Are Transformers the Final Frontier?

2024년도 하계 세미나

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

Presented By
Haeuk Lee

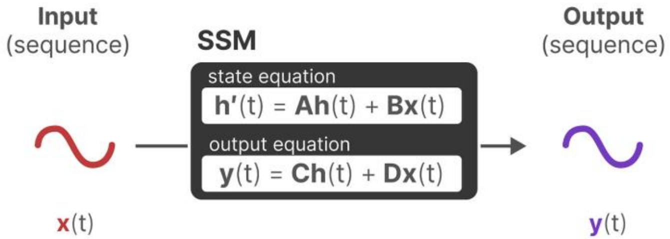
1) Tri, Dao; Albert, Gu. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality. *The International Conference on Machine Learning (ICML)*, 2024.
2) Yue, Liu; Yunjie, Tian; Yuzhong, Zhao; Hongtian, Yu; Lingxi, Xie; Yaowei, Wang; Qixiang, Ye; Yunfan, Liu. VMamba: Visual State Space Model. *arXiv preprint* arXiv:2401.10166, 2024.

Outline

- Background
- Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality[1)]
  - ICML 2024
- VMamba: Visual State Space Model[2)]
  - arXiv

Background

- **State Space Models (SSMs)**
  - SSMs are used to describe state representations and predict future states based on input
  - At time t, SSMs:
    - Map input sequence $x(t)$: e.g., moved left and down in a maze
    - Generate latent state representation $h(t)$: e.g., distance to exit, x/y coordinates
    - Derive predicted output sequence $y(t)$: e.g., move left again to reach the exit sooner
  - SSMs handle continuous sequences as input and predict continuous output sequences
  - Assumption: Dynamic systems can be predicted using state $h(t)$ through two core equations
  - By solving these equations, SSMs aim to uncover statistical principles to predict the system's state based on observed data
  - The goal of SSMs is to find the state representation $h(t)$ that allows transitioning from input to output sequences effectively



Two core equations of the State Space Model

- Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality[1)]
  - ICML 2024

Contribution

- Theoretical Framework
  - Established the Structured State Space Duality (SSD) framework
  - Connects structured state-space models (SSMs) with attention mechanisms
- Architecture Design
  - Introduced the Mamba-2 architecture
  - Integrates SSD into the core design, making the model faster and more efficient
- Algorithm Development
  - Developed new algorithms for SSMs
  - These algorithms improve the speed of Mamba-2 by 2-8 times compared to its predecessor

Why Mamba-2?

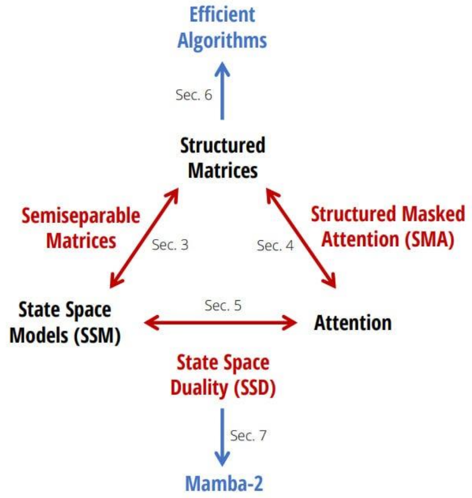
- Challenges in MAMBA:
  - Challenge 1: Complexity in Understanding
    - SSM vs. Attention Mechanism:
      - Mamba effectively uses SSMs for sequence modeling
      - However, certain aspects remain where the attention mechanism demonstrates superior performance
    - Key Questions:
      - What are the theoretical links between SSMs and attention?
      - Can these two approaches be integrated?

Why Mamba-2?

- Challenges in MAMBA:
  - Challenge 2: Efficiency
    - Computational Efficiency:
      - Hardware-aware algorithms in Mamba are less efficient than attention mechanisms
    - Hardware Optimization:
      - Modern GPUs and TPUs are optimized for matrix operations
      - There is a need to improve computational efficiency during training
    - Key Question:
      - Can Mamba be adapted to effectively use matrix multiplication?

SSD Framework

- State Space Duality (SSD) Framework
  - This framework establishes a connection between structured State-Space Models (SSMs) and attention mechanisms, particularly those used in Transformers
- Core Concept: Structured Matrices
  - The SSD framework leverages the concept of structured matrices
  - Structured matrices are a type of matrix with subquadratic parameters and efficient multiplication algorithms
  - These matrices serve as the bridge linking SSMs and attention mechanisms



