Multi-Scale VMamba: Hierarchy in Hierarchy Visual State Space Model

Yuheng Shi
City University of Hong Kong
yuhengshi99@gmail.com

Minjing Dong
City University of Hong Kong
minjdong@cityu.edu.hk

Chang Xu
University of Sydney
c.xu@sydney.edu.au

Abstract

Despite the significant achievements of Vision Transformers (ViTs) in various vision tasks, they are constrained by the quadratic complexity. Recently, State Space Models (SSMs) have garnered widespread attention due to their global receptive field and linear complexity with respect to the input length, demonstrating substantial potential across fields including natural language processing and computer vision. To improve the performance of SSMs in vision tasks, a multi-scan strategy is widely adopted, which leads to significant redundancy of SSMs. For a better trade-off between efficiency and performance, we analyze the underlying reasons behind the success of the multi-scan strategy, where long-range dependency plays an important role. Based on the analysis, we introduce Multi-Scale Vision Mamba (MSVMamba) to preserve the superiority of SSMs in vision tasks with limited parameters. It employs a multi-scale 2D scanning technique on both original and downsampled feature maps, which not only benefits long-range dependency learning but also reduces computational costs. Additionally, we integrate a Convolutional Feed-Forward Network (ConvFFN) to address the lack of channel mixing. Our experiments demonstrate that MSVMamba is highly competitive, with the MSVMamba-Tiny model achieving 83.0% top-1 accuracy on ImageNet, 46.9% box mAP, and 42.5% instance mAP with the Mask R-CNN framework, 1x training schedule on COCO, and 47.9% mIoU with single-scale testing on ADE20K. Code is available at `https://github.com/YuHengsss/MSVMamba`.

1 Introduction

In the domain of computer vision, the extraction of features plays a pivotal role in the performance of various tasks, ranging from image classification to more complex applications like detection and segmentation. Traditionally, Convolutional Neural Networks (CNNs) [25, 40, 18, 21, 33] have been the backbone of feature extraction methodologies, prized for their linear scaling complexity and proficiency in capturing local patterns. However, CNNs often fall short in encapsulating global context, a limitation that becomes increasingly apparent in tasks requiring a comprehensive understanding of the entire visual field. In contrast, Vision Transformers (ViTs) [6, 32, 50, 43] have emerged as a compelling alternative, boasting an inherent global receptive field that allows for the direct capture of long-range dependencies within an image. Despite their advantages, ViTs are hampered by their quadratic scaling complexity concerning the input size, which significantly constrains their applicability to downstream tasks such as object detection and segmentation, where efficiency is paramount. Recently, State Space Model (SSM)-based approaches [11, 41, 8] have garnered attention for their ability to combine the best of both worlds: a global receptive field and linear scaling complexity. Notably, Mamba [9] introduces a hardware-aware and input-dependent algorithm that significantly enhances the performance and efficiency of SSMs. Inspired by Mamba's success, a burgeoning body of work has sought to leverage its advantages for vision tasks, pioneering efforts such as ViM [60] and VMamba [30].

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

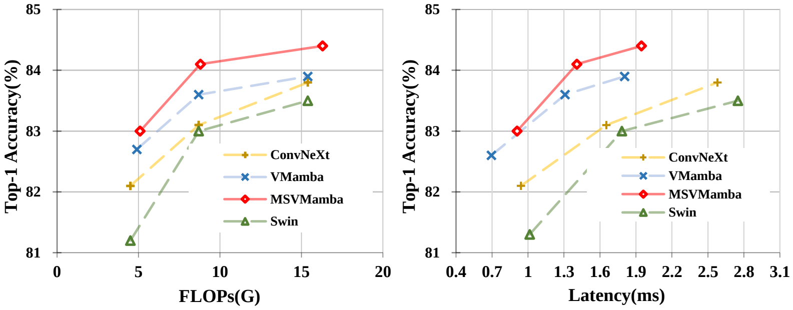


Figure 1: FLOPs and latency comparison on ImageNet. The latency was tested on a RTX 4090 GPU with a batch size of 128 using FP32 precision at an image resolution of 224.

