

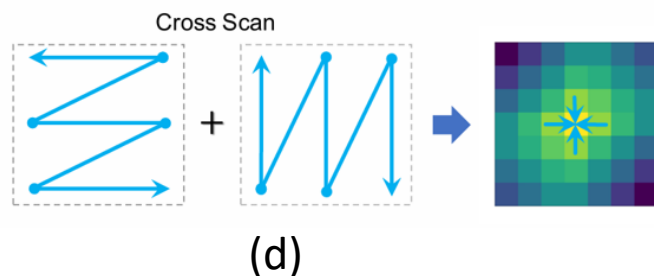
VMamba: Visual State Space Model

Yue Liu¹ Yunjie Tian¹ Yuzhong Zhao¹ Hongtian Yu¹
Lingxi Xie² Yaowei Wang³ Qixiang Ye¹ Jianbin Jiao¹ Yunfan Liu^{1‡}
¹ UCAS ² Huawei Inc. ³ Pengcheng Lab.

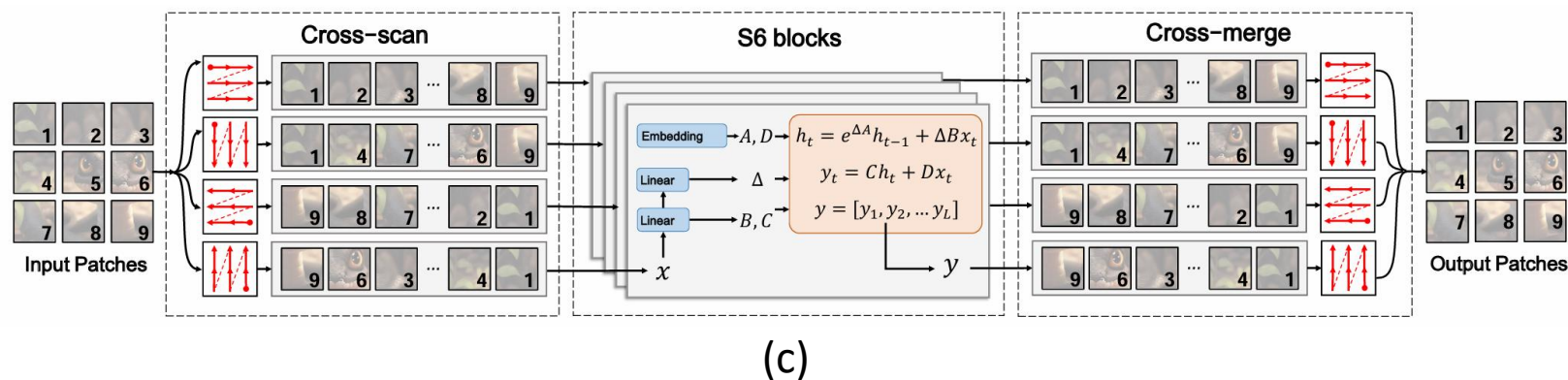
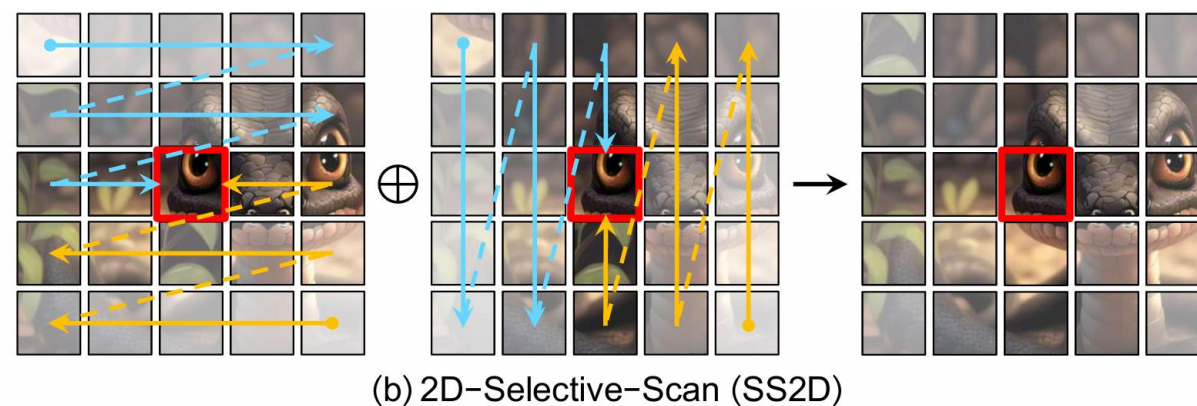
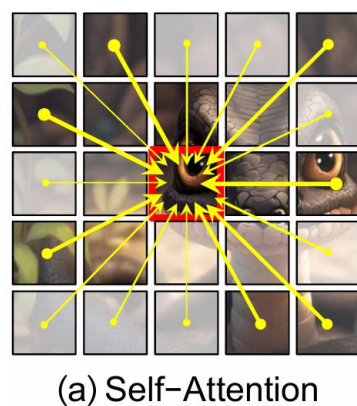
{liuyue171,tianyunjie19,zhaoyuzhong20,yuhongtian17}@mailsucas.ac.cn
198808xc@gmail.com, wangyw@pcl.ac.cn, {qxyc, jiaojb, liuyunfan}@ucas.ac.cn



Core-Idea: How to integrate RNN-like models into 2D vision tasks, while maintaining linear complexity?