SSD Framework

- Quadratic Mode
  - Matrix Multiplication
    - SSM is represented as a $T \times T$ matrix
    - Like the attention mechanism's Query, Key, and Value matrices
    - The SSM operation can be represented as a matrix multiplication
      - $y = SSM(A, B, C)(x) = Mx$
    - Here, $M_{ji} = C_j^T A_{j:i} B_i$, where $A_{j:i}$ is a product of state matrices from index i to j
    - The matrix M can be decomposed and rewritten in a form resembling attention
      - $M = L \circ (CB^T)$
    - L is a structured matrix derived from A, specifically a 1-Semiseparable (1-SS) matrix
    - Like the attention mechanism's Query, Key, and Value matrices
  - Analogies
    - SSM's state matrix $A \approx$ attention's pre-softmax score matrix
    - SSM's matrices B and $C \approx$ attention's Value matrix
    - Output Y is computed similarly to attention

SSD Framework

- Linear Mode
  - Defines the state-space model (SSM) as a mapping from $x \in R^T$ to $y \in R^T$
    - The state update is given by $h_t = A_t h_{t-1} + B_t x$, and the output is $y_t = C_t^T h_t$
  - Selective SSM allows the parameters A, B, and C to vary over time, enabling dynamic adaptation
  - Structured SSM enforces the matrix A to be diagonal for more efficient computation
    - This structure is also applied in models like S6
  - SSD Refinement:
    - Further simplifies A to a scalar times identity matrix, where all diagonal elements are equal
    - This allows A to be represented by its shape and a single scalar value, enhancing computational efficiency

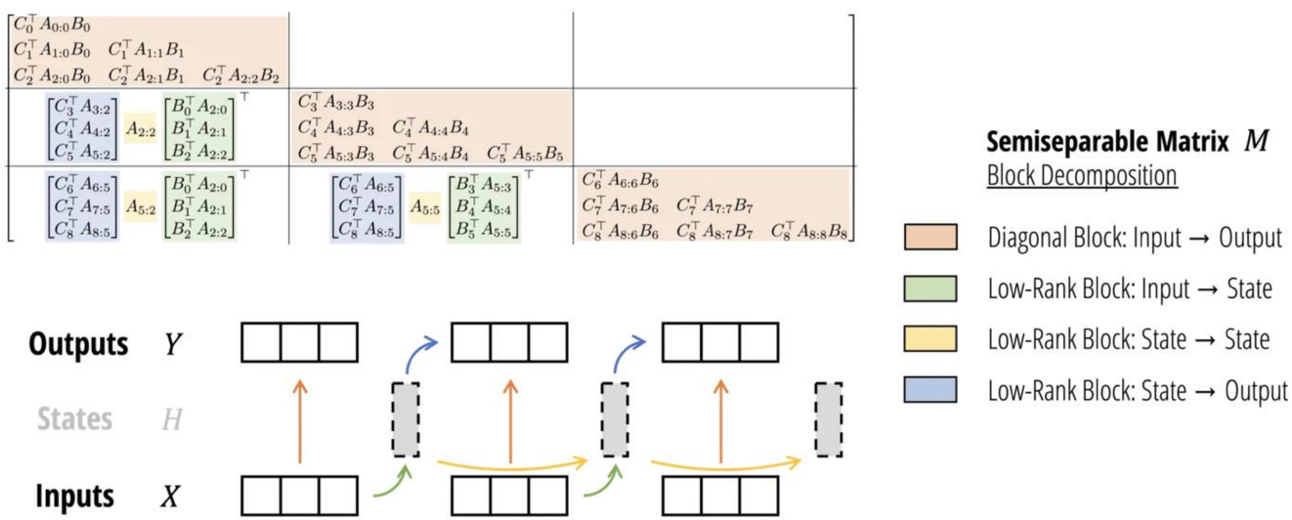
SSD Framework

- SSD vs. State Space Models: Key Differences from Mamba 1 to Mamba 2
  - Change in Matrix Structure
    - Transition from diagonal A to scalar-times-identity structure A
    - This structure shares recurrent dynamics across all elements of the state space
  - Channels and Heads
    - Mamba 1: P=1 (single channel)
    - Mamba 2: P>1, enabling shared dynamics across multiple channels via P heads
  - From Multiple Recurrences to Single Shared Recurrence
    - Mamba 1: Individual scalar recurrence for each of $P \times N$ elements
    - Mamba 2: A single shared recurrence, improving computational efficiency
  - Interpretation in Dual (Quadratic) Attention Form
    - Enables matrix operations, making the model computationally efficient
    - Concern about potential performance issues is mitigated by the model's selectivity—only relevant information is propagated, ensuring overall effectiveness

1) Tri, Dao; Albert, Gu. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality. *The International Conference on Machine Learning (ICML)*, 2024.

