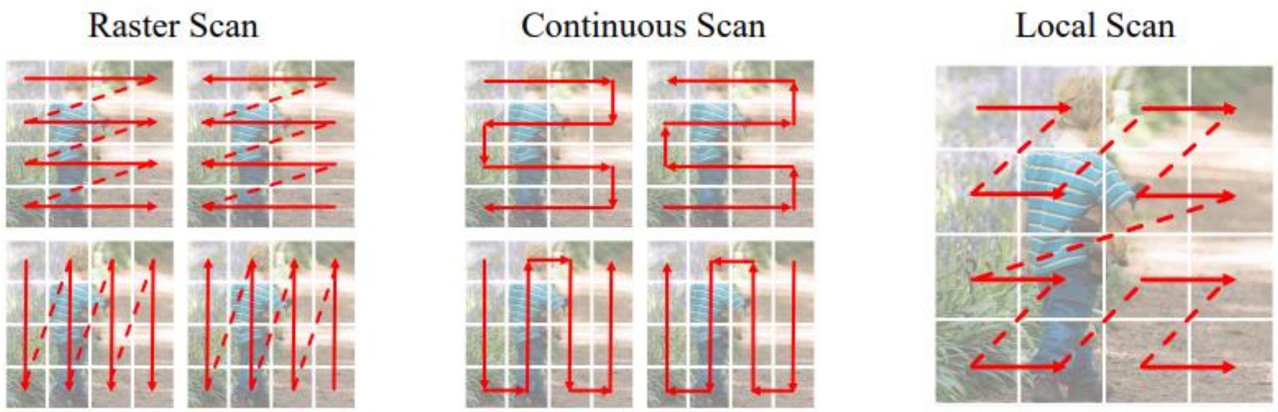
Mamba-Adaptor: State Space Model Adaptor for Visual Recognition

Fei Xie[1] Jiahao Nie[2*] Yujin Tan[1] Wenkang Zhang[3] Hongshen Zhao[3]
[1] Shanghai Jiao Tong University
[2] Hangzhou Dianzi University
[3] Southeast University
jaffe0319@gmail.com, jhnie@hdu.edu.cn

Motivation



SSM processing for visual data

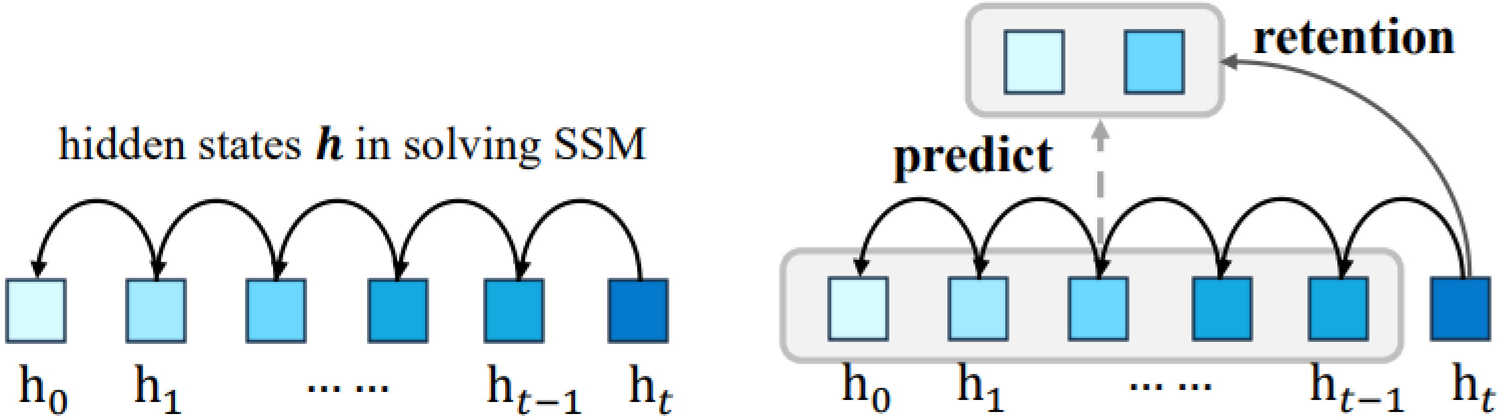
1. Temporal domain: long-range forgetting issue

   (Adaptor-T for Memory Retention)