In contrast to the self-attention mechanism, SS2D ensures that each image patch acquires contextual knowledge exclusively through a compressed hidden state computed along its corresponding scanning path.



Preliminaries

$$\begin{aligned} \mathbf{h}'(t) &= \mathbf{A}\mathbf{h}(t) + \mathbf{B}u(t), \\ y(t) &= \mathbf{C}\mathbf{h}(t) + Du(t), \end{aligned} \quad \Rightarrow \quad \mathbf{h}(t_b) = e^{\mathbf{A}(t_b-t_a)}\mathbf{h}(t_a) + e^{\mathbf{A}(t_b-t_a)} \int_{t_a}^{t_b} \mathbf{B}(\tau)u(\tau)e^{-\mathbf{A}(\tau-t_a)} d\tau.$$

$$\mathbf{h}_b = \mathbf{p}_{\mathbf{A},a}^b \mathbf{h}_a + \mathbf{p}_{\mathbf{B},a}^b \quad \Leftarrow \quad \mathbf{h}_b = e^{\mathbf{A}(\Delta_a + \dots + \Delta_{b-1})} \left(\mathbf{h}_a + \sum_{i=a}^{b-1} \mathbf{B}_i u_i e^{-\mathbf{A}(\Delta_a + \dots + \Delta_i)} \Delta_i \right),$$

$$\begin{aligned} \mathbf{p}_{\mathbf{A},a}^i &= e^{\mathbf{A}\Delta_{i-1}} \mathbf{p}_{\mathbf{A},a}^{i-1}, & \mathbf{p}_{\mathbf{B},a}^b &= e^{\mathbf{A}(\Delta_a + \dots + \Delta_{b-1})} \sum_{i=a}^{b-1} \mathbf{B}_i u_i e^{-\mathbf{A}(\Delta_a + \dots + \Delta_i)} \Delta_i \\ & & &= e^{\mathbf{A}\Delta_{b-1}} \mathbf{p}_{\mathbf{B},a}^{b-1} + \mathbf{B}_{b-1} u_{b-1} \Delta_{b-1}. \end{aligned}$$

```
template<>
struct SSMScanOp<float> {
    __device__ __forceinline__ float2 operator()(const float2 &ab0, const float2 &ab1) const {
        return make_float2(ab1.x * ab0.x, ab1.x * ab0.y + ab1.y);
    }
};
```



Preliminaries

$\mathbf{V} := [\mathbf{V}_1; \dots; \mathbf{V}_T] \in \mathbb{R}^{T \times D_v}$, where $\mathbf{V}_i := \mathbf{u}_{a+i-1} \odot \Delta_{a+i-1} \in \mathbb{R}^{1 \times D_v}$

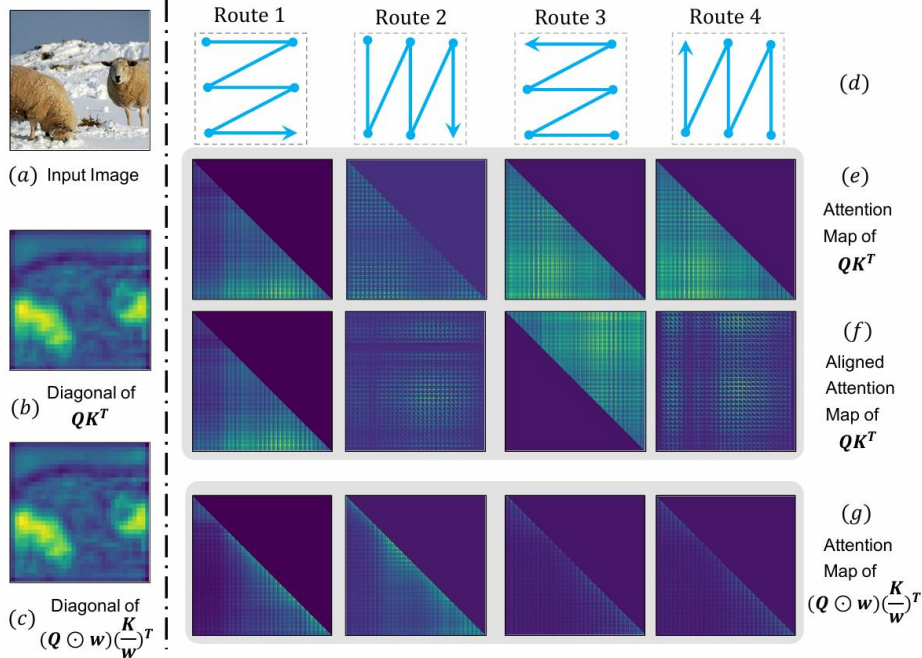
$\mathbf{K} := [\mathbf{K}_1; \dots; \mathbf{K}_T] \in \mathbb{R}^{T \times D_k}$, where $\mathbf{K}_i := \mathbf{B}_{a+i-1} \in \mathbb{R}^{1 \times D_k}$

