

---

# VMamba:视觉状态空间模型

---

刘悦<sup>1</sup>田云杰<sup>1</sup>赵玉中<sup>1</sup>余洪天<sup>1</sup>谢凌曦<sup>2</sup>王耀伟<sup>3</sup>叶启祥<sup>1</sup>谋建斌<sup>1</sup>刘云帆<sup>1‡</sup>

<sup>1</sup> UCAS <sup>2</sup>华为公司 <sup>3</sup>彭程实验室。

{liuyue171,tianyunjie19,zhaoyuzhong20,yuhongtian17}@mails.ucas.ac.cn

198808xc@gmail.com, wangyw@pcl.ac.cn, {qxye,jiaojb,liuyunfan}@ucas.ac.cn

## 摘要

在计算机视觉领域，设计计算高效的网络架构始终是持续发展的需求。本文将状态空间语言模型Mamba升级为具有线性时间复杂度的视觉骨干网络VMamba。VMamba的核心是包含二维选择性扫描（SS2D）模块的视觉状态空间（VSS）模块堆栈。通过沿四条扫描路径遍历，SS2D模块有效弥合了一维选择性扫描的有序特性与二维视觉数据的非序列结构之间的差距，从而促进从多源多视角采集上下文信息。基于VSS模块，我们开发了一系列VMamba架构，并通过架构优化和实现改进实现加速。大量实验表明，VMamba在多种视觉感知任务中展现出优异性能，其输入缩放效率显著优于现有基准模型。源代码可通过<https://github.com/MzeroMiko/VMamba>获取。

## 1 介绍

视觉表征学习仍然是计算机视觉领域的一个基础研究方向，在深度学习时代取得了显著进展。为了表征视觉数据中的复杂模式，已提出并广泛应用于各类视觉任务的两种主要骨干网络类别：即卷积神经网络（CNNs）[49、27、29、53、37]和视觉Transformer（ViTs）[13、36、57、66]。与CNN相比，ViTs通常在大规模数据上展现出更优的学习能力，这得益于其自注意力机制的整合[58、13]。然而，自注意力机制对标记数量的二次复杂度在涉及大空间分辨率的下游任务中带来了巨大的计算开销。

为应对这一挑战，学界已投入大量精力提升注意力计算效率[54、36、12]。然而现有方法要么限制有效感受野的尺寸[36]，要么在各类任务中出现显著性能下降[30、60]。这促使我们开发一种新型视觉数据架构，在保留原生自注意力机制固有优势（即全局感受野与动态加权参数）的同时[23]。

最近，Mamba[17]作为自然语言处理（NLP）领域中一种具有线性复杂度的长序列建模创新方法（状态空间模型 SSM）[17、43、59、71、48]，已成为极具前景的解决方案。受此进展启发，我们提出VMamba——一种整合基于SSM的模块以实现高效视觉表征学习的视觉骨干网络。然而，Mamba的核心算法即并行化选择性扫描操作本质上是为处理一维序列数据而设计的。这在将其适配于缺乏视觉组件固有序列排列的视觉数据处理时带来了挑战。为解决该问题，我们提出二维选择性扫描（SS2D）——一种专为空间域遍历设计的四向扫描机制。与自注意力机制（图1(a)）不同，SS2D确保

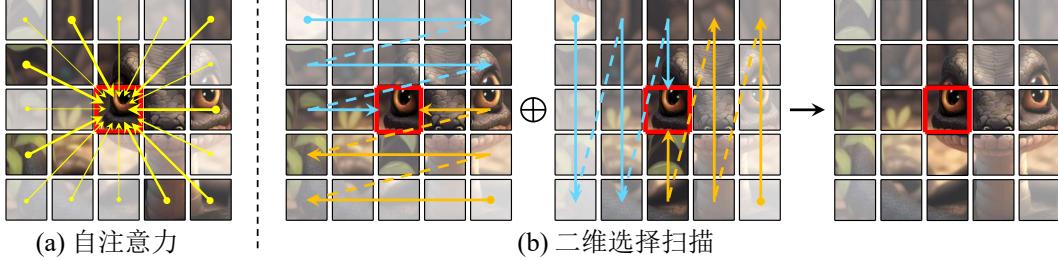


图1：通过(a)自注意力机制与(b)所提出的二维选择性扫描（SS2D）建立图像块间相关性的对比。红色方框标示查询图像块，其不透明度表示信息丢失程度。

每个图像块仅通过沿其对应扫描路径计算的压缩隐藏状态（图1(b)）获取上下文知识，从而将计算复杂度从二次方降低至线性。

基于VSS模块，我们开发了一组VMamba架构（即VMambaTiny/Small/Base），并通过架构改进和实现优化提升其性能。与基于CNN的基准视觉模型（ConvNeXt[37]）、ViTs（Swin[36]、HiViT[66]）以及SSMs（S4ND[44]、Vim[69]）相比，VMamba在ImageNet-1K[9]上始终在不同模型规模下实现更高的图像分类准确率。具体而言，VMamba-Base达到83.9%的top-1准确率，比Swin高出0.4%，吞吐量更是以40%以上的显著优势超越Swin（646 vs. 458）。VMamba的优越性延伸至多个下游任务，VMamba-Tiny/Small/Base在COCO[33]上物体检测任务中分别获得47.3%/48.7%/49.2%的mAP（1 × 训练计划），分别比Swin和ConvNeXt提升4.6%/3.9%/2.3%和3.1%/3.3%/2.2%。至于ADE20K[68]的单尺度语义分割任务，VMamba-Tiny/Small/Base达到47.9%/50.6%/51.0%的mIoU，比Swin提升3.4%/3.0%/2.9%和ConvNeXt分别达到1.9%/1.9%/1.9%。此外，与基于ViT的模型不同（其计算复杂度随输入标记数量呈二次增长），VMamba在保持性能相当的同时，其浮点运算次数（FLOPs）仅呈线性增长。这充分展现了其在输入可扩展性方面的前沿优势。