SSD Framework

- Efficiency: the SSD Mode
  - Which Mode to Use?
    - During inference, using SSM directly is ideal since there is no trade-off
    - However, during training, it's preferable to consider both computation time and hardware efficiency. Thus, matrix multiplication is more desirable for faster processing



SSD Algorithm

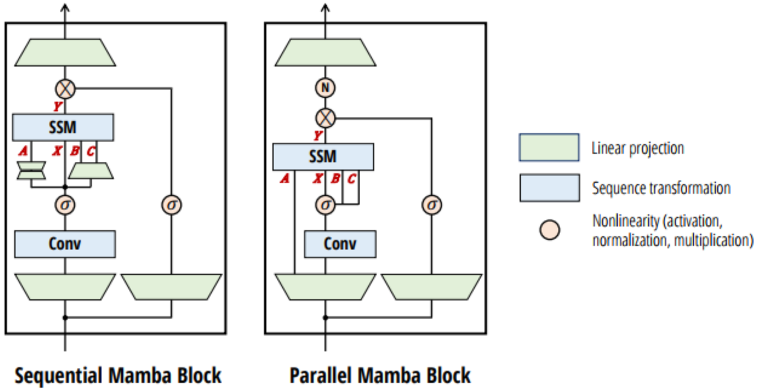
SSD Framework

- Efficiency: the SSD Mode
  - Block Decomposition
    - A specific structured matrix is decomposed to define the SSD "token mixing" sequence transformation
    - This decomposition is essential for the sequence transformation in SSD
  - Chunkwise Algorithm
    - The sequence is divided into segments, with quadratic attention computed on each segment
    - The results are then adjusted by passing SSM states between segments, ensuring that the final outcome reflects the state transitions across the entire sequence

	Attention	SSM	**SSD**
State size	$\mathtt{T}$	$\mathbf{N}$	$\mathbf{N}$
Training FLOPs	$\mathtt{T}^2\mathtt{N}$	$\mathbf{TN}^2$	$\mathbf{TN}^2$
Inference FLOPs	$\mathtt{TN}$	$\mathbf{N}^2$	$\mathbf{N}^2$
(Naive) memory	$\mathtt{T}^2$	$\mathtt{TN}^2$	$\mathbf{TN}$
Matrix multiplication	✓		✓

Mamba-2 Architecture

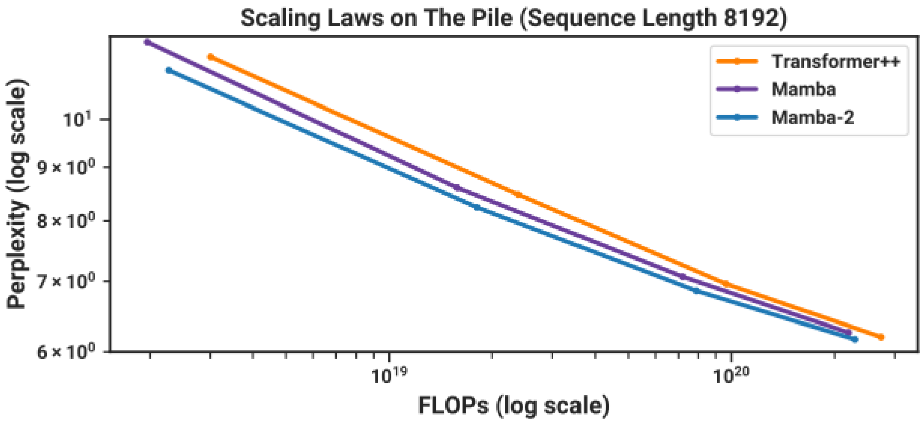
- Simplification of Mamba Block
  - Sequential linear projections removed
  - SSM parameters A, B, C generated at the start of the block
- Normalization Layer
  - Additional normalization layer added for stability
    - Inspired by NormFormer (Shleifer et al., 2021)
- Projections B and C
  - Single head shared across X heads
  - Analogous to Multi-Value Attention (MVA)



Mamba-2 Architecture

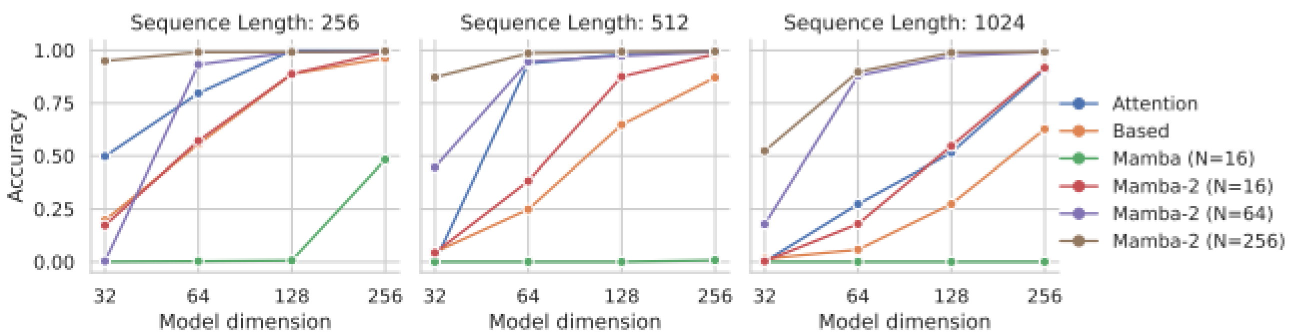
Experiments

- Scaling Laws on the Pile dataset
  - Model Size
    - The experiment involves models with sizes ranging from approximately 125 million (≈125M) to 1.3 billion (≈1.3B) parameters
    - The models were trained on the Pile dataset
  - Mamba-2 Performance
    - Mamba-2 matches or exceeds the performance of the original Mamba model and a strong "Transformer++" recipe
    - Compared to the baseline Transformer, Mamba-2 is Pareto dominant in terms of performance (perplexity), theoretical FLOPs, and actual wall-clock time