$\mathbf{Q} := [\mathbf{Q}_1; \dots; \mathbf{Q}_T] \in \mathbb{R}^{T \times D_k}$, where $\mathbf{Q}_i := \mathbf{C}_{a+i-1} \in \mathbb{R}^{1 \times D_k}$

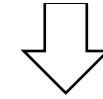
$\mathbf{w} := [\mathbf{w}_1; \dots; \mathbf{w}_T] \in \mathbb{R}^{T \times D_k \times D_v}$, where $\mathbf{w}_i := \prod_{j=1}^i e^{\mathbf{A} \Delta_{a-1+j}} \in \mathbb{R}^{D_k \times D_v}$

$\mathbf{H} := [\mathbf{h}_{a+1}; \dots; \mathbf{h}_b] \in \mathbb{R}^{T \times D_k \times D_v}$, where $\mathbf{h}_i \in \mathbb{R}^{D_k \times D_v}$

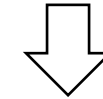
$\mathbf{Y} := [\mathbf{y}_{a+1}; \dots; \mathbf{y}_b] \in \mathbb{R}^{T \times D_v}$, where $\mathbf{y}_i \in \mathbb{R}^{D_v}$



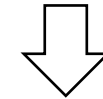
$$\mathbf{h}_b = e^{\mathbf{A}(\Delta_a + \dots + \Delta_{b-1})} \left(\mathbf{h}_a + \sum_{i=a}^{b-1} \mathbf{B}_i u_i e^{-\mathbf{A}(\Delta_a + \dots + \Delta_i)} \Delta_i \right),$$



$$\mathbf{h}_b = \mathbf{w}_T \odot \mathbf{h}_a + \sum_{i=1}^T \frac{\mathbf{w}_T}{\mathbf{w}_i} \odot (\mathbf{K}_i^\top \mathbf{V}_i),$$



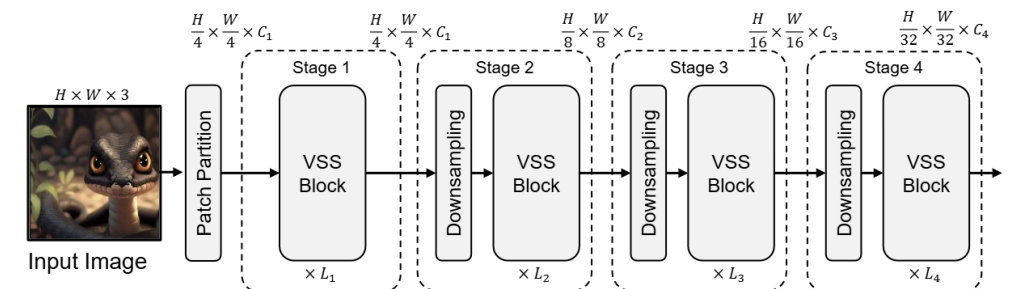
$$\mathbf{y}_b^{(j)} = (\mathbf{Q}_T \odot \mathbf{w}_T^{(j)}) \mathbf{h}_a^{(j)} + \sum_{i=1}^T \left(\frac{\mathbf{Q}_T \odot \mathbf{w}_T^{(j)}}{\mathbf{w}_i^{(j)}} \mathbf{K}_i^\top \right) \odot \mathbf{V}_i^{(j)}.$$



