second version, in addition to the classification loss, incorporates token labeling introduced by [11] for additional supervision. Models using token labeling are marked with"*".

COCO Object Detection and Instance Segmentation. We apply RetinaNet [14], Mask-RCNN [9] and Cascaded Mask-CNN [1] as the detection frameworks to conduct experiments. We implement them based on the MMDetection [2]. All models are trained under two common settings:"1×" (12 epochs for training) and"3×+MS" (36 epochs with multi-scale augmentation for training). For the "1×" setting, images are resized to the shorter side of 800 pixels. For the "3×+MS", we use the multi-scale training strategy and randomly resize the shorter side between 480 to 800 pixels. We apply AdamW optimizer with the initial learning rate of 1e-4. For RetinaNet, we use the weight decay of 1e-4 for RetinaNet while we set it to 5e-2 for Mask-RCNN and Cascaded Mask-RCNN. For all settings, we use the batch size of 16, which follows the previous works [16, 23, 24]

ADE20K Semantic Segmentation. Based on MMSegmentation [3], we implement UperNet [22] and SemanticFPN [12] to validate our models. For UperNet, we follow the previous setting of Swin-Transformer [16] and train the model for 160k iterations with the input size of 512×512. For SemanticFPN, we also use the input resolution of 512×512 but train the models for 80k iterations.

C. Finetuning on larger resolution.

To align the model's receptive field across resolutions, we adjust $\gamma' = \gamma^{res_{ori}/res_{new}}$. Using native γ, the model achieves 84.9%. With adjusted $\gamma' = \gamma^{224/384}$, it achieves 85.2% (Tab. 3).

D. Efficiency Comparison

We compare the inference speed of RMT with other backbones, as shown in Tab. 2. Our models achieve the best trade-off between speed and accuracy among many competitors.

E. Details of Explicit Decay

We use different γ for each head of the multi-head ReSA to control the receptive field of each head, enabling the ReSA to perceive multi-scale information. We keep all the γ of ReSA's heads within a certain range. Assuming the given receptive field control interval of a specific ReSA module is $[a, b]$, where both a and b are positive real numbers. And the total number of the ReSA module's heads is N. The γ for its ith head can be written as Eq. 1:

$$\gamma_i = 1 - 2^{-a-\frac{(b-a)i}{N}} \tag{1}$$

For different stages of different backbones, we use different values of a and b, with the details shown in Tab. 4.

Model	Blocks	Channels	Heads	Ratios	Params(M)	FLOPs(G)
RMT-T	[2, 2, 8, 2]	[64, 128, 256, 512]	[4, 4, 8, 16]	[3, 3, 3, 3]	14	2.5
RMT-S	[3, 4, 18, 4]	[64, 128, 256, 512]	[4, 4, 8, 16]	[4, 4, 3, 3]	27	4.5
RMT-B	[4, 8, 25, 8]	[80, 160, 320, 512]	[5, 5, 10, 16]	[4, 4, 3, 3]	54	9.7
RMT-L	[4, 8, 25, 8]	[112, 224, 448, 640]	[7, 7, 14, 20]	[4, 4, 3, 3]	95	18.2
RMT-DeiT-S	[12]	[384]	[6]	[4]	22	4.6
RMT-Swin-T	[2, 2, 6, 2]	[96, 192, 384, 768]	[3, 6, 12, 24]	[4, 4, 4, 4]	29	4.7
RMT-Swin-S	[2, 2, 18, 2]	[96, 192, 384, 768]	[3, 6, 12, 24]	[4, 4, 4, 4]	50	9.1

Table 1. Detailed Architectures of our models.

Model	Params (M)	FLOPs (G)	Troughput (imgs/s)	Top1 (%)
MPViT-XS [13]	11	2.9	1496	80.9
Swin-T [16]	29	4.5	1704	81.3
BiFormer-T [28]	13	2.2	1602	81.4
GC-ViT-XT [8]	20	2.6	1308	82.0
SMT-T [15]	12	2.4	636	82.2
RMT-T	14	2.5	1650	82.4
Focal-T [23]	29	4.9	582	82.2
CSWin-T [5]	22	4.3	1561	82.7
Eff-B4 [19]	19	4.2	627	82.9
MPViT-S [13]	23	4.7	986	83.0
Swin-S [16]	50	8.8	1006	83.0
SGFormer-S [6]	23	4.8	952	83.2
iFormer-S [18]	20	4.8	1051	83.4
CMT-S [7]	25	4.0	848	83.5
RMT-Swin-T	29	4.7	1192	83.6
CSwin-S [5]	35	6.9	972	83.6
MaxViT-T [21]	31	5.6	826	83.6
SMT-S [15]	20	4.8	356	83.7
BiFormer-S [28]	26	4.5	766	83.8
RMT-S	27	4.5	876	84.1