Experiments

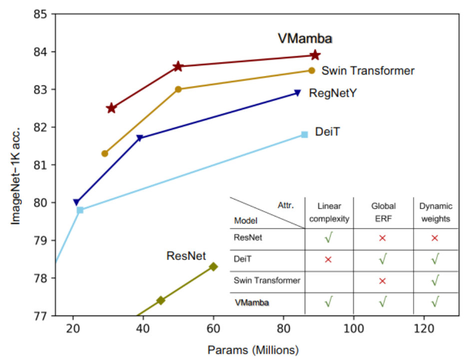
- Synthetic Language Modeling : MQAR
  - Associative Recall Challenge
    - Associative recall tasks require SSMs to store and retrieve all relevant information within their recurrent state
  - SSD Layer and Architecture Enhancement
    - The inclusion of the SSD layer and an improved architecture in Mamba-2 supports much larger state sizes
  - Mamba-2 Performance
    - Mamba-2 significantly outperforms both Mamba-1 and vanilla attention in these tasks



- VMamba: Visual State Space Model[1)]
  - arXiv

Contribution

- Built on State Space Models (SSMs) for efficient visual representation learning
- Cross-Scan Module (CSM)
  - Introduced the Cross-Scan Module (CSM) to address the direction-sensitive problem in vision data
  - Enables 1D selective scanning in 2D image space
  - Achieves global receptive fields without increasing computational complexity
- Efficient Computational Design
  - VMamba reduces the quadratic complexity of attention computation to linear complexity
  - Utilizes a four-way scanning strategy to ensure comprehensive information integration across the image

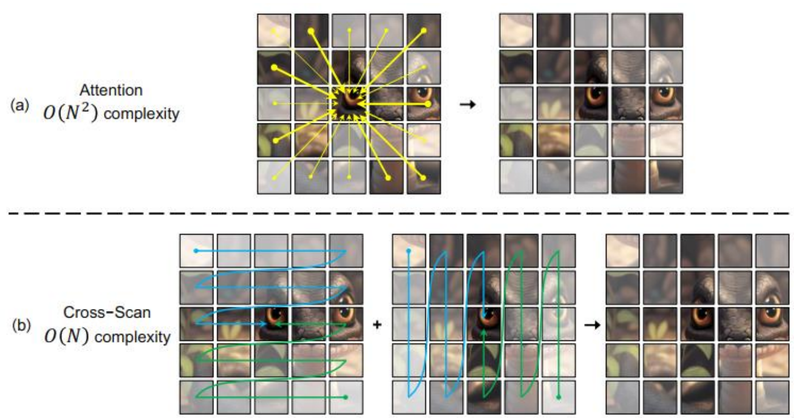


Model \ Attr.	Linear complexity	Global ERF	Dynamic weights
ResNet	√	×	×
DeiT	×	√	√
Swin Transformer		×	√
VMamba	√	√	√

Performance comparison on ImageNet-1K

2D-Selective-Scan

- 2D-Selective-Scan(SS2D) is proposed to adapt the S6 model to vision data
  - It incorporates global receptive fields, dynamic weights, and linear computational complexity
  - Addresses challenges posed by the non-sequential and spatial nature of vision data
  - SS2D involves three steps:
    - Cross-scan, selective scanning with S6 blocks, and cross-merge
- Cross-Scan Module (CSM)
  - CSM handles the unfolding of image patches into sequences along four distinct paths
  - CSM ensures each pixel integrates information from all other pixels in different directions



Comparison of information flow: Attention vs. Cross-Scan Module (CSM)

1) Yue, Liu; Yunjie, Tian; Yuzhong, Zhao; Hongtian, Yu; Lingxi, Xie; Yaowei, Wang; Qixiang, Ye; Yunfan, Liu. VMamba: Visual State Space Model. *arXiv preprint* arXiv:2401.10166, 2024.