2. Spatial domain: lack spatial structural modeling

   (Adaptor-S for Spatial Aggregation)

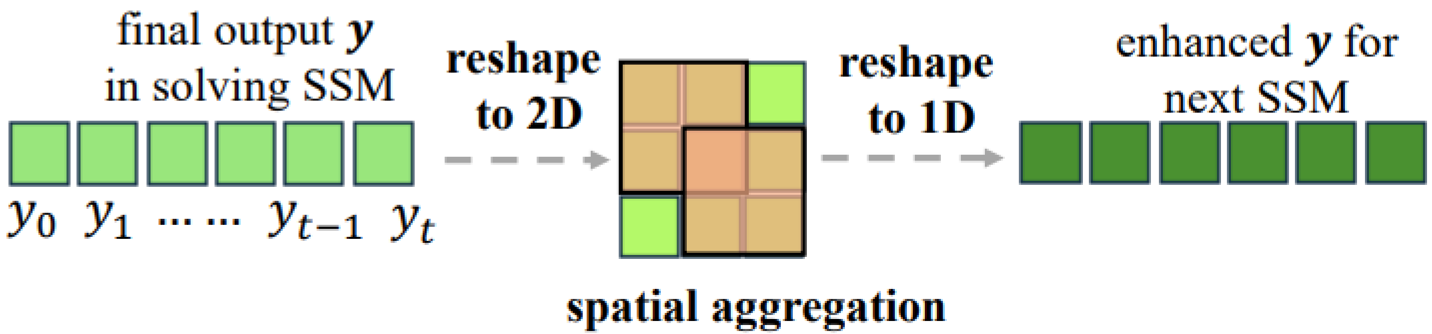
Adaptor-T for Memory Retention



Adaptor-T for memory retention

Memory augmentation for selected previous hidden states

Adaptor-S for Spatial Aggregation



Adaptor-S for spatial modeling

Spatial Convolution for enhancing spatial modeling

Implementation

Learnable memory selection

Adaptor-T

$$\{p_0, p_1, ..., p_k\} = \phi_p(h_i),$$
$$\{c_0, c_1, ..., c_k\} = \mathrm{SoftMax}(\phi_c(h_i)),$$

Multi-scale spatial aggregation

Adaptor-S

$$y_{(i,j)} = \sum_{\forall d} \sum_{\forall (i,j) \in \Omega_d} w^d_{(i,j)} y_{(i,j)},$$

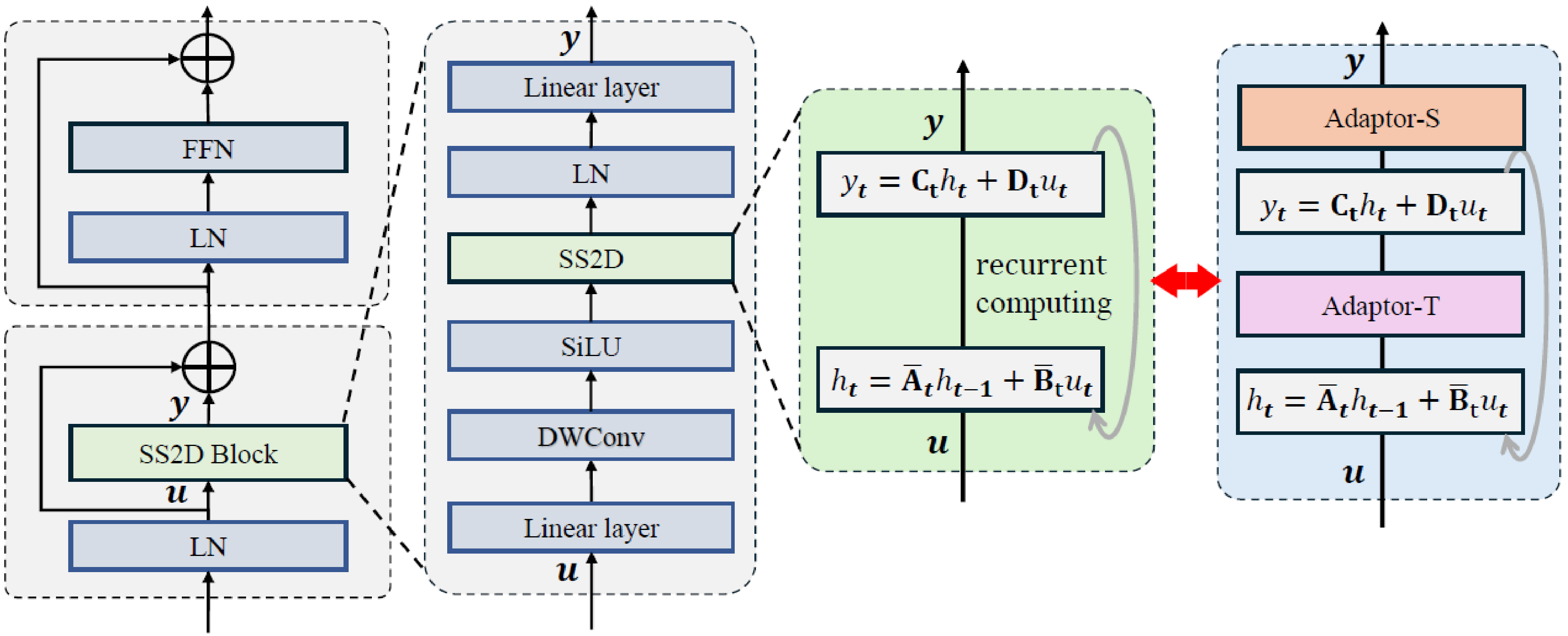
Integrated into highly-optimized Mamba operator