The S6 block, developed by Mamba [9], was originally designed for NLP tasks. To adapt S6 for vision tasks, images are first divided into patches and then flattened into a patch sequence along the scanning path. To accommodate the non-causal nature of image data, the multi-scan strategy is widely adopted for vision tasks, such as ViM [60] which enhances the sequence by summing it in both forward and reverse directions, and VMamba [30] which integrates horizontal and vertical scans. However, unlike NLP models, which can have up to billions of parameters, current vision backbones always take computational costs into consideration, *i.e.*, the trade-off between accuracy and efficiency. This constraint on model size inherently limits the long-range modeling capabilities of SSMs in vision tasks. Taking ViM-Tiny [60] as an example, placing the cls token in the middle of the sequence yields markedly better results than positioning it at the ends. This suggests that central placement compensates for the model's limited ability to integrate distant information, highlighting the difficulties of handling long-range dependencies in parameter-constrained vision models. We refer to this as the long-range forgetting problem. In this work, we analyze how the multi-scan strategy in [60, 30] helps to alleviate this problem. Compared to the single-scan strategy, the multi-scan one allows long-range decay to manifest in various directions within 2D images. However, the increased scanning routes bring multiples of computations, significantly increasing redundancy and limiting efficiency. Thus, we aim to pursue a better trade-off between the performance and efficiency of Mamba in vision tasks.

The most direct and effective method to address the long-range forgetting problem is to shorten the sequence length, which can be achieved by downsampling the feature map. However, placing all scanning routes on a downsampled feature map could result in the loss of fine-grained features and the downstream task performance. Through the visualization of different scans, we show that the decay rates could vary for different scanning routes, which motivates us to develop a hierarchical design of multi-scan. In this work, we propose a Multi-Scale 2D (MS2D) scanning strategy to alleviate the long-range forgetting problem with limited computational costs. Specifically, we divide the scanning directions of SS2D [30] into two groups: one retains the original resolution and is processed by the S6 block, while the others are downsampled, processed by the S6 block, and then upsampled, which not only shortens the sequence length for long-range dependencies learning but also alleviates redundancy. Building on the VMamba with its hierarchical architecture, we incorporate another hierarchical design within the block, creating a hierarchy within a hierarchy. Furthermore, we introduce a Convolutional Feed-Forward Network (ConvFFN) within each block to bolster the model's capability for channel-wise information exchange and local feature capture.

We conduct extensive experiments to validate the effectiveness of MSVmamba across a spectrum of tasks, including image classification, object detection, and semantic segmentation. Detailed comparisons on the ImageNet-1K [3] dataset are illustrated in Fig. 1. MSVMamba achieves a notable improvement over VMamba across different model sizes.

2 Related work

2.1 Generic Vision Backbone

The evolution of generic vision backbones has significantly shaped the landscape of computer vision, transitioning from CNNs [25, 40, 18, 54, 21, 19, 42, 33] to ViTs [6, 32, 31, 50, 51, 5, 43, 57]. CNNs have been the cornerstone of vision-based models, dominating vision tasks in the early era of deep learning. The classic CNNs such as AlexNet [25], VGG [40], and ResNet [18], have paved the way for numerous subsequent innovations [54, 21, 19, 42, 20, 33, 4, 49]. These designs have significantly improved performance on a wide range of vision tasks by enhancing the network's ability to capture complex patterns and features from visual data. The Vision Transformer [6], drawing inspiration from the success of transformers [47] in natural language processing, has emerged as a formidable contender to conventional CNNs for vision-related tasks. ViT reimagines image processing by segmenting an image into patches and employing self-attention mechanisms to process these segments. This innovative approach enables the model to discern global dependencies across the entire image, a significant leap forward in understanding complex visual data. However, the ViT architecture demands considerable computational resources and extensive datasets for effective training. Moreover, its performance is intricately tied to the input sequence length, exhibiting a quadratic complexity that can escalate processing costs. In response, subsequent research has focused on developing more efficient training strategies [43, 45, 24], hierarchical network structures [32, 31, 5, 50, 51, 15], refined spatial attention mechanisms [57, 59, 52, 46, 16, 5] and convolution-based design [2, 29, 14, 36] to address these issues.

