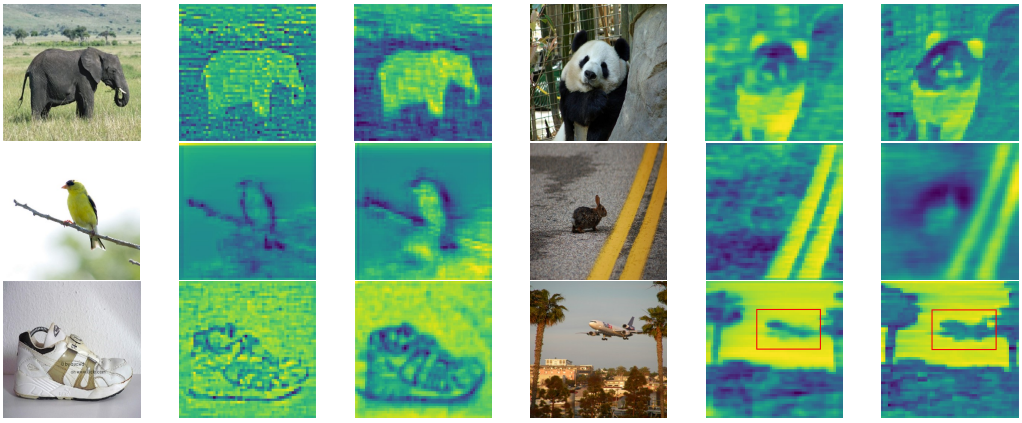
Appendix to "Spatial-Mamba: Effective Visual State Space Models via Structure-aware State Fusion"

In this appendix, we provide the following materials:

A More visual results for SASF (referring to Sec. 4.1 in the main paper);
B Architecture details of Spatial-Mamba and experimental settings (referring to Sec. 4.2 and Sec. 5.1 in the main paper);
C Derivations of the SSM formulas of Mamba and Spatial-Mamba (referring to Sec. 4.3 in the main paper);
D Results of Cascade Mask R-CNN detector head (referring to Sec. 5.2 in the main paper);
E Visual results and comparisons for detection and segmentation tasks (referring to Sec. 5.2 and Sec. 5.3 in the main paper);
F Further discussions about SASF (referring to Sec. 5.4 in the main paper);
G Comparisons of effective receptive field (referring to Sec. 5.4 in the main paper).

A More Visual Results

More visual comparisons between the original and fused structure-aware state variables are shown in Fig. 1. Due to the spatial modeling capability of SASF, the fused state variables demonstrate more accurate context information and superior structural perception across different scenarios.



(a) Input (b) Original state (c) Fused state (d) Input (e) Original state (f) Fused state

Figure 1: More visualizations of state variables before and after applying the SASF equation.

B Model Architectures and Experiment Settings in Classification

Network architecture. The detailed architectures of Spatial-Mamba models are outlined in Tab. 1. Following the common four-stage hierarchical framework (Liu et al., 2021; Han et al., 2024), we construct the Spatial-Mamba models by stacking our proposed Spatial-Mamba blocks at each stage. Specifically, an input image with resolution of 224×224 is firstly processed by a stem layer, which consists of Convolution (Conv), Batch Normalization (BN) and GELU activation function. The kernel size is 3×3 with a stride of 2 at the first and last convolution layers, and a stride of 1 for other layers. Each stage contains multiple Spatial-Mamba blocks, followed by a down-sampling layer except for the last block. The down-sampling layer consists of a 3×3 convolution with a stride of 2 and a Layer Normalization (LN) layer. Each block incorporates a structure-aware SSM layer and a Feed-Forward Network (FFN), both with residual connections. The structure-aware SSM contains a SASF branch with a 2D depth-wise convolution and a multiplicative gate branch with activation function, as illustrated in Sec. 4.2 of the main paper. The expand ratio of SSM is set to 2, doubling the

Table 1: Architectures of Spatial-Mamba models, where Linear refers to a linear layer, DWConv represents a depth-wise convolution layer, SASF module refers to Fig. 4 in Sec. 4.2 of the main paper, and FFN is a Feed-Forward Network.