本研究的主要贡献总结如下：

- 我们提出VMamba，一种基于SSM的视觉骨干网络，用于视觉表征学习且具有线性时间复杂度。通过采用一系列架构和实现改进，以提升VMamba的推理速度。
- 我们引入二维选择性扫描（SS2D）技术，以连接一维阵列扫描与二维平面遍历，从而扩展选择性空间选择性匹配（SSMs）在处理视觉数据中的应用。
- VMamba在图像分类、目标检测和语义分割等多种视觉任务中均展现出优异性能。该模型还具有显著的输入序列长度适应性，其计算复杂度仅呈现线性增长。

## 2 相关工作

**卷积神经网络（CNNs）**。自AlexNet[31]问世以来，学界持续投入大量精力提升基于CNN模型在各类视觉任务中的建模能力[49、52、27、29]与计算效率[28、53、64、46]。为增强CNN的灵活性与效能，深度卷积[28]与可变形卷积[5、70]等复杂运算已被引入。近期受Transformer[58]成功启发，现代CNN[37]通过在其架构中整合长距离依赖关系[11、47、34]与动态权重[23]展现出优异性能。

**视觉Transformer（ViTs）**。作为一项开创性研究，ViT[13]探索了基于标准Transformer架构的视觉模型的有效性，强调了大规模

为提升图像分类性能的预训练。为降低ViT对大型数据集的依赖，DeiT[57]引入了师生蒸馏策略，将CNN的知识迁移至ViTs，并强调视觉感知中归纳偏置的重要性。基于此方法，后续研究提出了分层ViTs[36、12、61、39、66、55、6、10、67、1]。

另一个研究方向聚焦于提升自注意力机制的计算效率，该机制是视觉Transformer（ViTs）的核心。线性注意力[30]通过将自注意力重构为核特征图的线性点积，利用矩阵乘积的结合律特性将计算复杂度从二次方降低至线性。GLA[65]提出了一种硬件高效的线性注意力变体，平衡了内存移动与并行化能力。RWKV[45]同样利用线性注意力机制，将可并行化的Transformer训练与循环神经网络（RNNs）的高效推理相结合。RetNet[51]通过引入门控机制实现可并行化的计算路径，为循环机制提供了替代方案。RMT[15]进一步将时间衰减机制应用于空间域，拓展了视觉表征学习的边界。

**状态空间模型（SSMs）**。尽管在视觉任务中被广泛采用，但由于自注意力的二次复杂度，ViT架构在处理长输入序列（例如高分辨率图像）时面临重大挑战。为提高扩展效率[8、7、45、51、41]，SSMs已成为Transformer的有力替代方案，吸引了大量研究关注。Gu等人[21]展示基于SSM的模型在使用HiPPO初始化处理长程依赖方面的潜力[18]。为提升实际可行性，S4[20]提出将参数矩阵归一化为对角结构。此后涌现出多种结构化SSM模型，每种模型都提供了独特的架构增强功能，例如复杂对角结构[22, 19]、支持多输入多输出[50]、对角加低秩分解[24]以及选择机制[17]。这些进展也被整合到更大的表征模型中[43, 41, 16]，进一步凸显了结构化状态空间模型在各类应用中的通用性和可扩展性。尽管这些模型主要针对文本和语音等长程及序列数据，但关于将SSMs应用于具有二维结构的视觉数据的研究仍较为有限。

### 3 前言

**SSMs的构建。**SSMs源自卡尔曼滤波器[32]，是线性时不变（LTI）系统，通过隐藏状态 $\mathbf{h}(t) \in \mathbb{R}^N$ 将输入信号 $u(t) \in \mathbb{R}$ 映射到输出响应 $y(t) \in \mathbb{R}$ 。具体而言，连续时间SSMs可表示为以下线性常微分方程（ODEs）：

$$\begin{aligned}\mathbf{h}'(t) &= \mathbf{Ah}(t) + \mathbf{Bu}(t), \\ y(t) &= \mathbf{Ch}(t) + Du(t),\end{aligned}\tag{1}$$

其中 $\mathbf{A} \in \mathbb{R}^{N \times N}$ 、 $\mathbf{B} \in \mathbb{R}^{N \times I}$ 、 $\mathbf{C} \in \mathbb{R}^{1 \times N}$ 以及 $D \in \mathbb{R}^1$ 是权重参数。

**SSM的离散化。**为了集成到深度模型中，连续时间SSMs必须预先进行离散化。具体而言，对于时间区间 $[t_a, tb]$ ，隐藏状态变量 $\mathbf{h}(t)$ 在 $t=t_b$ 时刻的解析解可表示为

$$\mathbf{h}(tb) = e^{\mathbf{A}(tb-ta)} \mathbf{h}(ta) + e^{\mathbf{A}(tb-ta)} \int_{ta}^{tb} \mathbf{B}(\tau) u(\tau) e^{-\mathbf{A}(\tau-ta)} d\tau.\tag{2}$$

通过使用时间尺度参数 $\Delta$ 进行采样（即， $d\tau | t_i t_{i+1} = \Delta$ ）， $\mathbf{h}(tb)$ 可通过

$$\mathbf{h}(tb) = e^{\mathbf{A}(\Delta a + \dots + \Delta b - 1)} \left( \mathbf{h}_a + \sum_{i=a}^{b-1} \mathbf{B}_i u_i e^{-\mathbf{A}(\Delta a + \dots + \Delta i) \Delta} \dot{\mathbf{i}} \right)', \tag{3}$$