2D-Selective-Scan

- Selective Scanning Process
  - Image patches are processed by separate S6 blocks in parallel
  - Outputs from the scanning process are merged to reconstruct the 2D feature map (i.e., cross-merge)
  - This process facilitates the establishment of global receptive fields
- Advantages of SS2D
  - Maintains the advantages of the S6 model in vision tasks
  - Ensures effective context-aware data modeling while preserving linear computational efficiency
  - Overcomes the limitations of existing methods in capturing long-range dependencies in 2D vision data

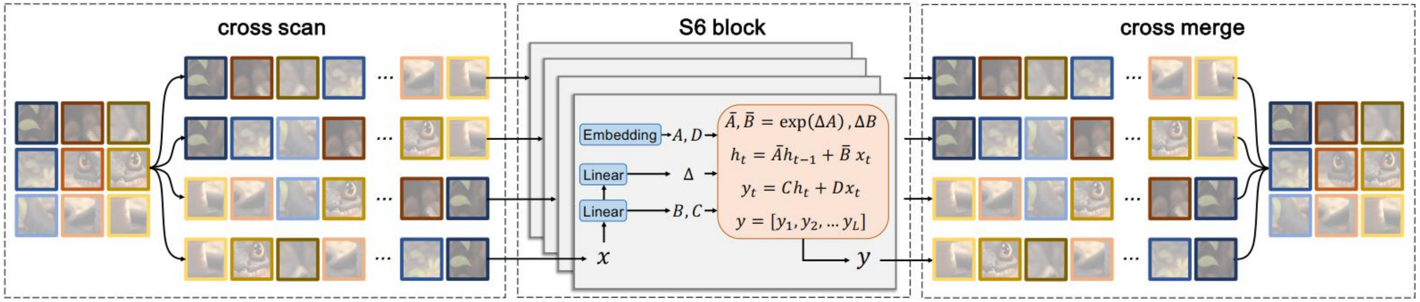


Illustration of the 2D-Selective-Scan (SS2D) operation

1) Yue, Liu; Yunjie, Tian; Yuzhong, Zhao; Hongtian, Yu; Lingxi, Xie; Yaowei, Wang; Qixiang, Ye; Yunfan, Liu. VMamba: Visual State Space Model. *arXiv preprint* arXiv:2401.10166, 2024.

The VMamba Model Family

- Overview of VMamba Architecture
  - VMamba is developed in three scales:
    - VMamba-Tiny (T), VMamba-Small (S), and VMamba-Base (B)
  - The architecture is designed to process input images through a series of stages, each producing hierarchical representations with decreasing resolutions
  - The model begins with a stem module that partitions the input image into patches, followed by VSS blocks at each stage

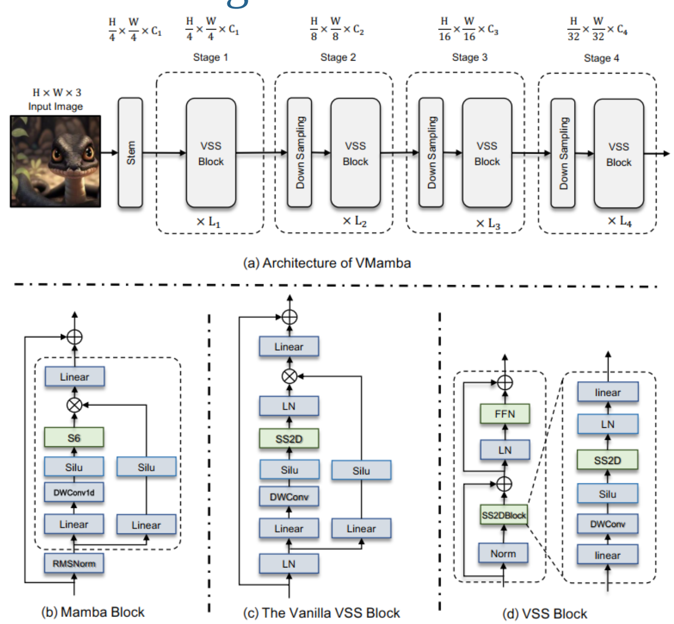


Illustration of network architectures

서강대학교 SOGANG UNIVERSITY

VDS LAB

The VMamba Model Family

- Vanilla VSS Block
  - The Vanilla VSS Block serves as the core module in building VMamba
  - It is a residual network with a skip connection, consisting of two branches:
    - One branch uses a 3×3 depth-wise convolution layer for feature extraction
    - The other branch includes a linear mapping followed by an activation layer, responsible for computing the gating signal
  - The SS2D module is integrated to adapt selective scanning to 2D vision data, replacing the S6 module found in the original Mamba block
- Differences from Vision Transformers
  - Unlike Vision Transformer (ViT) blocks, the Vanilla VSS Block does not use position embedding bias
  - The architecture of VMamba is shallower than typical ViT blocks, allowing for more blocks to be stacked within a similar depth budget

1) Yue, Liu; Yunjie, Tian; Yuzhong, Zhao; Hongtian, Yu; Lingxi, Xie; Yaowei, Wang; Qixiang, Ye; Yunfan, Liu. VMamba: Visual State Space Model. *arXiv preprint* arXiv:2401.10166, 2024.