Layer	Output size	Spatial-Mamba-T	Spatial-Mamba-S	Spatial-Mamba-B
Stem	56×56	Conv 3×3 stride 2, BN, GELU; Conv 3×3 stride 1, BN; Conv 3×3 stride 2, BN		
Stage1	28×28	Spatial-Mamba Blocks [Linear $64 \to 128$, DWConv 128, SASF 128, Linear $128 \to 64$, FFN 64] $\times 2$	Spatial-Mamba Blocks [Linear $64 \to 128$, DWConv 128, SASF 128, Linear $128 \to 64$, FFN 64] $\times 2$	Spatial-Mamba Blocks [Linear $96 \to 192$, DWConv 192, SASF 192, Linear $192 \to 96$, FFN 96] $\times 2$
		Down Sampling Conv 3×3 stride 2, LN		
Stage2	14×14	Spatial-Mamba Blocks [Linear $128 \to 256$, DWConv 256, SASF 256, Linear $256 \to 128$, FFN 128] $\times 4$	Spatial-Mamba Blocks [Linear $128 \to 256$, DWConv 256, SASF 256, Linear $256 \to 128$, FFN 128] $\times 4$	Spatial-Mamba Blocks [Linear $192 \to 384$, DWConv 384, SASF 384, Linear $384 \to 192$, FFN 192] $\times 4$
		Down Sampling Conv 3×3 stride 2, LN		
Stage3	7×7	Spatial-Mamba Blocks [Linear $256 \to 512$, DWConv 512, SASF 512, Linear $512 \to 256$, FFN 256] $\times 8$	Spatial-Mamba Blocks [Linear $256 \to 512$, DWConv 512, SASF 512, Linear $512 \to 256$, FFN 256] $\times 21$	Spatial-Mamba Blocks [Linear $384 \to 768$, DWConv 768, SASF 768, Linear $768 \to 384$, FFN 384] $\times 21$
		Down Sampling Conv 3×3 stride 2, LN		
Stage4	7×7	Spatial-Mamba Blocks [Linear $512 \to 1024$, DWConv 1024, SASF 1024, Linear $1024 \to 512$, FFN 512] $\times 4$	Spatial-Mamba Blocks [Linear $512 \to 1024$, DWConv 1024, SASF 1024, Linear $1024 \to 512$, FFN 512] $\times 5$	Spatial-Mamba Blocks [Linear $768 \to 1536$, DWConv 1536, SASF 1536, Linear $1536 \to 768$, FFN 768] $\times 5$
Head	1×1	Average pool, Linear 1000, Softmax		

number of channels. The SSM state dimension is set to 1 for better performance and efficiency. We modify the embedding dimension and number of blocks to build our Spatial-Mamba-T/S/B models.

Settings for ImageNet-1K classification. The Spatial-Mamba-T/S/B models are trained from scratch for 300 epochs using AdamW optimizer with betas set to (0.9, 0.999), momentum set to 0.9, and batch size set to 1024. The initial learning rate is set to 0.001 with a weight decay of 0.05. A cosine annealing learning rate schedule is adopted with a warm-up of 20 epochs. We adopt the common data augmentation strategies as in previous works (Liu et al., 2021; 2024). Moreover, label smoothing (0.1), exponential moving average (EMA) and MESA (Du et al., 2022) are also applied. The drop path rate is set to 0.2 for Spatial-Mamba-T, 0.3 for Spatial-Mamba-S and 0.5 for Spatial-Mamba-B.

Implementation details. The Spatial-Mamba models employ a hardware-aware selective scan algorithm adapted from the original Mamba framework, with modifications to the CUDA kernels for decoupling state transition and observation equations. The SASF module of Spatial-Mamba is implemented by the general matrix multiplication (GEMM) with optimized CUDA kernels.

C Derivation of SSM Formulas

Based on the Mamba formulation provided in Sec. 3 of the main paper, the state transition equation in the recursive form can be rewritten as follows:

$$
\begin{aligned}
x_t &= \overline{\boldsymbol{A}}_t x_{t-1} + \overline{\boldsymbol{B}}_t u_t \\
&= \overline{\boldsymbol{A}}_t \left(\overline{\boldsymbol{A}}_{t-1} x_{t-2} + \overline{\boldsymbol{B}}_{t-1} u_{t-1} \right) + \overline{\boldsymbol{B}}_t u_t \\
&= \overline{\boldsymbol{A}}_t \left(\overline{\boldsymbol{A}}_{t-1} \left(\overline{\boldsymbol{A}}_{t-2} x_{t-3} + \overline{\boldsymbol{B}}_{t-2} u_{t-2} \right) + \overline{\boldsymbol{B}}_{t-1} u_{t-1} \right) + \overline{\boldsymbol{B}}_t u_t \\
&= \overline{\boldsymbol{A}}_t \overline{\boldsymbol{A}}_{t-1} \overline{\boldsymbol{A}}_{t-2} x_{t-3} + \overline{\boldsymbol{A}}_t \overline{\boldsymbol{A}}_{t-1} \overline{\boldsymbol{B}}_{t-2} u_{t-2} + \overline{\boldsymbol{A}}_t \overline{\boldsymbol{B}}_{t-1} u_{t-1} + \overline{\boldsymbol{B}}_t u_t \\
&= \Pi_{i=1}^{t} \overline{\boldsymbol{A}}_i x_0 + \Pi_{i=2}^{t} \overline{\boldsymbol{A}}_i \overline{\boldsymbol{B}}_1 u_1 + \cdots + \Pi_{i=t-1}^{t} \overline{\boldsymbol{A}}_i \overline{\boldsymbol{B}}_{t-2} u_{t-2} + \Pi_{i=t}^{t} \overline{\boldsymbol{A}}_i \overline{\boldsymbol{B}}_{t-1} u_{t-1} + \overline{\boldsymbol{B}}_t u_t
\end{aligned}
\tag{1}
$$