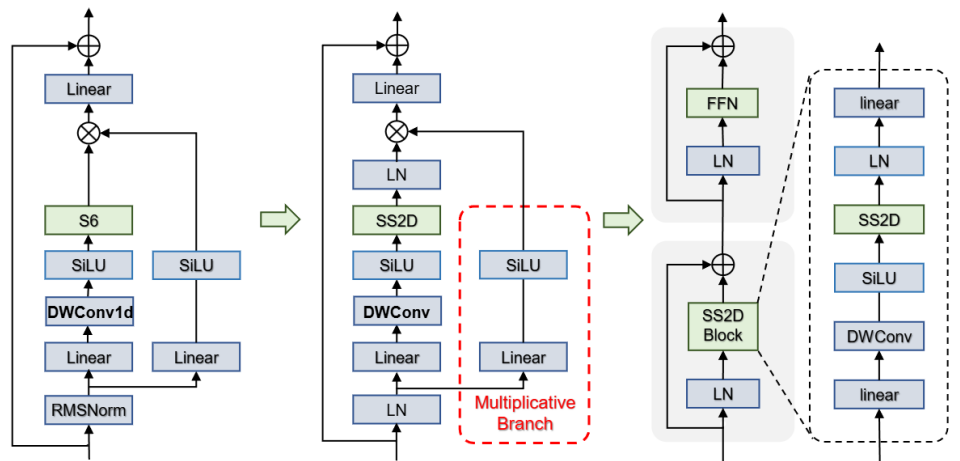
$$\mathbf{Y}^{(j)} = (\mathbf{Q} \odot \mathbf{w}^{(j)}) \mathbf{h}_a^{(j)} + \left[(\mathbf{Q} \odot \mathbf{w}^{(j)}) \left(\frac{\mathbf{K}}{\mathbf{w}^{(j)}} \right)^\top \odot \mathbf{M} \right] \mathbf{V}^{(j)},$$



