

PlainMamba: Improving Non-Hierarchical Mamba in Visual Recognition (Supplementary Materials)

Chenhongyi Yang*¹
chenhongyi.yang@ed.ac.uk

Zehui Chen*²
lovesnow@mail.ustc.edu.cn

Miguel Espinosa*¹
miguel.espinosa@ed.ac.uk

Linus Ericsson¹
linus.ericsson@ed.ac.uk

Zhenyu Wang³
2301210449@stu.pku.edu.cn

Jiaming Liu³
jjiamingliu@stu.pku.edu.cn

Elliot J. Crowley¹
elliot.j.crowley@ed.ac.uk

¹ School of Engineering,
University of Edinburgh

² University of Science and Technology of
China

³ Peking University

1 More Experiments

1.1 COCO Object Detection using RetinaNet

We report COCO RetinaNet object detection in Table 1. Similar to Mask R-CNN, whose results are reported in the main paper, PlainMamba also performs well with the single-stage RetinaNet object detector. For example, with only half the model size and similar FLOPs, PlainMamba-L1 achieves 0.2 higher AP than Swin-Tiny.

1.2 Ablation Studies and Discussions

Setting: Here, we conduct ablation studies to test our model designs and to gain a deeper understanding of the proposed method. We use our L1 model, with less than 10M parameters, for most experiments. The models are all pre-trained on ImageNet-1K following the same training settings described in the main paper.

Depth v.s. Width When designing neural architectures for a given parameter count, it's usually important to find a good balance between the network's depth, i.e., the number of layers, and its width, i.e., the feature dimensions. While this problem was studied for existing

* Equal Contribution

Table 1: RetinaNet object detection on MS COCO *mini-val* with $1\times$ schedule. FLOPs are computed using input size 1280×800 .

Backbone	Hierarchical	Params	FLOPs	AP^{bb}	AP_{50}^{bb}	AP_{75}^{bb}
CNN						
ResNeXt101-32x4d [10]	✓	56M	319G	39.9	-	-
ResNeXt101-64x4d [10]	✓	95M	413G	41.0	-	-
Transformer						
Swin-Tiny [11]	✓	38M	245G	41.5	-	-
Swin-Small [11]	✓	60M	335G	44.5	-	-
Focal-Tiny [12]	✓	39M	265G	43.7	-	-
Focal-Small [12]	✓	62M	367G	45.6	-	-
PVT-Small [13]	✓	34M	-	40.4	61.3	43.0
PVT-Medium [13]	✓	54M	-	41.9	63.1	44.3
PVT-Large [13]	✓	71M	-	42.6	63.7	45.4
State Space Modeling						
EfficientVMamba-T [14]	✓	13M	-	37.5	57.8	39.6
EfficientVMamba-S [14]	✓	19M	-	39.1	60.3	41.2
EfficientVMamba-B [14]	✓	44M	-	42.8	63.9	45.8
PlainMamba-Adapter-L1	✗	19M	250G	41.7	62.1	44.4
PlainMamba-Adapter-L2	✗	40M	392G	43.9	64.9	47.0
PlainMamba-Adapter-L3	✗	67M	478G	44.8	66.0	47.9

Table 2: Ablation study of model depth v.s. width on ImageNet-1K..

Depth	Width	Params	FLOPs	Top-1
6	376	7.3 M	2.5 G	74.6
12	272	7.5 M	2.7 G	76.8
24	192	7.3 M	3.0 G	77.9
36	156	7.2 M	3.3 G	77.9

architectures [14, 15], it is still unclear whether the previous conclusions are applicable to vision SSMs. In Table 2, we study the depth and width trade-off of the proposed PlainMamba. Firstly, the results show that deeper models tend to perform better than shallow ones. For example, when the parameter count is around 7.4M, the 12-layer model achieves 2.2% higher ImageNet top-1 accuracy than the 6-layer counterparts, and the 24-layer model is further 1.1% higher than the 12-layer one. However, when we further increase the depth to 36 while reducing the width accordingly, the top-1 accuracy remains similar. On the other hand, we also notice that deeper models are less efficient than shallower but wider models. For instance, the 24-layer model is 0.3G FLOPs higher than the 12-layer model. These results suggest the necessity of a good balance between network depth and width.