Algorithm 1 Pseudo code of Mamba-Adaptor

```
#input: u; params: delta, A, B, C, D; output: y
def Mamba-Adaptor(images, maps):
   # Generating identity Matrix for C
   C_identiy = torch.ones_like(C)
   # Generating zero Matrix for C
   D_zeros = torch.zeros_like(D)
   # Calculate hidden state using Mamba solver
   hidden_state = SelectiveScanCuda(u, A, B,
       C_identiy, D_zeros, delta)
   # Adaptor-T
   hidden_state = Adaptor_T(hidden_state)
   # Calculate output using matrix multiplication
   y = C*hidden_state + D*u
   # Adaptor-S
   y = Adaptor_S(y)
   return y
```

We propose efficient implementation for Adaptor-T/-S and integration.

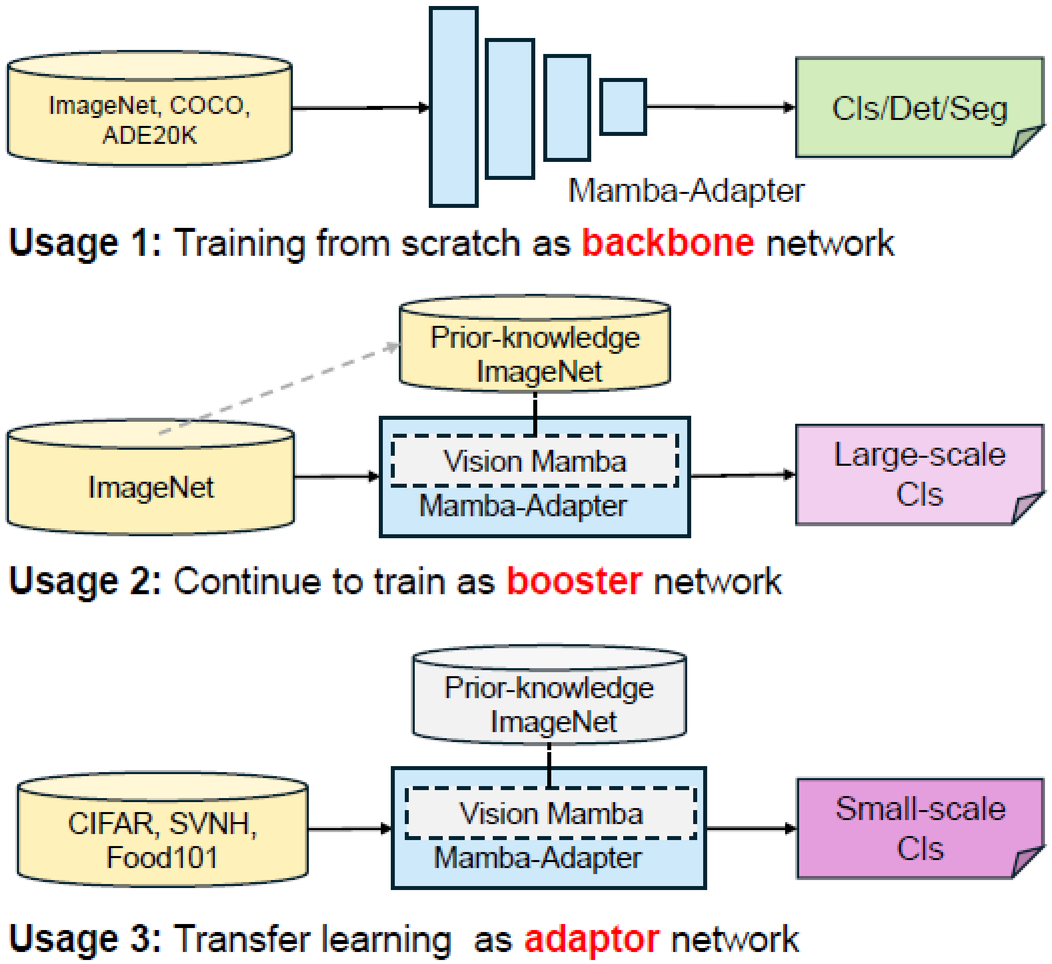
Adaptor



(a) vision Mamba layer (b) SS2D (c) Original SSM (d) SSM + Mamba-Adapter

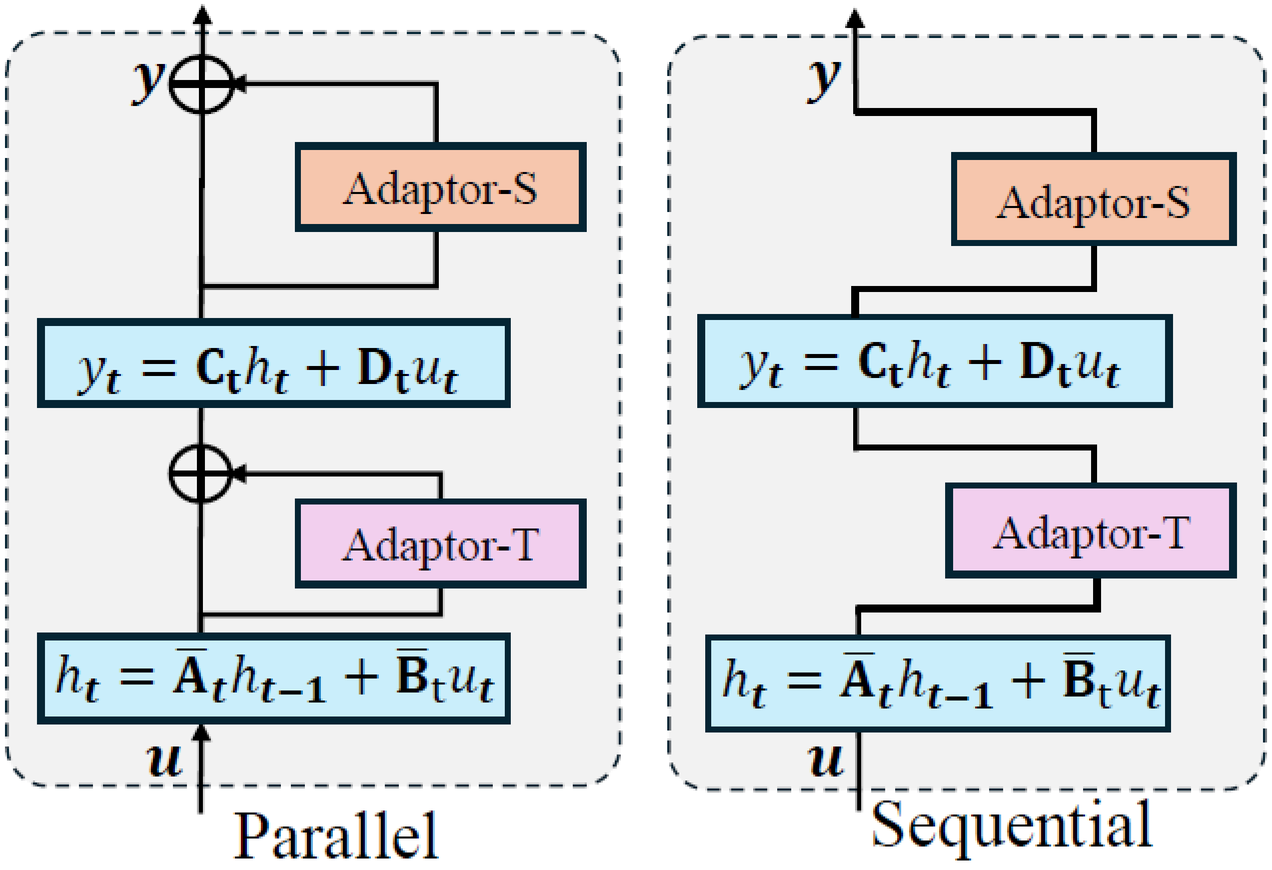
Mamba-Adaptor can be integrated into the Mamba block seamlessly