Architecture and Performance



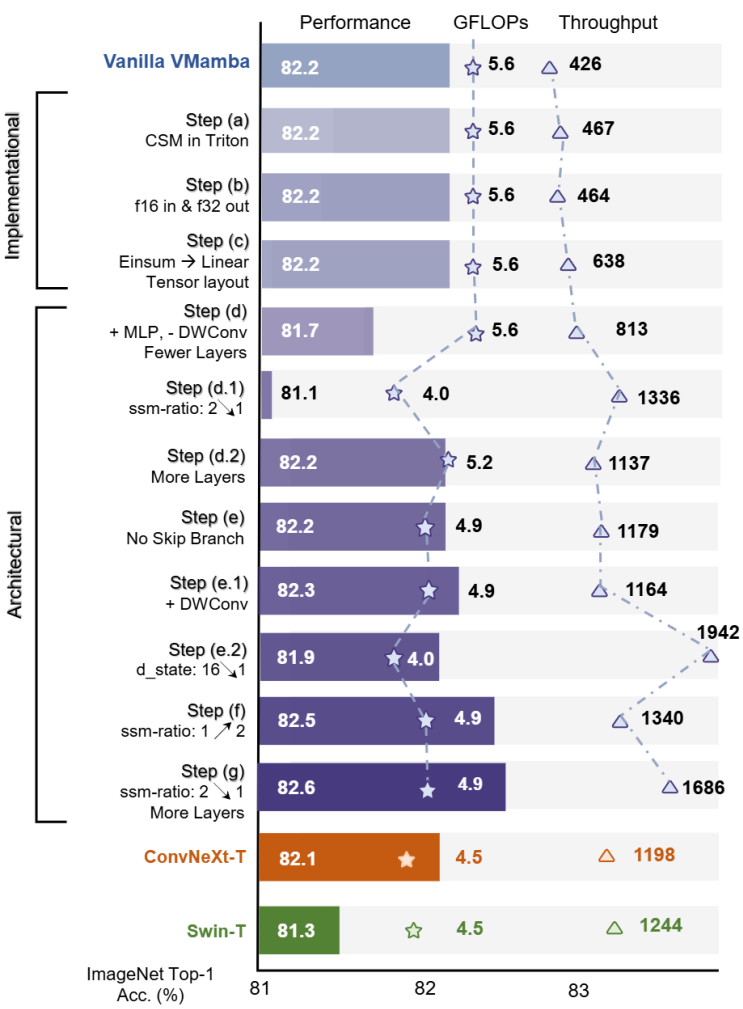
(a) Architecture of VMamba



(b) Mamba Block

(c) The Vanilla VSS Block

(d) VSS Block

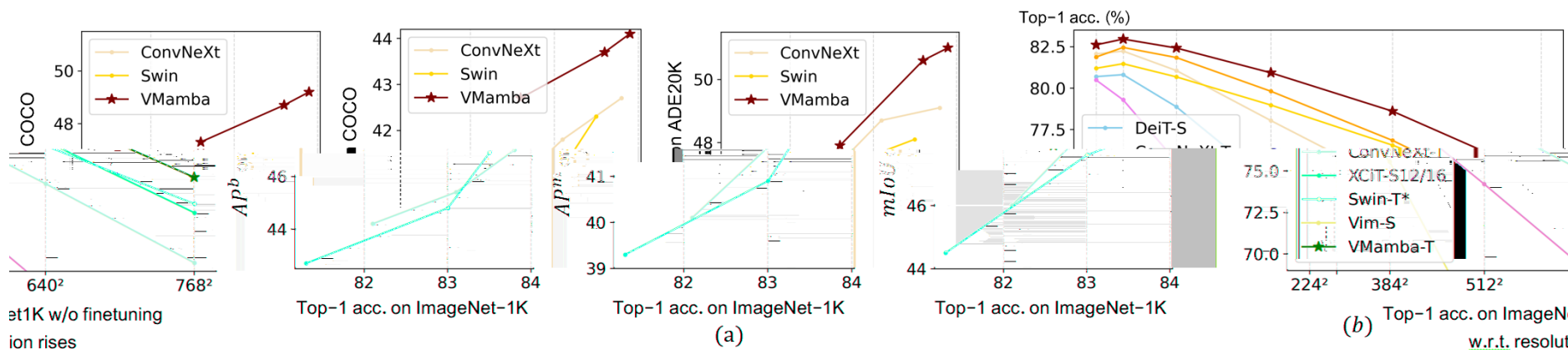


(e) Performance Comparison

ImageNet-1K				
Model	Params (M)	FLOPs (G)	TP. (img/s)	Top-1 (%)
Transformer-Based				
DeiT-S	22M	4.6G	1761	79.8
DeiT-B	86M	17.5G	503	81.8
HiViT-T	19M	4.6G	1393	82.1
HiViT-S	38M	9.1G	712	83.5
HiViT-B	66M	15.9G	456	83.8
Swin-T	28M	4.5G	1244	81.3
Swin-S	50M	8.7G	718	83.0
Swin-B	88M	15.4G	458	83.5
XCiT-S24	48M	9.2G	671	82.6
XCiT-M24	84M	16.2G	423	82.7
ConvNet-Based				
ConvNeXt-T	29M	4.5G	1198	82.1
ConvNeXt-S	50M	8.7G	684	83.1
ConvNeXt-B	89M	15.4G	436	83.8
SSM-Based				
S4ND-Conv-T	30M	5.2G	683	82.2
S4ND-ViT-B	89M	17.1G	397	80.4
Vim-S	26M	5.3G	811	80.5
VMamba-T	30M	4.9G	1686	82.6
VMamba-S	50M	8.7G	877	83.6
VMamba-B	89M	15.4G	646	83.9

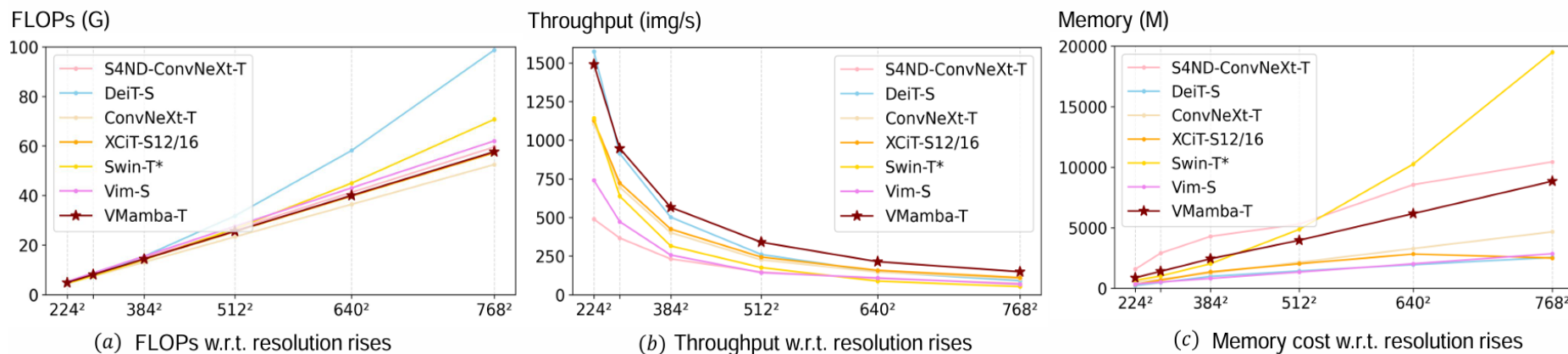


Down-stream Tasks



Mask R-CNN 1× schedule				
Backbone	AP ^b	AP ^m	Params	FLOPs
Swin-T	42.7	39.3	48M	267G
ConvNeXt-T	44.2	40.1	48M	262G
VMamba-T	47.3	42.7	50M	271G
Swin-S	44.8	40.9	69M	354G
ConvNeXt-S	45.4	41.8	70M	348G
VMamba-S	48.7	43.7	70M	349G
Swin-B	46.9	42.3	107M	496G
ConvNeXt-B	47.0	42.7	108M	486G
VMamba-B	49.2	44.1	108M	485G