其中 $[a, b]$ 为对应的离散步长区间。值得注意的是，该公式近似了零阶保持（ZOH）方法所得结果，该方法在基于SSM的模型文献中被广泛采用（具体证明详见附录A）。

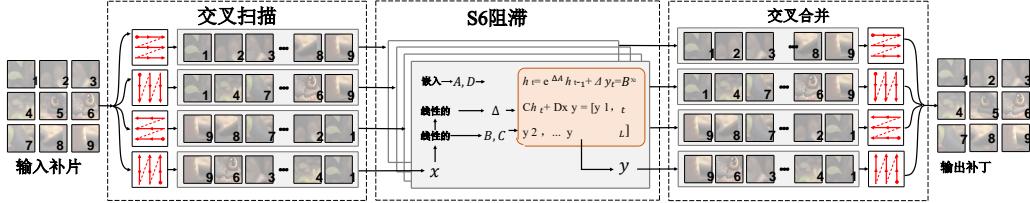


图2：二维选择性扫描（SS2D）示意图。输入图像块沿四条不同扫描路径（交叉扫描）进行遍历，每条序列由独立的S6模块分别处理。随后将结果合并以构建二维特征图作为最终输出（交叉合并）。

**选择性扫描机制。**为解决 LTI SSMs (1) 在捕获上下文信息方面的局限性，Gu 等人 [17] 提出一种新颖的SSMs参数化方法，该方法整合了输入依赖的选择机制（称为S6）。然而，对于选择性SSMs而言，时变权重参数对隐藏状态的高效计算构成挑战，因为卷积无法容纳动态权重，使其不适用。尽管如此，由于等式3中 $h_b$ 的递推关系可以推导出来，响应 $y_b$ 仍可使用关联扫描算法[2、42、50]高效计算，该算法具有线性复杂度（详见附录B）。

## 4 VMamba:视觉状态空间模型

### 4.1 网络体系结构

我们在三个尺度上开发了VMamba: Tiny、Small和Base（分别称为VMamba-T、VMamba-S和VMamba-B）。VMamba-T的架构概述如图3(a)所示，详细配置见附录E。输入图像 $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ 首先通过茎模块分割成图像块，生成空间维度为 $H/4 \times W/4$ 的二维特征图。在不引入额外位置嵌入的情况下，采用多网络阶段构建分层表征，其分辨率分别为 $H/8 \times W/8$ 、 $H/16 \times W/16$ 和 $H/32 \times W/32$ 。具体而言，每个阶段包含一个下采样层（首阶段除外），随后接视觉状态空间（VSS）模块堆栈。

VSS模块作为Mamba模块[17]（图3(b)）在表征学习中的视觉对应物。VSS模块的初始架构（图3(c)中称为‘基础VSS模块’）通过替换S6模块构建而成。S6是Mamba的核心模块，实现了全局感受野、动态权重（即选择性）和线性复杂度。我们用新提出的二维选择性扫描（SS2D）模块替代了它，更多细节将在下一小节介绍。为进一步提升计算效率，我们移除了整个乘法分支（图3(c)中红色框标注的部分），因为SS2D的选择性已经实现了门控机制的效果。改进后的VSS模块（图3(d)所示）由单个网络分支和两个残差模块组成，模拟了基础Transformer模块[58]的架构。本文所有结果均基于采用该架构的VSS模块构建的VMamba模型获得。

### 4.2 视觉数据的二维选择扫描 (SS2D)

虽然S6中扫描操作的顺序性与涉及时间数据的自然语言处理任务高度契合，但当应用于视觉数据时却面临重大挑战——视觉数据本质上是非顺序性的，并包含空间信息（例如局部纹理和全局结构）。为解决这一问题，S4ND[44]通过卷积运算重构 SSM，借助外积运算将核函数从一维直接扩展到二维。然而这种修改限制了权重的输入依赖性，导致其捕捉上下文信息的能力受限。因此，我们坚持采用选择性扫描方法[17]进行输入处理，并提出二维选择性扫描（SS2D）模块，使S6在适应视觉数据的同时仍保持其优势。

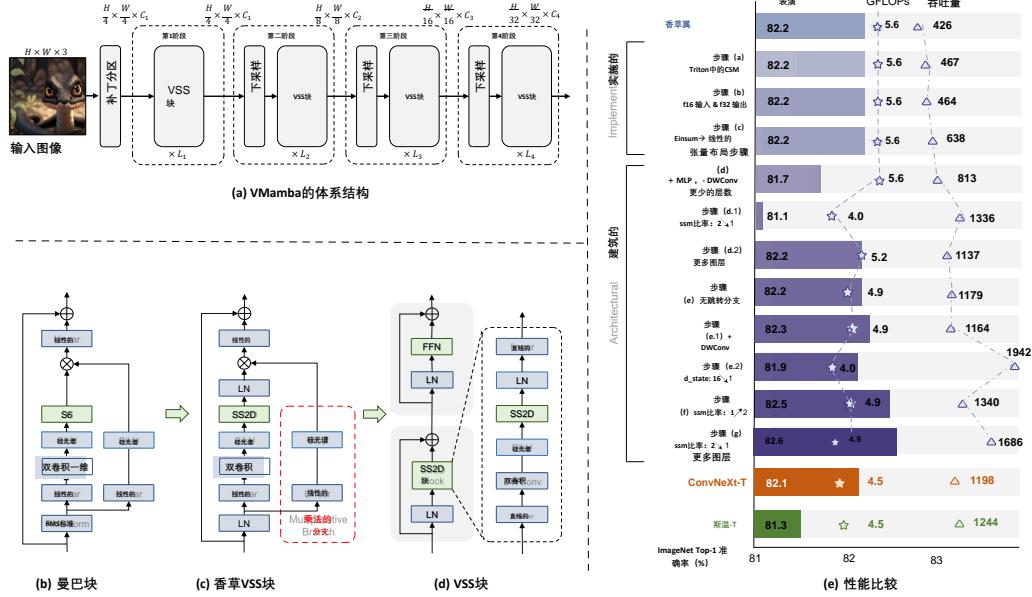


图3：左图：(a)VMamba整体架构示意图，(b)-(d)Mamba与VSS模块结构示意图。右图：VMamba各变体与基准方法在分类准确率和计算效率方面的对比。