Model	Params (M)	FLOPs (G)	Troughput (imgs/s)	Top1 (%)
Focal-S [23]	51	9.1	351	83.5
Eff-B5 [19]	30	9.9	302	83.6
SGFormer-M [6]	39	7.5	598	84.1
SMT-B [15]	32	7.7	237	84.3
BiFormer-B [28]	57	9.8	498	84.3
RMT-Swin-S	50	9.1	722	84.5
MaxViT-S [21]	69	11.7	546	84.5
CMT-B [7]	46	9.3	447	84.5
iFormer-B [18]	48	9.4	688	84.6
RMT-B	54	9.7	457	85.0
Swin-B [16]	88	15.5	756	83.5
Eff-B6 [19]	43	19.0	172	84.0
Focal-B [23]	90	16.4	256	84.0
CSWin-B [5]	78	15.0	660	84.2
MPViT-B [13]	75	16.4	498	84.3
SMT-L [15]	80	17.7	158	84.6
SGFormer-B [6]	78	15.6	388	84.7
iFormer-L [18]	87	14.0	410	84.8
MaxViT-B [21]	120	23.4	306	84.9
RMT-L	95	18.2	326	85.5

Table 2. Comparison of inference speed.

Model	Res	Params(M)	FLOPs(G)	Top1(%)
CSwin-T	384	23	14.0	84.3
iFormer-S	384	20	16.1	84.6
RMT-S	384	27	14.7	**84.9**
RMT-S'	384	27	14.7	**85.2**

Table 3. Finetuning results on larger resolution.

Model	a	b
RMT-T	[2, 2, 2, 2]	[6, 6, 8, 8]
RMT-S	[2, 2, 2, 2]	[6, 6, 8, 8]
RMT-B	[2, 2, 2, 2]	[7, 7, 8, 8]
RMT-L	[2, 2, 2, 2]	[8, 8, 8, 8]
RMT-DeiT-S	[2]	[8]
RMT-Swin-T	[2, 2, 2, 2]	[8, 8, 8, 8]
RMT-Swin-S	[2, 2, 2, 2]	[8, 8, 8, 8]

Table 4. Details about the γ decay.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 1

[2] Kai Chen, Jiaqi Wang, Jiangmiao Pang, et al. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 1

[3] MMSegmentation Contributors. Mmsegmentation, an open source semantic segmentation toolbox, 2020. 1

[4] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, et al. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, 2020. 1

[5] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, et al. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *CVPR*, 2022. 2

[6] SG-Former: Self guided Transformer with Evolving Token Reallocation. Sucheng ren, xingyi yang, songhua liu, xinchao wang. In *ICCV*, 2023. 2

[7] Jianyuan Guo, Kai Han, Han Wu, Chang Xu, Yehui Tang, Chunjing Xu, and Yunhe Wang. Cmt: Convolutional neural networks meet vision transformers. In *CVPR*, 2022. 2

[8] Ali Hatamizadeh, Hongxu Yin, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Global context vision transformers. In *ICML*, 2023. 2

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. In *ICCV*, 2017. 1

[10] Gao Huang, Yu Sun, and Zhuang Liu. Deep networks with stochastic depth. In *ECCV*, 2016. 1

[11] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. In *NeurIPS*, 2021. 1

[12] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019. 1

[13] Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. Mpvit: Multi-path vision transformer for dense prediction. In *CVPR*, 2022. 2

[14] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, and Kaiming He andPiotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 1

[15] Weifeng Lin, Ziheng Wu, Jiayu Chen, Jun Huang, and Lianwen Jin. Scale-aware modulation meet transformer. In *ICCV*, 2023. 2

[16] 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*, 2021. 1, 2

[17] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *arXiv preprint arXiv:1906.07155*, 2019. 1

[18] Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, and Shuicheng YAN. Inception transformer. In *NeurIPS*, 2022. 2

[19] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 2

[20] Hugo Touvron, Matthieu Cord, Matthijs Douze, et al. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 1

[21] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *ECCV*, 2022. 2

[22] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 1

[23] 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. In *NeurIPS*, 2021. 1, 2

[24] Jianwei Yang, Chunyuan Li, Xiyang Dai, and Jianfeng Gao. Focal modulation networks. In *NeurIPS*, 2022. 1

[25] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, et al. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 1

[26] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, et al. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 1

[27] Zhun Zhong, Liang Zheng, Guoliang Kang, et al. Random erasing data augmentation. In *AAAI*, 2020. 1

[28] Lei Zhu, Xinjiang Wang, Zhanghan Ke, Wayne Zhang, and Rynson Lau. Biformer: Vision transformer with bi-level routing attention. In *CVPR*, 2023. 2