PlainMamba Block Design. Here, we test different designs of the PlainMamba block by comparing it with the block designs in Vision Mamba [14] and VMamba [15]. For a fair comparison, we use the same model depth and width settings for all designs and train all models with the same training recipe. We also remove the CLS tokens from

the Vision Mamba [14] block and use the global averaging pooling as an alternative. We report the results in Table 3. We can see that our design achieves the best results. Specifically, Vision Mamba only achieves a 74.4% ImageNet accuracy, which is 3.5% lower than ours. We also notice that the model with a Vision Mamba block is inferior to the original Vision Mamba

Table 3: Ablation study of model depth v.s. width on ImageNet-1K.

Method	Params	FLOPs	Top-1
VisionMamba [14]	7.8M	1.3G	74.4
VMamba [15]	7.3M	3.0G	77.1
Ours	7.3M	3.0G	77.9

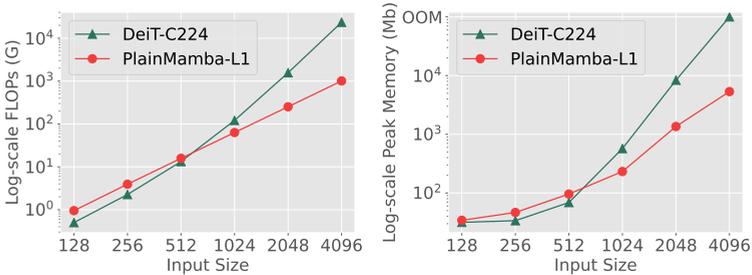


Figure 1: Efficiency comparison between PlainMamba and DeiT. We modify the DeiT-Tiny model by changing its channel number to 224, resulting in a similar-size model (7.4M) to PlainMamba-L1. The peak memory is measured using a batch size of 1.

model, which is caused by the removal of the CLS token. These results suggest that our model still retains its ability when the CLS token is absent. Also, our design performs better than the VMamba block [14] with a 0.8% accuracy advantage, indicating that the improvements come from our proposed Continuous 2D Scanning and Direction-aware Updating, which validate the effectiveness of our proposed techniques in adapting SSM for 2D images.

Efficiency Comparison with ViT One particular advantage of SSMs, e.g., Mamba, is their ability to capture global information while maintaining efficiency. In Figure 1, we compare the PlainMamba’s efficiency with the vision transformer. Specifically, to ensure a fair comparison, we create a DeiT model with channel numbers of 224, resulting in a model with 7.4M parameters, which is used to compare with PlainMamba-L1.

Specifically, we compare the model FLOPs and the peak inference memory using inputs of different sizes. The results show that our model is able to keep the computation cost low when the input size is scaled up to high resolutions, e.g., 4096×4096 . However, DeiT’s FLOPs and memory consumption increase rapidly when using such high-resolution inputs. On the other hand, we also notice that our model’s efficiency is inferior to the similar-sized DeiT when using low-resolution images, e.g., 128×128 . To further investigate such a difference in their efficiency, we decompose their FLOPs into three parts [14]: 1) *token mixing*, 2) *channel mixing*, and 3) *others*. Specifically, token mixing refers to the multi-head attention part in DeiT and the selective scanning part in PlainMamba, and channel mixing refers to the feed-forward network in DeiT and the input & output projection in PlainMamba. We report the results in Table 4. These suggest that PlainMamba’s FLOPs are evenly distributed across the three parts in low and high resolutions. On the contrary, when using 128×128 inputs, DeiT’s FLOPs are dominated by channel-mixing and the other two parts are negligible. However, because of the quadratic complexity of self-attention operation, DeiT’s FLOPs in token mixing grow to 23T when using 4096×4096 inputs, 23 times more expensive than PlainMamba. These results verify PlainMamba’s high efficiency for high-resolution inputs.

Table 4: Comparison of decomposed FLOPs between DeiT and PlainMamba.

Resolution	Part	DeiT-C224	PlainMamba-L1
128×128	Token Mixing	0.18G	0.34G
	Channel Mixing	0.31G	0.33G
	Others	0.01G	0.30G
4096×4096	Token Mixing	23244G	350G
	Channel Mixing	315G	348G
	Others	12G	311G

References

- [1] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024.
- [2] 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*, 2021.
- [3] Xiaohuan Pei, Tao Huang, and Chang Xu. Efficientvmamba: Atrous selective scan for light weight visual mamba. *arXiv preprint arXiv:2403.09977*, 2024.
- [4] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34:24261–24272, 2021.
- [5] 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*, 2021.
- [6] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90:119–133, 2019.
- [7] 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.
- [8] Fuzhao Xue, Ziji Shi, Futao Wei, Yuxuan Lou, Yong Liu, and Yang You. Go wider instead of deeper. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8779–8787, 2022.
- [9] 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.
- [10] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.