By defining the initial state as zero, $x_0 = 0$, Eq. (1) can be expressed as:

$$x_t = \sum_{s \leq t} \overline{\boldsymbol{A}}^{\times}_{s:t} \, \overline{\boldsymbol{B}}_s u_s, \quad \text{where } \overline{\boldsymbol{A}}^{\times}_{s:t} := \begin{cases} \Pi_{i=s+1}^{t} \overline{\boldsymbol{A}}_i, & s < t \\ 1, & s = t \end{cases}. \tag{2}$$

We omit the term $\boldsymbol{D}_t u_t$ for simplicity. According to the observation equation, we can derive the final output $y_t = \sum_{s \leq t} \boldsymbol{C}_t \, \overline{\boldsymbol{A}}^{\times}_{s:t} \, \overline{\boldsymbol{B}}_s u_s$. Suppose the input vector is $u = [u_1, \cdots, u_L]^T \in \mathbb{R}^L$ with length L, the corresponding output vector is $y \in \mathbb{R}^L$. Then the above calculation can be written in matrix multiplication form, *i.e.*, $y = \boldsymbol{M}u$, where $\boldsymbol{M}$ is a structured lower triangular matrix and $\boldsymbol{M}_{ij} = \boldsymbol{C}_i \overline{\boldsymbol{A}}^{\times}_{j:i} \overline{\boldsymbol{B}}_j$.

Similarly, we can represent Spatial-Mamba in the same matrix transformation form. Based on the definition of Spatial-Mamba in Sec. 4.1 and Eq. (2), the SASF equation can be rewritten as $h_t = \sum_{k \in \Omega} \sum_{s \leq \rho_k(t)} \alpha_k \overline{\boldsymbol{A}}^{\times}_{s:\rho_k(t)} \overline{\boldsymbol{B}}_s u_s$. By multiplying $\boldsymbol{C}_t$, we can derive the final output of Spatial-Mamba as $y_t = \sum_{k \in \Omega} \sum_{s \leq \rho_k(t)} \alpha_k \boldsymbol{C}_t \overline{\boldsymbol{A}}^{\times}_{s:\rho_k(t)} \overline{\boldsymbol{B}}_s u_s$. It can also be concisely represented as a matrix multiplication form $y = \boldsymbol{M}u$, where $\boldsymbol{M}$ is a structured adjacency matrix and $\boldsymbol{M}_{ij} = \sum_k \boldsymbol{C}_i \overline{\boldsymbol{A}}^{\times}_{j:\rho_k(i)} \overline{\boldsymbol{B}}_j$.

D Results of Cascade Mask R-CNN Detector Head

Detailed results of object detection and instance segmentation on the COCO dataset with Cascade Mask R-CNN framework are reported in Tab. 2. We can see that Spatial-Mamba-T achieves a box mAP of 52.1 and a mask mAP of 44.9, surpassing Swin-T/NAT-T by 1.7/0.7 in box mAP and 1.2/0.4 in mask mAP with fewer parameters and FLOPs, respectively. Similarly, Spatial-Mamba-S demonstrates superior performance under the same configuration.

Table 2: Results of object detection and instance segmentation on the COCO dataset using Cascade Mask R-CNN (Cai & Vasconcelos, 2018) under 3× schedule. FLOPs are calculated with input resolution of 1280 × 800.