Mask R-CNN 3× MS schedule				
Backbone	AP ^b	AP ^m	Params	FLOPs
Swin-T	46.0	41.6	48M	267G
ConvNeXt-T	46.2	41.7	48M	262G
VMamba-T	48.8	43.7	50M	271G
Swin-S	48.2	43.2	69M	354G
ConvNeXt-S	47.9	42.9	70M	348G
VMamba-S	49.9	44.2	70M	349G

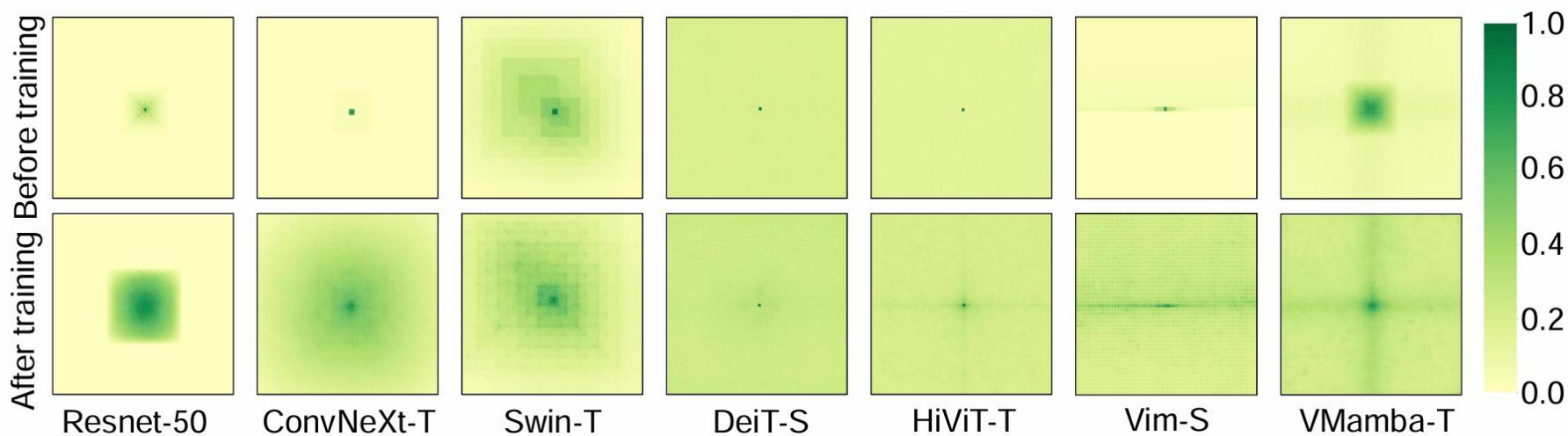
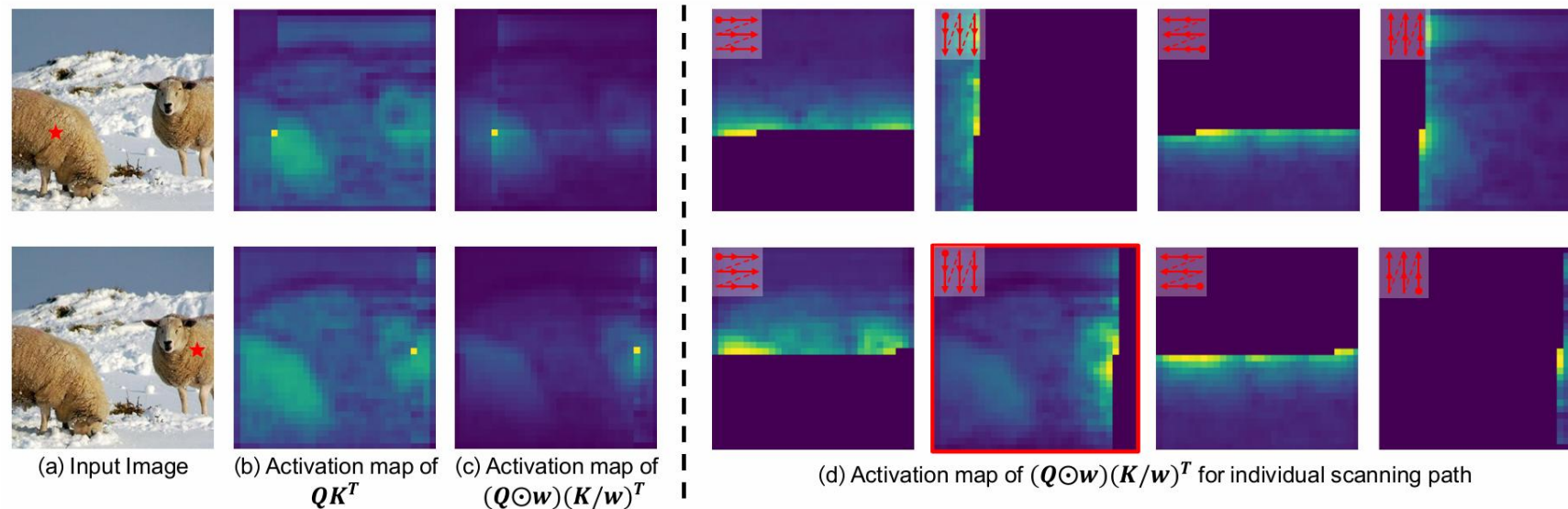