2.2 State Space Models

State Space Models (SSMs) [10, 12, 11, 41, 8] have garnered increasing attention from researchers due to their computational complexity, which grows linearly with the length of the input sequence, and their inherent global awareness properties. To reduce the computational resource consumption of SSMs, S4 [11] introduced a diagonal structure and combined it with a diagonal plus low-rank approach to construct structured SSMs. Subsequently, S5 [41] and H3 [8] further enhanced the efficiency of SSM-based models by introducing parallel scanning techniques and improving hardware utilization. Mamba [9] then introduced the S6 block, incorporating data-dependent parameters to amend the Linear Time Invariant(LTI) characteristics of previous SSM models, demonstrating superior performance over transformers on large-scale datasets. In the realm of vision tasks, S4ND [35] pioneered the application of SSM models in vision tasks by treating visual data as 1D, 2D, and 3D signals. U-Mamba [34] combined CNNs with SSMs for medical image segmentation. ViM [60] and VMamba [30] integrated the S6 block into vision backbone design, employing multiple scanning directions to accommodate the non-casual nature of image data, achieving competitive results against ViTs and CNNs. Motivated by the success of ViM and VMamba, a plethora of Mamba-based works [37, 23, 7, 55, 1, 26, 56, 39] have emerged across various vision tasks, including vision backbone design [37, 23, 55], medical image segmentation [39, 27], and video classification [26], showcasing the potential of SSM-based approaches in advancing the field of computer vision.

3 Method

In this section, we first summarize the state space model in Section 3.1. Subsequently, in Section 3.2, we provide an in-depth analysis of the multi-scan strategy in existing vision Mamba models. Following the analysis, Section 3.3 tackles the redundancy and long-range dependency issue by introducing a Multi-Scale 2D (MS2D) scanning strategy. Finally, Section 3.4 details the integration of the Multi-Scale State Space (MS3) block, which incorporates the MS2D technique alongside a ConvFFN. Building upon the MS3 block, various model configurations are developed across different scales, illustrating the adaptability and scalability of our proposed approach.

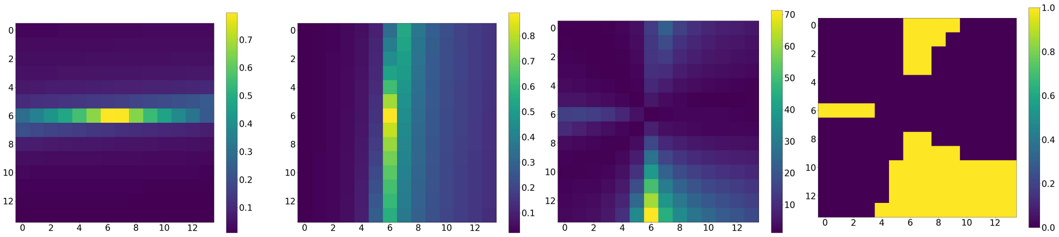
3.1 Preliminaries

State Space Models. Classical State Space Models (SSMs) represent a continuous system that maps an input sequence $x(t) \in \mathbb{R}^L$ to a latent space representation $h(t) \in \mathbb{R}^N$ and subsequently predicts an output sequence $y(t) \in \mathbb{R}^L$ based on this representation. Mathematically, an SSM can be described as follows:

$$h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t),\ y(t) = \mathbf{C}h(t), \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$, $\mathbf{B} \in \mathbb{R}^{N\times 1}$ and $\mathbf{C} \in \mathbb{R}^{1\times N}$ are learnable parameters.

Discretization. To adapt continuous State Space Models (SSMs) for use within deep learning frameworks, it is crucial to implement discretization operations. By incorporating a timescale parameter $\mathbf{\Delta} \in \mathbb{R}$ and employing the widely utilized zero-order hold (ZOH) as the discretization rule,



(a) Horizontal Scan (b) Vertical Scan (c) Decay Ratio (d) Binary Decay Ratio

Figure 2: Illustration of decay along horizontal, vertical scanning routes and their ratio.