Experiments

- Image Classification on ImageNet-1K
  - Settings
    - VMamba models (Tiny, Small, Base) were trained from scratch for 300 epochs
    - Used AdamW optimizer with a batch size of 1024
  - Results
    - VMamba-T achieved 82.5% top-1 accuracy, outperforming other models like RegNetY-4G, DeiT-S, and Swin-T
    - VMamba-S and VMamba-B also demonstrated superior performance compared to their counterparts

Method	Image size	#Param.	FLOPs	Throughput	Train Throughput	ImageNet top-1 acc.
RegNetY-4G [41]	224^2	21M	4.0G	–	–	80.0
RegNetY-8G [41]	224^2	39M	8.0G	–	–	81.7
RegNetY-16G [41]	224^2	84M	16.0G	–	–	82.9
EffNet-B3 [47]	300^2	12M	1.8G	–	–	81.6
EffNet-B4 [47]	380^2	19M	4.2G	–	–	82.9
EffNet-B5 [47]	456^2	30M	9.9G	–	–	83.6
EffNet-B6 [47]	528^2	43M	19.0G	–	–	84.0
ViT-B/16 [12]	384^2	86M	55.4G	–	–	77.9
ViT-L/16 [12]	384^2	307M	190.7G	–	–	76.5
DeiT-S [50]	224^2	22M	4.6G	1759	2397	79.8
DeiT-B [50]	224^2	86M	17.5G	500	1024	81.8
DeiT-B [50]	384^2	86M	55.4G	498	344	83.1
ConvNeXt-T [33]	224^2	29M	4.5G	1189	701	82.1
ConvNeXt-S [33]	224^2	50M	8.7G	682	444	83.1
ConvNeXt-B [33]	224^2	89M	15.4G	435	334	83.8
HiViT-T [64]	224^2	19M	4.6G	1391	1300	82.1
HiViT-S [64]	224^2	38M	9.1G	711	697	83.5
HiViT-B [64]	224^2	66M	15.9G	456	541	83.8
Swin-T [32]	224^2	28M	4.6G	1247	985	81.3
Swin-S [32]	224^2	50M	8.7G	719	640	83.0
Swin-B [32]	224^2	88M	15.4G	457	494	83.5
S4ND-ConvNeXt-T [40]	224^2	30M	-	684	331	82.2
S4ND-ViT-B [40]	224^2	89M	-	404	340	80.4
ViM-S [68]	224^2	26M	-	811	$232^\dagger$	80.5
VMamba-T	224^2	31M	4.9G	1335	464	82.5
VMamba-S	224^2	50M	8.7G	874	313	83.6
VMamba-B	224^2	89M	15.4G	645	246	83.9

Performance comparison on ImageNet-1K

Experiments

- Object Detection on COCO
  - Settings
    - Used the Mask-RCNN detector with fine-tuning for 12 and 36 epochs
    - The VMamba models were evaluated on object detection (APb) and instance segmentation (APm)
  - Results
    - VMamba-T/S/B models showed significant improvements over Swin and ConvNeXt models in both object detection and instance segmentation
    - With the 12-epoch schedule, VMamba-T achieved 47.4% APb, outperforming Swin-T by 4.7% and ConvNeXt-T by 3.2%

Backbone	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}	#param.	FLOPs
Mask R-CNN 1× schedule								
ResNet-50	38.2	58.8	41.4	34.7	55.7	37.2	44M	260G
Swin-T	42.7	65.2	46.8	39.3	62.2	42.2	48M	267G
ConvNeXt-T	44.2	66.6	48.3	40.1	63.3	42.8	48M	262G
PVTv2-B2	45.3	67.1	49.6	41.2	64.2	44.4	45M	309G
ViT-Adapter-S	44.7	65.8	48.3	39.9	62.5	42.8	48M	403G
VMamba-T	47.4	69.5	52.0	42.7	66.3	46.0	50M	270G
ResNet-101	38.2	58.8	41.4	34.7	55.7	37.2	63M	336G
Swin-S	44.8	66.6	48.9	40.9	63.2	44.2	69M	354G
ConvNeXt-S	45.4	67.9	50.0	41.8	65.2	45.1	70M	348G
PVTv2-B3	47.0	68.1	51.7	42.5	65.7	45.7	65M	397G
VMamba-S	48.7	70.0	53.4	43.7	67.3	47.0	64M	357G
Swin-B	46.9	–	–	42.3	–	–	107M	496G
ConvNeXt-B	47.0	69.4	51.7	42.7	66.3	46.0	108M	486G
PVTv2-B5	47.4	68.6	51.9	42.5	65.7	46.0	102M	557G
ViT-Adapter-B	47.0	68.2	51.4	41.8	65.1	44.9	102M	557G
VMamba-B	49.2	70.9	53.9	43.9	67.7	47.6	108M	485G
Mask R-CNN 3× MS schedule								
Swin-T	46.0	68.1	50.3	41.6	65.1	44.9	48M	267G
ConvNeXt-T	46.2	67.9	50.8	41.7	65.0	44.9	48M	262G
PVTv2-B2	47.8	69.7	52.6	43.1	66.8	46.7	45M	309G
ViT-Adapter-S	48.2	69.7	52.5	42.8	66.4	45.9	48M	403G
VMamba-T	48.9	70.6	53.6	43.7	67.7	46.8	50M	270G
Swin-S	48.2	69.8	52.8	43.2	67.0	46.1	69M	354G
ConvNeXt-S	47.9	70.0	52.7	42.9	66.9	46.2	70M	348G
PVTv2-B3	48.4	69.8	53.3	43.2	66.9	46.7	65M	397G
VMamba-S	49.9	70.9	54.7	44.2	68.2	47.7	70M	384G