图2展示了SS2D的数据转发流程包含三个关键步骤：交叉扫描、S6模块选择性扫描以及交叉合并。具体而言，SS2D首先将输入图像块沿四条不同遍历路径展开（即交叉扫描）。随后每个图像块序列通过独立的S6模块并行处理，最终生成的序列经过重构和合并形成输出图（即交叉合并）。通过采用互补的一维遍历路径，SS2D使图像中每个像素都能整合来自不同方向的所有其他像素信息。这种信息整合机制有助于在二维空间中建立全局感受野。

### 4.3 加速VMamba

如图3(e)所示，采用标准VSS模块的VMamba-T模型（简称‘标准VMamba’）以426张图像/秒的吞吐量运行，包含2290万参数和56亿浮点运算量。尽管该模型在微尺度分类任务中取得82.2%的最高准确率（较Swin-T[36]提升0.9个百分点），但其低吞吐量和高内存开销仍给实际部署带来显著挑战。

本小节阐述了我们提升模型推理速度的改进方案，重点聚焦于实现细节与架构设计的优化。我们通过ImageNet-1K数据集的图像分类任务对模型进行评估，各项渐进式改进的效果总结如下：其中（%、img/s）分别表示ImageNet-1K数据集上top-1准确率和推理吞吐量的提升幅度。更多讨论详见附录E。

步骤(a) (+0.0%， +41 img/s) 通过在Triton中重新实现交叉扫描（Cross-Scan）和交叉合并（Cross-Merge）完成。

步骤(b) (+0.0%， -3img/s) 通过调整选择性扫描的CUDA实现以适配float16输入和float32输出。这显著提升了训练效率（吞吐量从165提升至184），尽管测试时速度略有波动。

步骤(c) (+0.0%， +174 img/s) 通过将选择性扫描中相对缓慢的einsum替换为线性变换（即torch.nn.functional.linear）实现。我们还采用(B, C, H, W)的张量布局以消除不必要的数据置换。

表1: ImageNet-1K上的性能比较。吞吐量值是使用A100 GPU和AMD EPYC 7542 CPU测量的, 使用[62]发布的工具包, 遵循[36]中提出的协议。所有图像的尺寸为 $224 \times 224$ 。

| 模型           | 参数浮点运算次数<br>数(M) | 东帝汶<br>(G) | 第1名<br>(%) | 模型   | 参数浮点运算<br>次数(G)  | 东帝汶<br>(img/s) | 第1名<br>(%) |      |
|--------------|------------------|------------|------------|------|------------------|----------------|------------|------|
| 基于变分区的       |                  |            |            |      |                  |                |            |      |
| DeiT-S [57]  | 22M              | 4.6G       | 1761       | 79.8 | ConvNeXt-T [37]  | 29M            | 4.5G       | 1198 |
| DeiT-B [57]  | 86M              | 17.5G      | 503        | 81.8 | ConvNeXt-S [37]  | 50M            | 8.7G       | 684  |
| HiViT-T [66] | 19M              | 4.6G       | 1393       | 82.1 | ConvNeXt-B [37]  | 89M            | 15.4G      | 436  |
| HiViT-S [66] | 38M              | 9.1G       | 712        | 83.5 | SSM 基础           |                |            |      |
| HiViT-B [66] | 66M              | 15.9G      | 456        | 83.8 | S4ND-Conv-T [44] | 30M            | 5.2G       | 683  |
| Swin-T [36]  | 28M              | 4.5G       | 1244       | 81.3 | S4ND-ViT-B [44]  | 89M            | 17.1G      | 397  |
| Swin-S [36]  | 50M              | 8.7G       | 718        | 83.0 | Vim-S [69]       | 26M            | 5.3G       | 811  |
| Swin-B [36]  | 88M              | 15.4G      | 458        | 83.5 | 虚拟磁带管理器          | 30M            | 4.9G       | 1686 |
| XCiT-S24 [1] | 48M              | 9.2G       | 671        | 82.6 | 虚拟磁带存储器          | 50M            | 8.7G       | 877  |
| XCiT-M24 [1] | 84M              | 16.2G      | 423        | 82.7 | VMamba-B         | 89M            | 15.4G      | 646  |
| 基于卷积网络的      |                  |            |            |      |                  |                |            |      |

步骤(d) (-0.6%, +175 img/s) 通过引入 MLP 到 VMamba 中, 因其计算效率较高。我们还移除了 DWConv (深度卷积[23]) 层, 并将层配置从 [2,2,9,2] 调整为 [2,2,2,2] 以降低浮点运算次数。

步骤(e) (+0.6%, +366 img/s) 通过将参数 ssm-ratio (特征扩展因子) 从 2.0 降至 1.0 (亦称步骤 (d.1)) , 将层数提升至 [2,2,5,2] (亦称步骤 (d.2)) , 并如图3(c)所示舍弃整个乘法分支来实现。

步骤(f) (+0.3%, +161img/s) 通过引入 DWConv 层 (亦称步骤 (e.1)) 并将参数 d\_state (SSM 状态维度) 从 16.0 降至 1.0 (亦称步骤 (e.2)) , 同时将 ssm-ratio 恢复至 2.0。