the discretized versions of $\mathbf{A}$ and $\mathbf{B}$ (denoted as $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$, respectively) can be derived, with which, Eq. 1 can be reformulated into a discretized manner as:

$$\begin{aligned} &h(t) = \overline{\mathbf{A}}h(t-1) + \overline{\mathbf{B}}x(t),\ y(t) = \mathbf{C}h(t), \\ &\text{where } \overline{\mathbf{A}} = e^{\mathbf{\Delta A}},\ \overline{\mathbf{B}} = (\mathbf{\Delta A})^{-1}(e^{\mathbf{\Delta A}} - \mathbf{I})\mathbf{\Delta B} \approx \mathbf{\Delta B}, \end{aligned} \tag{2}$$

where $\mathbf{I}$ denotes the identity matrix. Afterward, the process of Eq. 2 could be implemented in a global convolution manner as:

$$y = x \odot \overline{\mathbf{K}},\ \overline{\mathbf{K}} = \left(\mathbf{C}\overline{\mathbf{B}}, \mathbf{C}\overline{\mathbf{A}}\overline{\mathbf{B}}, \ldots, \mathbf{C}\overline{\mathbf{A}}^{L-1}\overline{\mathbf{B}}\right), \tag{3}$$

where $\overline{\mathbf{K}} \in \mathbb{R}^L$ is the convolution kernel.

Selective State Space Models. The Selective State Space (S6) mechanism, introduced by Mamba [9], renders the parameters $\overline{\mathbf{B}}$, $\overline{\mathbf{C}}$, and $\mathbf{\Delta}$ input-dependent, thereby enhancing the performance of SSM-based models. After making $\overline{\mathbf{B}}$, $\overline{\mathbf{C}}$, and $\mathbf{\Delta}$ input-dependent, the global convolution kernel in Eq. 3 could be rewritten as:

$$\overline{\mathbf{K}} = \left(\mathbf{C}_L\overline{\mathbf{B}}_L, \mathbf{C}_L\overline{\mathbf{A}}_{L-1}\overline{\mathbf{B}}_{L-1}, \ldots, \mathbf{C}_L\prod_{i=1}^{L-1}\overline{\mathbf{A}}_i\overline{\mathbf{B}}_1\right). \tag{4}$$

3.2 Analysis of Multi-Scan Strategy

When processing image data using the S6 block, the 2D feature map $\mathbf{Z} \in \mathbb{R}^{H\times W\times D}$ is flattened into a 1D sequence of image tokens, denoted as $\mathbf{X} \in \mathbb{R}^{L\times D}$. According to Eq. 4 and Eq. 2, the contribution of the m_{th} token to the construction of the n_{th} token ($m < n$) in S6 can be expressed as:

$$\mathbf{C}_n\prod_{i=m}^{n}\overline{\mathbf{A}}_i\overline{\mathbf{B}}_m = \mathbf{C}_n\overline{\mathbf{A}}_{(m\to n)}\overline{\mathbf{B}}_m,\ \text{where } \overline{\mathbf{A}}_{(m\to n)} = e^{\sum_{i=m}^{n}\mathbf{\Delta}_i\mathbf{A}}. \tag{5}$$

Typically, the learned $\mathbf{\Delta}_i\mathbf{A}$ is negative, which biases the model towards prioritizing recent tokens' information. Consequently, as the sequence length increases, the exponential term $e^{\sum_{i=m}^{n}\mathbf{\Delta}_i\mathbf{A}}$ in Eq. 5 decays significantly, resulting in minimal contributions from distant tokens. We refer to it as the long-range forgetting issue, which has also been observed in recent studies applying S6 to vision tasks [13]. Although this problem can be mitigated by increasing the number of parameters and the depth of the model, such adjustments introduce additional computational costs. Furthermore, the causal property of the S6 block ensures that information can only propagate in a unidirectional manner between tokens, preventing earlier tokens from accessing information from subsequent tokens.