Multiple Usages



As visual backbone, booster to raise performance, adaptor for transfer learning

Insertion form



Parallel form to play as an adaptor network for transfer learning

Experiment

Method	Parmas (M)	FLOPs (G)	Top1-acc (%)
RegNetY-1.6G	11	1.6	78.0
EfficientNet-B3	12	1.8	81.6
PVTv2-b1 [52]	13	2.1	78.7
BiFormer-T [65]	13	2.2	81.4
Vim-T	7	1.5	76.1
LocalViM-T	8	1.5	76.2
Mamba-Adaptor-b1	7.8	1.4	78.4
CoAtNet-T [33]	29	4.5	82.1
UniRepLKNet-T [33]	29	4.5	82.1
ConvNeXt-T [33]	29	4.5	82.1
MambaoutOut-T	27	4.5	82.7
InternImage-T [54]	30	5.0	83.5
DeiT-S [47]	22	4.6	79.9
Swin-T [31]	29	4.5	81.3
Vim-S	26	5.1	80.3
VMamba-T	22	5.6	82.6
LocalVMamba	26	5.7	82.7
Mamba-Adaptor-b2	32	5.4	83.0

Table 1. Comparison of state-of-the-art methods for ImageNet-1K [40] classification.

As a visual backbone, Mamba-Adaptor achieves the state-of-the-art performance in ImageNet1k classification task.

Experiment

Backbone	Params (M)	FLOPs (G)	Mask R-CNN 1×+MS						Mask R-CNN 3×+MS					
			AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet18	21.3	189	31.8	49.6	33.6	16.3	34.3	43.2	-	-	-	-	-	-
PVT-T	33	208	36.7	59.2	39.3	35.1	56.7	37.3	39.8	62.2	43.0	37.4	59.3	39.9
EffVMamba-S	31	197	39.3	61.8	42.8	36.7	58.9	39.2	41.6	63.9	45.6	38.2	60.8	40.7
Mamba-Adaptor-b1	32	218	43.2	65.5	47.7	39.5	60.1	42.7	45.1	67.2	49.4	-41.2	61.9	43.8
ResNet-50	44	260	38.2	58.8	41.4	34.7	55.7	37.2	-	-	-	-	-	-
Swin-T	48	267	42.7	65.2	46.8	39.3	62.2	42.2	46.0	68.1	50.3	41.6	65.1	44.9
VMamba-T	50	271	46.5	68.5	52.0	42.1	65.5	45.3	48.8	70.4	53.5	43.7	67.4	47.0
LocalVMamba-T	45	291	46.7	68.7	50.8	42.2	65.7	45.5	48.7	70.1	53.0	43.4	67.0	46.4
Mamba-Adaptor-b2	42	259	47.3	69.8	52.3	43.4	66.9	46.4	49.1	71.5	54.1	44.8	67.3	48.3

Table 2. Comparison to the state-of-the-art backbone networks using Mask R-CNN with ”1×” and ”3×” training schedules.

As visual backbone, Mamba-Adaptor achieves state-of-the-art performance in COCO detection and instance segmentation tasks.

Experiment

Backbone	Params(M)	FLOPs(G)	mIoU-SS(%)	mIoU-MS(%)
EffVMamba-S [14]	29M	505G	41.5	42.1
MSVMamba-M [17]	42M	875G	45.1	45.4
Mamba-Adaptor-b1	38M	708G	45.4	46.0
Swin-T [11]	60M	945G	44.4	45.8
ConvNeXt-T [12]	60M	939G	46.0	46.7
VMamba-T [10]	55M	964G	47.3	48.3
EffVMamba-B [14]	65M	930G	46.5	47.3
MSVMamba-T [17]	65M	942G	47.6	48.5
Mamba-Adaptor-b2	-58M	971G	47.8	48.6

Table 1. Additional experiments on the semantic segmentation in ADE20K [25] benchmark. SS and MS denote single-scale and multi-scale inference settings.