Results of object detection and instance segmentation on COCO dataset

Experiments

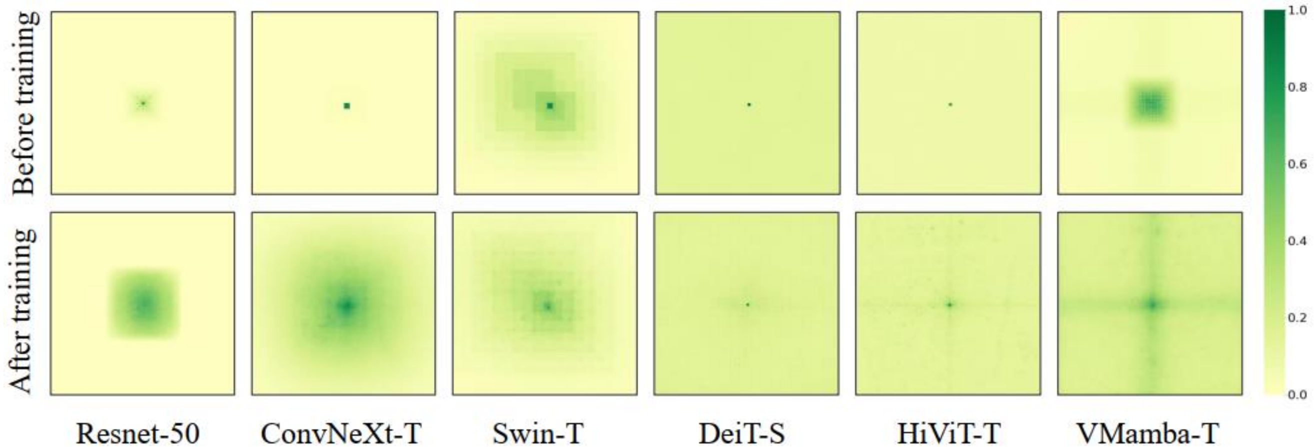
- Semantic Segmentation on ADE20K
  - Settings
    - UperHead was constructed on top of VMamba models, fine-tuned for 160k iterations
    - Experiments included both single-scale (SS) and multi-scale (MS) testing
  - Results
    - VMamba-T achieved 48.3% mIoU (SS) and 48.6% mIoU (MS), outperforming ResNet, DeiT, Swin, and ConvNeXt
    - VMamba-S and VMamba-B similarly outperformed their respective benchmarks

method	crop size	mIoU (SS)	mIoU (MS)	#param.	FLOPs
ResNet-50	512^2	42.1	42.8	67M	953G
DeiT-S + MLN	512^2	43.8	45.1	58M	1217G
Swin-T	512^2	44.4	45.8	60M	945G
ConvNeXt-T	512^2	46.0	46.7	60M	939G
VMamba-T	512^2	48.3	48.6	62M	948G
ResNet-101	512^2	42.9	44.0	85M	1030G
DeiT-B + MLN	512^2	45.5	47.2	144M	2007G
Swin-S	512^2	47.6	49.5	81M	1039G
ConvNeXt-S	512^2	48.7	49.6	82M	1027G
VMamba-S	512^2	50.6	51.2	82M	1039G
Swin-B	512^2	48.1	49.7	121M	1188G
ConvNeXt-B	512^2	49.1	49.9	122M	1170G
VMamba-B	512^2	51.0	51.6	122M	1170G

Results of semantic segmentation on ADE20K using UperNet

Experiments

- Analytical Experiments
  - Effective Receptive Field (ERF)
    - VMamba demonstrated a global ERF, unlike CNN-based models, which showed local ERFs
    - VMamba's ERF coverage expanded from local to global during training, contributing to better image perception



Visualization of the Effective Receptive Field (ERF)

1) Yue, Liu; Yunjie, Tian; Yuzhong, Zhao; Hongtian, Yu; Lingxi, Xie; Yaowei, Wang; Qixiang, Ye; Yunfan, Liu. VMamba: Visual State Space Model. *arXiv preprint* arXiv:2401.10166, 2024.