步骤(g) (+0.1%, +346 img/s) 通过将 ssm 比值降至 1.0, 同时将层配置从 [2,2,5,2] 更改为 [2,2,8,2]。

## 5 实验

在本节中, 我们通过一系列实验评估 VMamba 的性能表现, 并将其与各类视觉任务中的主流基准模型进行对比。同时, 我们通过与替代方法的对比验证了所提出的二维特征图遍历方法的有效性。此外, 我们通过可视化有效感受野 (ERF) 和激活图, 分析了 VMamba 的特征, 并考察了其在处理更长输入序列时的扩展性。实验主要沿用了 Swin[36] 中的超参数设置和实验配置。具体实验设置详见附录E和F, 更多消融实验结果参见附录H。所有实验均在配备  $8 \times$  NVIDIA 特斯拉 A100 GPU 的服务器上完成。

### 5.1 图像分类

我们在 ImageNet-1K[9] 数据集上评估了 VMamba 的图像分类性能, 并将对比结果与表1中总结的基准方法进行比较。在计算量相近的情况下, VMamba-T 模型的 top-1 准确率达到 82.6%, 较 DeiT-S 提升 2.8%, 较 Swin-T 提升 1.3%。值得注意的是, VMamba 在小尺度和大尺度数据集上均保持性能优势。例如, VMamba-B 模型的 top-1 准确率达到 83.9%, 较 DeiT-B 提升 2.1%, 较 Swin-B 提升 0.4%。

在计算效率方面, VMamba-T 实现了 1,686 张图像/秒的吞吐量, 这一性能表现优于或可与当前最先进的方法相媲美。VMamba-S 和 VMamba-B 延续了这一优势, 分别达到 877 张图像/秒和 646 张图像/秒的吞吐量。相较于基于 SSM 的模型, VMamba-T 的吞吐量比 S4ND-Conv-T [44] 高出 1.47 ×, 比 Vim-S[69] 高出 1.08 ×, 同时分别保持了 0.4% 和 2.1% 的明显性能优势。

表2：左侧：MSCOCO 上的目标检测与实例分割结果。 $AP^b$  和  $AP^m$  分别表示框 AP 和 掩码 AP。FLOPs 计算时输入尺寸为  $1280 \times 800$ 。符号 ‘ $1 \times$ ’ 表示经过 12 个周期微调的模型，而 ‘ $3 \times MS$ ’ 表示经过 36 个周期多尺度训练。右侧：ADE20K 上的语义分割结果。FLOPs 计算时输入尺寸为  $512 \times 2048$ 。‘SS’ 和 ‘MS’ 分别表示单尺度与多尺度测试。

| Mask R-CNN 1 × 计划    |                 |      |      | ADE20K, 裁切尺寸 512 |              |    |        |
|----------------------|-----------------|------|------|------------------|--------------|----|--------|
|                      | AP <sup>b</sup> | APM  | 参数   | mIoU<br>(SS)     | mIoU<br>(MS) | 参数 | 浮点运算次数 |
| 脊梁骨                  | 42.7            | 39.3 | 48M  | 267G             |              |    |        |
| Swin-T               | 44.2            | 40.1 | 48M  | 262G             |              |    |        |
| ConvNeXt-T           | 47.3            | 42.7 | 50M  | 271G             |              |    |        |
| 虚拟磁带管理               | 44.8            | 40.9 | 69M  | 354G             |              |    |        |
| 猪                    | 45.4            | 41.8 | 70M  | 348G             |              |    |        |
| ConvNeXt-S           | 48.7            | 43.7 | 70M  | 349G             |              |    |        |
| 虚拟磁带存                | 46.9            | 42.3 | 107M | 496G             |              |    |        |
| 储器                   | 47.0            | 42.7 | 108M | 486G             |              |    |        |
| VMamba-B             | 49.2            | 44.1 | 108M | 485G             |              |    |        |
| Mask R-CNN 3 × MS 调度 |                 |      |      | ADE20K, 裁切尺寸 512 |              |    |        |
| Swin-T               | 46.0            | 41.6 | 48M  | 267G             |              |    |        |
| ConvNeXt-T           | 46.2            | 41.7 | 48M  | 262G             |              |    |        |
| NAT-T                | 47.7            | 42.6 | 48M  | 258G             |              |    |        |
| 虚拟磁带管理               | 48.8            | 43.7 | 50M  | 271G             |              |    |        |
| 猪                    | 48.2            | 43.2 | 69M  | 354G             |              |    |        |
| ConvNeXt-S           | 47.9            | 42.9 | 70M  | 348G             |              |    |        |
| NAT-S                | 48.4            | 43.2 | 70M  | 330G             |              |    |        |
| 虚拟磁带存                | 49.9            | 44.2 | 70M  | 349G             |              |    |        |
| 储器                   |                 |      |      |                  |              |    |        |
| ResNet-50            | 42.1            | 42.8 | 67M  | 953G             |              |    |        |
| DeiT-S + MLN         | 43.8            | 45.1 | 58M  | 1217G            |              |    |        |
| Swin-T               | 44.5            | 45.8 | 60M  | 945G             |              |    |        |
| ConvNeXt-T           | 46.0            | 46.7 | 60M  | 939G             |              |    |        |
| NAT-T                | 47.1            | 48.4 | 58M  | 934G             |              |    |        |
| Vim-S                | 44.9            | -    | 46M  | -                |              |    |        |
| 虚拟磁带管理器              | 47.9            | 48.8 | 62M  | 949G             |              |    |        |
| ResNet-101           | 43.8            | 44.9 | 86M  | 1030G            |              |    |        |
| DeiT-B + MLN         | 45.5            | 47.2 | 144M | 2007G            |              |    |        |
| 猪-                   | 47.6            | 49.5 | 81M  | 1039G            |              |    |        |
| ConvNeXt-S           | 48.7            | 49.6 | 82M  | 1027G            |              |    |        |
| NAT-S                | 48.0            | 49.5 | 82M  | 1010G            |              |    |        |
| 虚拟磁带存储器              | 50.6            | 51.2 | 82M  | 1028G            |              |    |        |
| 猪B型                  | 48.1            | 49.7 | 121M | 1188G            |              |    |        |
| ConvNeXt-B           | 49.1            | 49.9 | 122M | 1170G            |              |    |        |
| NAT-B                | 48.5            | 49.7 | 123M | 1137G            |              |    |        |
| RepLKNet-31B         | 49.9            | 50.6 | 112M | 1170G            |              |    |        |
| VMamba-B             | 51.0            | 51.6 | 122M | 1170G            |              |    |        |