ADE20K with crop size 512				
Backbone	mIOU (SS)	mIOU (MS)	Params	FLOPs
Swin-T	44.5	45.8	60M	945G
ConvNeXt-T	46.0	46.7	60M	939G
Vim-S	44.9	-	46M	-
VMamba-T	47.9	48.8	62M	949G
Swin-S	47.6	49.5	81M	1039G
ConvNeXt-S	48.7	49.6	82M	1027G
VMamba-S	50.6	51.2	82M	1028G
Swin-B	48.1	49.7	121M	1188G
ConvNeXt-B	49.1	49.9	122M	1170G
VMamba-B	51.0	51.6	122M	1170G

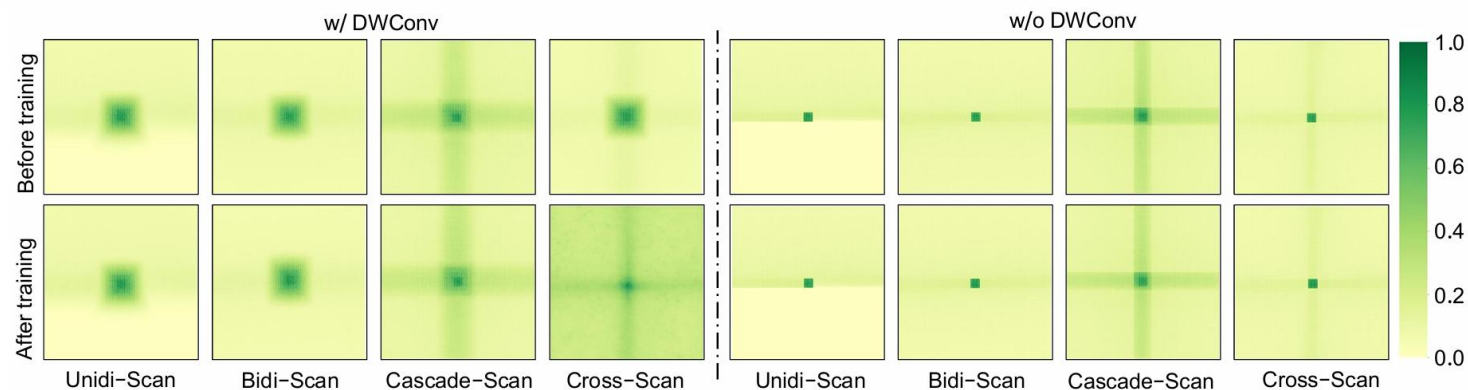
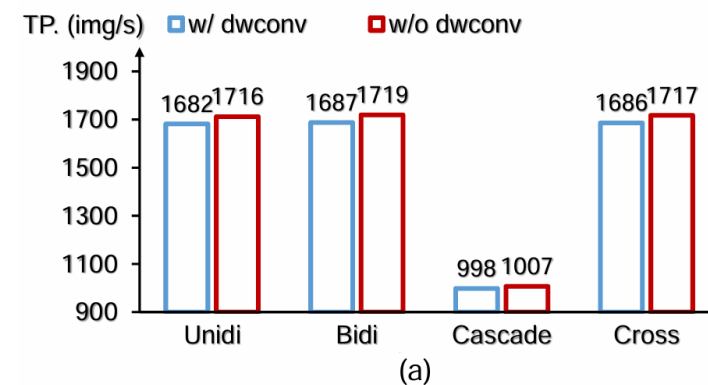
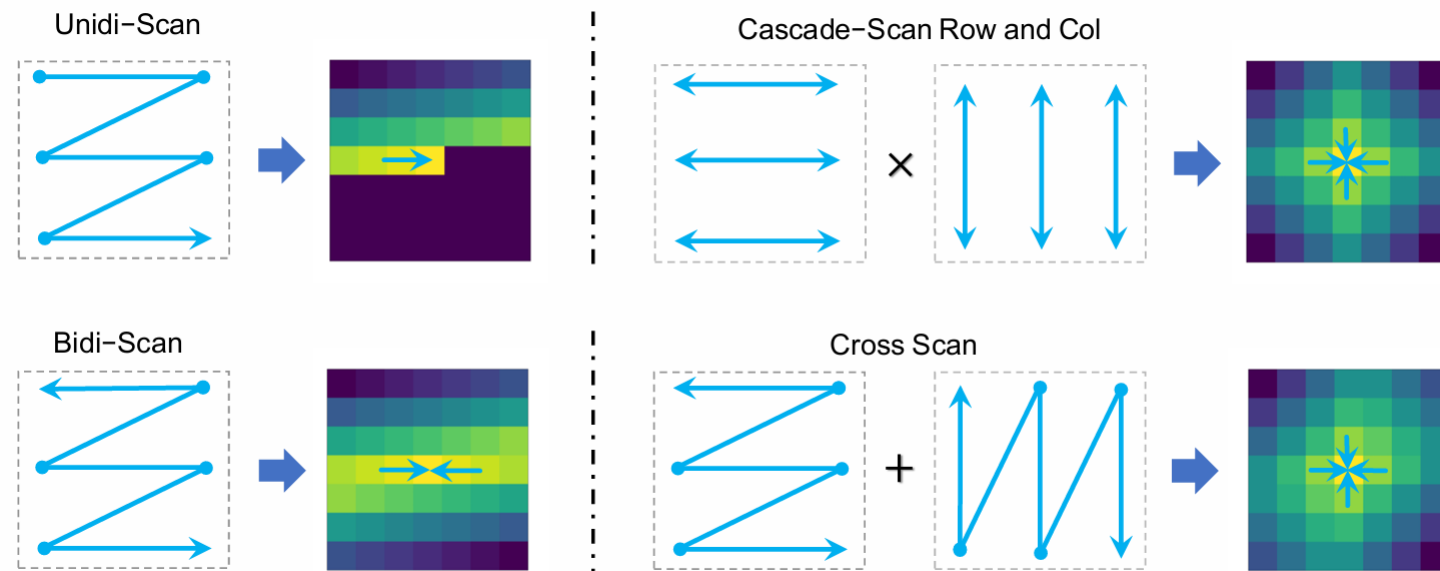