Experiments

- Analytical Experiments
  - Computational Efficiency with Increasing Resolutions
    - VMamba maintained stable performance across different input resolutions, with linear growth in computational complexity
    - Showed better scalability compared to models like Swin and ResNet with increasing input sizes

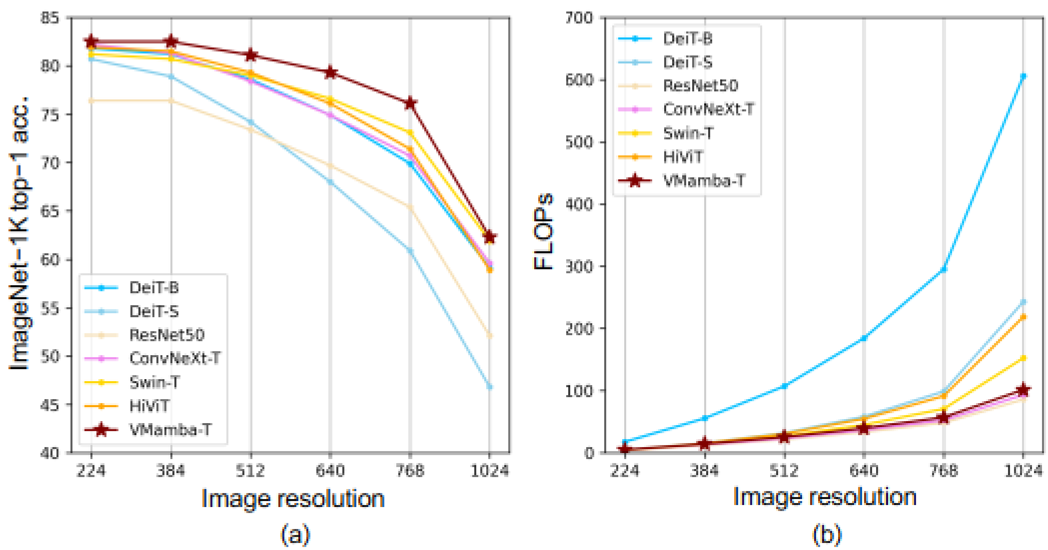


Illustration of the change in (a) classification accuracy and (b) FLOPs with progressively larger test image resolutions

Experiments

- Diagnostic Study on Selective Scan Patterns
  - Compared different scanning methods (Unidi-Scan, Bidi-Scan, Cascade-Scan) with the Cross-Scan Module (CSM)
  - CSM demonstrated superior data modeling capacity and stability, with higher classification accuracy and better computational efficiency

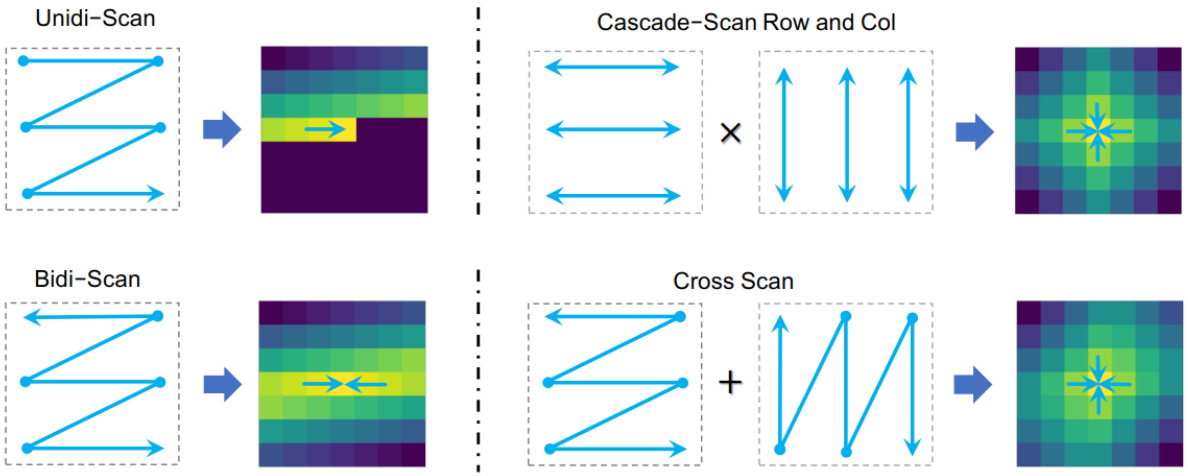


Illustration of different scanning methods for selective scan

Method	#Param.	FLOPs	Throughput	Train Throughput.	ImageNet top-1 acc.
Unidi-Scan	30.70	4.86	1342	464	82.2
Bidi-Scan	30.70	4.86	1344	465	74.8†
Cascade-Scan	30.70	4.86	817	253	–
CSM	30.70	4.86	1343	464	82.5

Performance comparison of different scanning approaches