## 5.2 下游任务

在本小节中，我们评估了 VMamba 在下游任务上的性能，包括在 MSCOCO2017[33] 上的目标检测和实例分割，以及在 ADE20K[68] 上的语义分割。训练框架基于 MMDetection[3] 和 MM Segmentation[4] 库，遵循[35] 的方法，分别采用 Mask R-CNN[26] 和 UperNet[63] 作为检测和分割网络。

**目标检测与实例分割。** 表2展示了 MSCOCO 上的结果。VMamba 在不同训练方案下均展现出更优的框平均精度 ( $AP^b$ ) 和掩码平均精度 ( $AP^m$ )。在 12 轮微调方案下，VMamba-T/S/B 的检测 mAP 分别达到 47.3%/48.7%/49.2%，较 Swin-T/S/B 提升 4.6%/3.9%/2.3%，较 ConvNeXt-T/S/B 提升 3.1%/3.3%/2.2%。其实例分割 mAP 则分别超越 Swin-T/S/B 3.4%/2.8%/1.8%，以及 ConvNeXt-T/S/B 2.6%/1.9%/1.4%。此外，采用多尺度训练的 36 轮微调方案中，VMamba 的优势依然显著，充分展现了其在需要密集预测的下游任务中的强大潜力。

**语义分割。** 与先前实验结果一致，VMamba 在 ADE20K 数据集上展现出更优的语义分割性能，且参数量相当。如表2所示，在单尺度 (SS) 设置下，VMamba-T 的 mIoU 值比 Swin-T 高 3.4%，比 ConvNeXt-T 高 1.9%；这种优势在多尺度 (MS) 输入时依然存在。对于小尺度和基础层级模型，VMamba-S/B 在 SS 设置下较 NAT-S/B[25] 的 mIoU 分别提升 2.6%/2.5%，在 MS 设置下提升 1.7%/1.9%。

讨论本小节的实验结果展示了 VMamba 在目标检测、实例分割和语义分割任务中的适应性。图4(a) 中，我们通过对 VMamba 与 Swin、ConvNeXt 的性能，突显了其在处理下游任务时的优势——在 ImageNet-1K 数据集上保持了相当的分类准确率。这一结果与图4(b) 相呼应：VMamba 在不同输入图像尺寸下展现出最稳定的性能（即，性能下降幅度较小），在 768 × 768 的输入分辨率下，无需微调即可达到 74.7% 的 top-1 分类准确率（线性调优时为 79.2%）。尽管 VMamba 对变化表现出更强的适应性

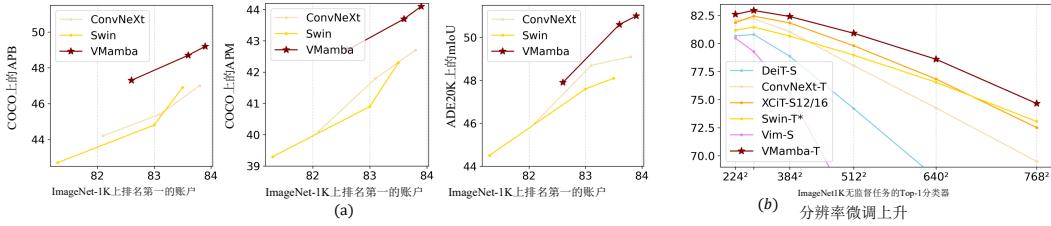


图4: VMamba对(a)下游任务及(b)分辨率逐步升高的输入图像的适应性示意图。Swin-T\*表示采用缩放窗口尺寸测试的Swin-T。

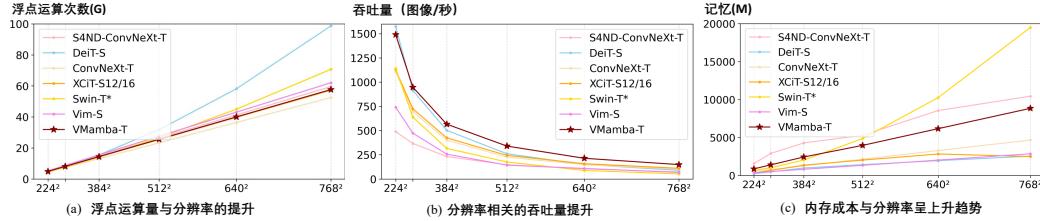


图5: VMamba资源消耗随分辨率逐步增加的示意图。Swin-T\*表示采用不同窗口尺寸测试的Swin-T。

在输入分辨率方面，VMamba在浮点运算次数（FLOPs）和内存消耗（见图5(a)和(c)）上保持线性增长，并维持高吞吐量（图5(b)），这使得其在适应具有更高空间分辨率输入的下游任务时，相较于基于ViT的方法更具优势和效率。这与Mamba在高效长序列建模方面的先进能力相一致[17]。