As visual backbone, Mamba-Adaptor achieves state-of-the-art performance in ADE20k semantic segmentation task.

Experiment

Method	Parmas (M)	FLOPs (G)	Top1-acc (%)
VMamba-T	30	4.9	82.6
Adaptor-VMamba-T	31 (3.2% ↑)	5.2 (6.1% ↑)	**82.7**
VMamba-S	50	8.7	83.6
Adaptor-VMamba-S	53 (6.1% ↑)	9.3 (8.0% ↑)	**83.7**
VMamba-B	89	15.4	83.9
Adaptor-VMamba-B	94 (5.6% ↑)	16.5 (7.1% ↑)	**84.1**

Table 3. Improvements over ImageNet-1K classification. Adaptor-Baseline denotes the baseline model, which is equipped with our Mamba-Adaptor as a booster module.

As a booster, Mamba-Adaptor further raises the classification accuracy of a baseline model.

Experiment

Adaptor Method	Base Model	Pretrain	Parmas (M)	CIFAR-100 (Acc%)	SVHN (Acc%)	Food101 (Acc%)
Full-Tuning	ViT-B [7]	MAE [19]	86.04 (100%)	85.90	97.67	90.09
Linear	ViT-B [7]	MAE [19]	0.07 (0.08%)	69.83 (-16.07)	66.91 (-30.76)	69.74 (-20.35)
VPT [25]	ViT-B [7]	MAE [19]	0.08 (0.09%)	82.44 (-3.46)	94.02 (-3.65)	82.98 (-7.11)
Full-Tuning	VMamba-T [30]	Cls. [30]	30.25 (100%)	87.48	97.82	90.22
Linear	VMamba-T [30]	Cls. [30]	0.02 (0.06%)	61.23 (-26.25)	54.36 (-43.46)	61.62 (-28.60)
VPT [25]	VMamba-T [30]	Cls. [30]	0.03 (0.09%)	80.68 (-6.80)	89.23 (-8.59)	80.34 (-9.78)
Mamba-Adaptor	VMamba-T [30]	Cls. [30]	1.68 (5.56%)	86.82 (-0.66)	93.24 (-4.58)	86.45 (-3.76)
Full-Tuning	VMamba-S [30]	Cls. [30]	55.25 (100%)	89.59	97.90	91.24
Linear	VMamba-S [30]	Cls. [30]	0.02 (0.04%)	65.34 (-24.25)	52.12 (-45.78)	68.98 (-22.26)
VPT [25]	VMamba-S [30]	Cls. [30]	0.03 (0.05%)	82.26 (-6.33)	82.48 (-15.42)	64.23 (-27.01)
Mamba-Adaptor	VMamba-S [30]	Cls. [30]	5.10 (9.25%)	88.30 (-1.39)	85.69 (-12.21)	82.33 (-8.91)
Full-Tuning	VMamba-B [30]	Cls. [30]	95.36 (100%)	89.89	97.96	91.68
Linear	VMamba-B [30]	Cls. [30]	0.03 (0.03%)	67.23 (-22.66)	55.69 (-42.27)	69.42 (-22.16)
VPT [25]	VMamba-B [30]	Cls. [30]	0.04 (0.04%)	81.32 (-8.57)	87.45 (-10.51)	80.34 (-11.36)
Mamba-Adaptor	VMamba-B [30]	Cls. [30]	6.8 (7.13%)	88.34 (-1.5)	88.52 (-8.44)	83.21 (-8.47)

Table 4. Fine-tuning with the pre-trained base models for transfer learning tasks, where Mamba-Adaptor serves as an adaptor network. For tunable parameters, we report the percentage of the parameters. Additionally, we show both the absolute value and the gap value relative to the top 1 accuracy of the full-tuning setting. Cls. denotes normal image classificaiton training [30, 31].

As an adaptor, Mamba-Adaptor achieves competitive results on transfer learning tasks.

Thanks