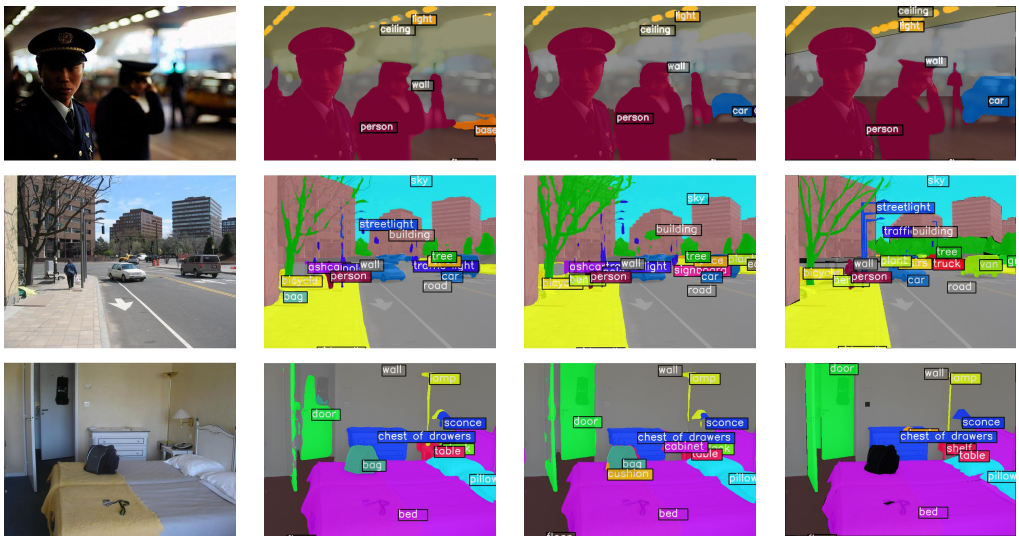
Backbone	$AP^b \uparrow$	$AP^b_{50} \uparrow$	$AP^b_{75} \uparrow$	$AP^m \uparrow$	$AP^m_{50} \uparrow$	$AP^m_{75} \uparrow$	#Param.	FLOPs
Swin-T	50.4	69.2	54.7	43.7	66.6	47.3	86M	745G
ConvNeXt-T	50.4	69.1	54.8	43.7	66.5	47.3	86M	741G
NAT-T	51.4	70.0	55.9	44.5	67.6	47.9	85M	737G
Spatial-Mamba-T	**52.1**	**71.0**	**56.5**	**44.9**	**68.3**	**48.7**	84M	740G
Swin-S	51.9	70.7	56.3	45.0	68.2	48.8	107M	838G
ConvNeXt-S	51.9	70.8	56.5	45.0	68.4	49.1	108M	827G
NAT-S	52.0	70.4	56.3	44.9	68.1	48.6	108M	809G
Spatial-Mamba-S	**53.3**	**71.9**	**57.9**	**45.8**	**69.4**	**49.7**	101M	794G

E Qualitative Results

In this section, we present the visualization results of object detection and instance segmentation in Fig. 2, and present the results of semantic segmentation in Fig. 3. Compared with VMamba, our Spatial-Mamba demonstrates superior performance in both tasks, producing more accurate detection boxes segmentation masks, particularly in areas where local structural information is crucial. For example, in the second row of Fig. 2, VMamba mistakenly identifies the shoes on a skateboard as a person, probably because it observes the shoes from four directions independently and resembles them as a human. Our method avoids this mistake by simultaneously perceiving the shoes and their surrounding context. Similarly, in the semantic segmentation task, as shown in the second and third rows of Fig. 3, our approach achieves more precise structures of trees and doors. These results highlight the effectiveness of our proposed Spatial-Mamba in leveraging local structural information for better visual understanding.

(a) Input (b) VMamba (c) Spatial-Mamba (d) Ground Truth

Figure 2: Visualization examples of object detection and instance segmentation on COCO dataset with Mask R-CNN 1× schedule (He et al., 2017) by VMamba and Spatial-Mamba.



(a) Input (b) VMamba (c) Spatial-Mamba (d) Ground Truth

Figure 3: Visualization examples of semantic segmentation on ADE20K dataset with single-scale inputs by VMamba and Spatial-Mamba.

F Further Discussions about SASF

SASF extension. To further explore and validate the spatial modeling capability of our proposed SASF, we adapt SASF into Vim and VMamba backbones, resulting in Vim+SASF (by replacing the middle class token with average pooling and setting state dimension to 1) and VMamba+SASF (directly integrated with SASF). Under the same training settings, the classification results on ImageNet-1K are presented in Tab. 3. It can be seen that by integrating with SASF, the bi-directional Vim model is improved by 0.5% in top-1 accuracy and the cross-scan VMamba is improved by 0.3%. These findings verify that SASF can even enhance the performance of multi-directional models, which have already incorporated (yet in an inefficient manner) spatial modeling operations.

Table 3: Comparison of classification performance on ImageNet-1K by integrating with SASF, where 'Throughput' is measured using an A100 GPU with an input resolution of 224×224.