### 5.3 分析

**SS2D与自注意力的关系。**为在长度为 $T$ 的时间区间 $[a, b]$ 内构建响应 $\mathbf{Y}$ ，我们将对应的SSM相关变量 $\mathbf{u}_i, \theta, \Delta_i \in \mathbb{R}^{1 \times D_v}$ 、 $\mathbf{B}_i \in \mathbb{R}^{1 \times D_k}$ 以及 $\mathbf{C}_i \in \mathbb{R}^{1 \times D_k}$ 分别表示为 $\mathbf{v} \in \mathbb{R}^{T \times D_v}$ 、 $\mathbf{k} \in \mathbb{R}^{T \times D_k}$ 和 $\mathbf{q} \in \mathbb{R}^{T \times D_k}$ 。因此， $\mathbf{y}_b$ 沿维度 $D_v$ 的第 $j$ 个切片，记为 $\mathbf{y}_{b(j)} \in \mathbb{R}$ ，可表示为

$$\mathbf{y}_{b(j)} = (\mathbf{Q}_T \odot \mathbf{w}_T^{(j)}) \mathbf{h}_a^{(j)} + \mathbf{q} \tau \sum_{i=1}^T \left( \frac{\mathbf{w}_T^{(j)}}{\mathbf{w}_i^{(j)}} \odot \mathbf{k}^i \right)^T \odot (\mathbf{v}_i^{(j)}). \quad (4)$$

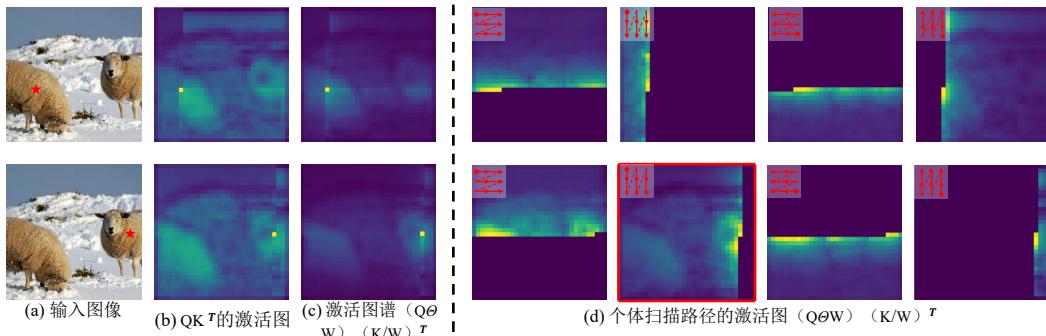


图6: 红色星号标示的查询补丁激活图示。图(b)和(c)中的可视化结果是通过整合SS2D中各扫描路径的激活图获得的。

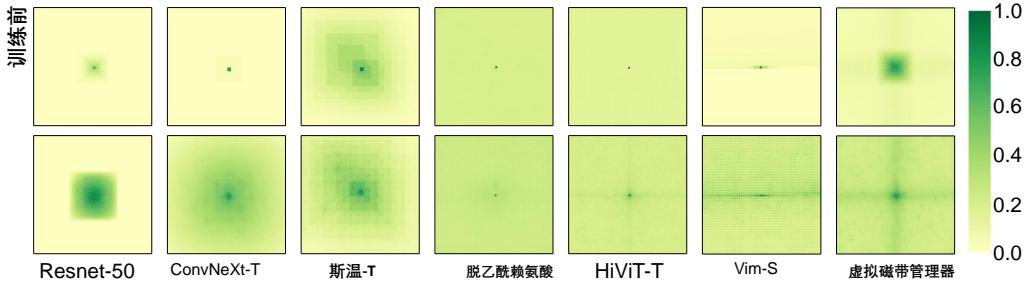


图7: VMamba与其他基准模型有效感受野 (ERF) [40]的比较。强度较高的像素表示与中心像素相关的响应较大。

其中  $\mathbf{h}_a \in \mathbb{R}^{D_k}$  是步骤  $a$  的隐藏状态,  $\Theta$  表示逐元素乘积。特别地,  $\mathbf{V}_i^{(j)}$  仅是一个标量。 $\mathbf{w} := [\mathbf{w}_1; \dots; \mathbf{w}_T] \in \mathbb{R}^{T \times D_k \times D_v}$  中每个元素的表达式即,  $\mathbf{w}_i \in \mathbb{R}^{D_k \times D_v}$  可表示为  $\mathbf{w}_i = \prod_{j=1}^T e^{\Delta_{ra+1+j}}$ , 表示沿扫描路径计算的步骤  $i$  的累积注意力权重。

因此,  $\mathbf{Y}$  的第  $j$  维, 即,  $\mathbf{Y}^{(j)} \in \mathbb{R}^{T \times 1}$ , 可表示为

$$\mathbf{Y}^{(j)} = [\mathbf{Q} \odot \mathbf{w}^{(j)}] \mathbf{h}_a^{(j)} + \left[ (\mathbf{Q} \odot \mathbf{w}^{(j)}) \left( \frac{\mathbf{K}}{\mathbf{w}^{(j)}} \right)^\top \odot \mathbf{M} \right] \mathbf{V}^{(j)}, \quad (5)$$

其中  $\mathbf{M}$  表示尺寸为  $T \times T$  的时间掩码矩阵, 其下三角部分设为 1, 其余部分为 0。更多详细推导过程请参阅附录C。