Activation Map & ERF

The selective scan mechanism allows VMamba to accumulate history along the scanning path, facilitating the establishment of long-term dependencies across image patches.



Different Scan patterns



Different Hyper-parameters

VMamba is robust across different initialization, activation, and learning rate.

The performance of VMamba-T with different initialization

initialization	Params (M)	FLOPs (G)	TP. (img/s)	Train TP. (img/s)	Top-1 acc. (%)
mamba	30.2	4.91	1686	571	82.60
rand	30.2	4.91	1682	570	82.58
zero	30.2	4.91	1683	570	82.67

The performance of VMamba-T with different activation functions

activation	Params (M)	FLOPs (G)	TP. (img/s)	Train TP. (img/s)	Top-1 acc. (%)
SiLU	30.2	4.91	1686	571	82.60
GELU	30.2	4.91	1680	570	82.53
ReLU	30.2	4.91	1684	577	82.65

The performance of VMamba-T with different learning rate

learning rate	Params (M)	FLOPs (G)	TP. (img/s)	Train TP. (img/s)	Top-1 acc. (%)
5e-4	30.2	4.91	1686	571	82.16
1e-3	30.2	4.91	1686	571	82.62
2e-3	30.2	4.91	1686	571	82.70

It is important to choose an optimal combination of ssm-ratio, mlp-ratio, and layer numbers for constructing a model that balances effectiveness and efficiency.

The trade-off between d_state and ssm-ratio with VMamba-T

d_state	ssm-ratio	Params (M)	FLOPs (G)	TP. (img/s)	Train TP. (img/s)	Top-1 acc. (%)
1	2.0	30.7	4.86	1340	464	82.49
2	2.0	30.8	4.98	1269	432	82.50
4	2.0	31.0	5.22	1147	382	82.51
8	1.5	28.6	5.04	1148	365	82.69
16	1.0	26.3	4.87	1164	358	82.31

The trade-off between layer numbers and ssm-ratio with VMamba-T

layer numbers	ssm-ratio	Params (M)	FLOPs (G)	TP. (img/s)	Train TP. (img/s)	Top-1 acc. (%)
[2,2,5,2]	2.0	30.7	4.86	1340	464	82.49
[2,2,5,2]	1.0	25.6	3.98	1942	647	81.87
[2,2,8,2]	1.0	30.2	4.91	1686	571	82.60

The trade-off between mlp-ratio and ssm-ratio with VMamba-T

mlp-ratio	ssm-ratio	Params (M)	FLOPs (G)	TP. (img/s)	Train TP. (img/s)	Top-1 acc. (%)
4.0	1.0	30.2	4.91	1686	571	82.60
3.0	1.5	28.5	4.65	1419	473	82.75
2.0	2.5	29.9	4.95	1075	361	82.86



Thank you