Method	Im. size	#Param. (M)	FLOPs (G)	Throughput↑	Top-1 acc.↑
Vim-Ti	224^2	6M	1.4G	-	71.7
Vim-Ti+SASF	224^2	7M	1.6G	-	**72.2**
VMamba-T	224^2	30M	4.9G	1686	82.6
VMamba-T+SASF	224^2	31M	5.1G	1126	**82.9**

Fusion operators. There are also different choices of the operators that can be used for fusing state variables in SASF. We opt for depth-wise convolution due to its simplicity and computational efficiency. However, operators like dynamic convolution (Chen et al., 2020), deformable convolution (Dai et al., 2017), and attention mechanisms have also demonstrated superior performance in computer vision tasks. Therefore, we anticipate that they can be used as alternatives to the depth-wise convolution in our Spatial-Mamba. We simply train a Spatial-Mamba-T network with dynamic convolution and the results are shown Tab. 4. We can see that dynamic convolution yields a 0.2% performance gain in accuracy but significantly reduces throughput from 1438 to 907. This result suggests the potential advantages of more flexible fusion operators for SASF, but also highlights the importance of considering computational cost.

Table 4: Comparison of classification performance by Spatial-Mamba-T with depth-wise conv. and dynamic conv. on ImageNet-1K, where 'Throughput' is measured using an A100 GPU with an input resolution of 224×224.

Method	Im. size	#Param. (M)	FLOPs (G)	Throughput↑	Top-1 acc.↑
Depth-wise Conv.	224^2	27M	4.5G	1438	83.5
Dynamic Conv.	224^2	33M	4.8G	907	83.7

G Effective Receptive Field (ERF)

We compare the Effective Receptive Field (ERF) (Ding et al., 2022) of the center pixel on popular backbone networks before and after training, as shown in Fig. 4. The ERF values represent the contributions of every pixel on input space to the central pixel in the final output feature maps. To visualization, we randomly select 50 images from the ImageNet-1K validation set, resize them to a resolution of 1024×1024, and then calculate the ERF values with the auto-grad mechanism. Before training, our Spatial-Mamba-T initially exhibits a larger receptive field than other methods except DeiT-S due to neighborhood connectivity in the state space. After training, our method, along with DeiT-S, Vim-S, and VMamba-T, all demonstrate a global ERF. In addition, both Vim-S and VMamba-T exhibit noticeable accumulation contributions along either horizontal or vertical directions, which can be attributed to their multi-directional fusion mechanisms. In contrast, our unidirectional Spatial-Mamba-T effectively eliminates this directional bias.

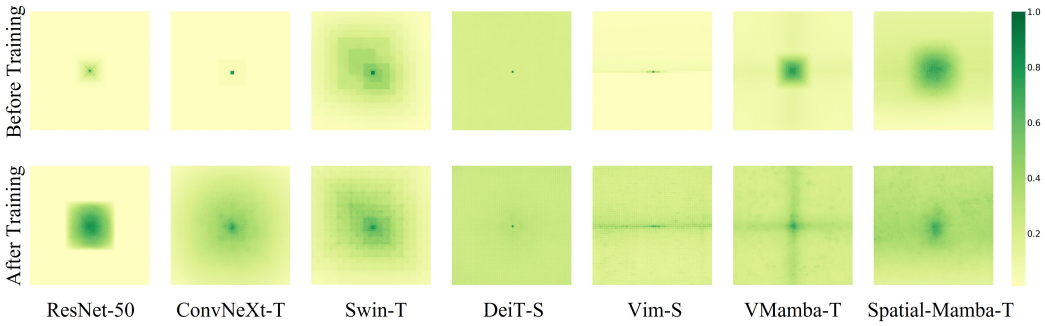


Figure 4: Comparison of Effective Receptive Field (ERF) among popular backbone networks.

REFERENCES

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*, pp. 6154–6162, 2018.

Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11030–11039, 2020.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 764–773, 2017.

Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11963–11975, 2022.

Jiawei Du, Daquan Zhou, Jiashi Feng, Vincent Tan, and Joey Tianyi Zhou. Sharpness-aware training for free. *Advances in Neural Information Processing Systems*, 35:23439–23451, 2022.

Dongchen Han, Ziyi Wang, Zhuofan Xia, Yizeng Han, Yifan Pu, Chunjiang Ge, Jun Song, Shiji Song, Bo Zheng, and Gao Huang. Demystify mamba in vision: A linear attention perspective. *arXiv preprint arXiv:2405.16605*, 2024.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

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.

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*, pp. 10012–10022, 2021.