在等式5中, 涉及  $\mathbf{Q}$ 、 $\mathbf{K}$  和  $\mathbf{V}$  的矩阵乘法过程与自注意力机制极为相似, 尽管包含了  $\mathbf{w}$ 。

**激活图的可视化。**为了直观深入地理解SS2D, 我们进一步可视化了前景物体中特定查询块对应的  $\mathbf{Q}\mathbf{K}^\top$  和  $(\mathbf{Q} \Theta \mathbf{w}) (\mathbf{K}/\mathbf{w})^\top$  中的注意力值 (称为“激活图”)。如图6(b)所示,  $\mathbf{Q}\mathbf{K}^\top$  的激活图展示了SS2D在捕获和保留遍历信息方面的有效性, 前景区域中所有先前扫描的标记都被激活。此外, 引入  $\mathbf{w}$  后, 激活图更聚焦于查询块的邻域 (图6(c)), 这与  $\mathbf{w}$  公式中固有的时间加权效应一致。然而, 选择性扫描机制使VMamba能够沿扫描路径积累历史信息, 从而促进图像块间长期依赖关系的建立。这一点在红色方框标注的子图 (图6(d)) 中尤为明显, 其中左侧最远处的绵羊斑块 (先前步骤中扫描的区域) 仍保持激活状态。更多可视化结果及讨论请参阅附录D。

**有效感受野的可视化。**有效感受野 (ERF) [40, 11]指输入空间中对特定输出单元激活起作用的区域。我们对不同视觉骨干网络在训练前后的中心像素ERF进行了对比分析。图7所示结果表明, 在所考察的模型中, 仅有DeiT、HiViT、Vim和VMamba展现出全局ERF, 而其他模型尽管理论上具备全局覆盖潜力, 却仅呈现局部ERF。此外, VMamba的线性时间复杂度相比DeiT和HiViT (其计算成本与输入补丁数量呈二次方关系) 提升了计算效率。虽然VMamba和Vim均基于Mamba架构, 但VMamba的ERF比Vim更均匀且具有二维感知能力, 这或许直观解释了其性能优势。

**选择性扫描模式诊断研究。**我们将提出的扫描模式 (即交叉扫描) 与三种基准模式进行对比: 单向扫描 (Unidi-Scan)、双向扫描 (Bidi-Scan) 以及级联扫描 (Cascade-Scan, 即按行和列顺序依次扫描数据)。通过调整特征维度以保持相似的架构参数。

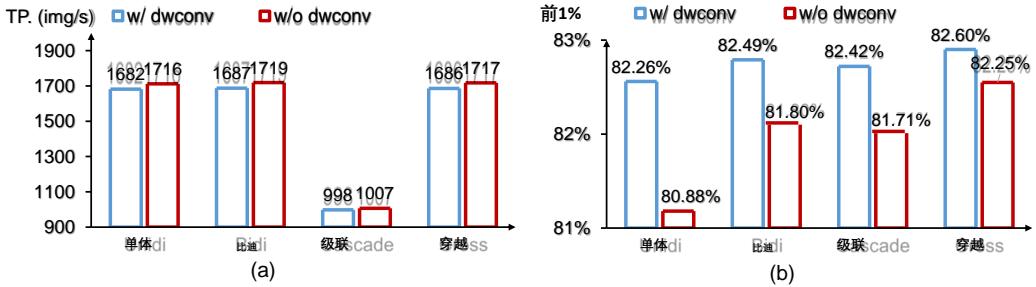


图8：不同扫描模式的性能比较。所提出的交叉扫描（Cross-Scan）在保持相同参数数量和浮点运算次数（FLOPs）的情况下，实现了更优的运算速度。

为确保公平比较，需明确浮点运算次数（FLOPs）。如图8所示，交叉扫描模式在计算效率和分类准确率方面均优于其他扫描模式，充分展现了其在实现二维选择性扫描（2D-Selective-Scan）方面的优势。移除DWConv层（该层已被证实有助于模型学习二维空间信息）后，这一优势进一步凸显。这表明交叉扫描模式通过采用四向扫描技术，天然具备捕捉二维上下文信息的强项。

## 6 结论

本文提出了一种基于状态空间模型（SSMs）构建的高效视觉骨干网络VMamba。该模型通过创新的SS2D模块，将自然语言处理任务中选择性SSMs的优势融入视觉数据处理，成功弥合了顺序一维扫描与非顺序二维遍历之间的技术鸿沟。通过系列架构优化与实现改进，我们显著提升了VMamba的推理速度。大量实验证明了VMamba系列模型的有效性，其线性时间复杂度使其在处理大分辨率输入的下游任务中具有显著优势。

**局限性。**尽管VMamba展现出令人鼓舞的实验结果，但本研究仍有改进空间。先前研究已验证了大规模数据集（例如ImageNet-21K）上无监督预训练的有效性。然而，现有预训练方法与VMamba这类基于SSM的架构的兼容性，以及针对此类模型专门设计的预训练技术的识别，仍处于未探索阶段。研究这些方面可能为未来架构设计研究提供重要方向。此外，有限的计算资源限制了我们对VMamba架构进行大规模探索及开展精细超参数搜索以进一步提升实验性能。虽然VMamba的核心组件SS2D对输入数据的布局或模态不做特定假设，使其具备跨任务泛化能力，但VMamba在更广泛任务中集成的潜力仍有待探索。弥合SS2D与这些任务之间的差距，并提出一种更通用的视觉任务扫描模式，代表了一个具有前景的研究方向。

## 7 致谢

本研究获得以下资助：国家自然科学基金（NSFC）项目编号62225208和62406304、中国科学院青年科学家基础研究项目（项目编号YSBR-117）、中国博士后科学基金（项目编号2023M743442）以及CPSF博士后奖学金计划（项目编号GZB20240730）。