The inherent non-causal nature of images renders the direct application of the S6 block to vision-related tasks less than optimal, as identified by ViM [60]. To mitigate this limitation, ViM [60] and VMamba [30] have introduced methodologies that entail scanning image features across various directions and then integrating these features. Generally, the updated token along one of the scanning routes, denoted as $Scan(\mathbf{Z}_{(p,q)})$, where (p, q) indicates the coordinate, could be obtained by:

$$Scan(\mathbf{Z}_{(p,q)}) = \mathbf{C}_\alpha \textstyle\sum_{i=1}^{\alpha} \overline{\mathbf{A}}_{(i\to\alpha)}\overline{\mathbf{B}}_i\sigma(\mathbf{Z})_i. \tag{6}$$

In Eq. 6, σ represents the transformation that converts a 2D feature map into a 1D sequence, and α denotes the corresponding index of $\mathbf{Z}(p, q)$ in the transformed 1D sequence. Afterward, results from

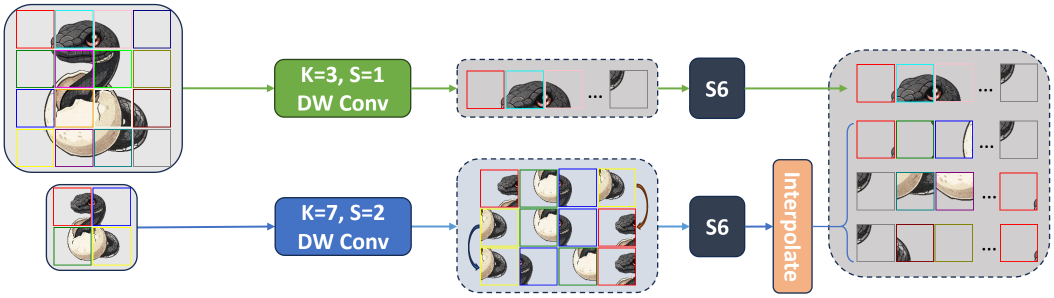


Figure 3: Illustration of the Multi-Scale 2D-Selective-Scan on an image

multi-scan routes are added together to produce enhanced feature $\mathbf{Z}'(p,q)$, which can be denoted as:

$$\mathbf{Z}'_{(p,q)} = \sum_{k} Scan_k(\mathbf{Z}_{(p,q)}) = \sum_{k} \mathbf{C}_{\alpha_k} \sum_{i=1}^{\alpha_k} \overline{\mathbf{A}}_{(i \to \alpha_k)} \overline{\mathbf{B}}_i \sigma_k(\mathbf{Z})_i. \tag{7}$$

This multi-scan strategy allows tokens to access information from each other. In ViM [60], two distinct scanning routes correspond to two different transformations in Eq. 6, specifically, the flatten and the flatten with flip transformations. Similarly, VMamba [30] extends the basic bidirectional scanning by incorporating both horizontal and vertical scanning directions, yielding four distinct scanning routes. Besides, the multi-scan strategy also alleviates the long-range forgetting problem by minimizing the effective distance between tokens. For tokens at coordinates (p_1, q_1) and (p_2, q_2), the strategy employs multiple scanning routes, each potentially altering their relative positions. The minimum distance across these routes is given by $\min_k d_k((p_1, q_1) \to (p_2, q_2))$, where d_k represents the distance between the tokens in the sequence generated by the k-th scan. By reducing this distance, the multi-scan strategy reduces the decay of influence between distant tokens, thereby enhancing the model's ability to maintain and utilize long-range information.

To more intuitively demonstrate the relationship between the multi-scan strategy and long-range decay, we visualize the exponential term $e^{\sum_{i=1}^{\alpha} \mathbf{\Delta}_i \mathbf{A}}$ along the horizontal and vertical scanning directions in VMamba-Tiny with respect to the central token in Fig. 2a and Fig. 2b. Specifically, we randomly select 50 images from the ImageNet [3] validation set and compute the average decay along scanning routes at the last layer of the final stage across these images and feature dimensions. We use a higher input resolution to enhance the quality of the visualization.

According to these observations, the success of the multi-scan strategy in VMamba can be attributed to its mitigation of the non-causal properties of image data and alleviation of the long-range forgetting problem. However, as the number of scanning routes increases, the computational cost also rises linearly, introducing computational redundancy. In Fig. 2c, we illustrate the maximum ratio of Fig. 2a to Fig. 2b and vice versa. While in Fig. 2d, we present a binarized version of Fig. 2c, applying a threshold of 10, which covers more than 40% of the entire figure. This phenomenon indicates that the varying decay rates across different scanning routes lead to certain routes dominating the decay dynamics, which can also be attributed to the long-range forgetting problem. The existence of dominant scanning routes implies that some scans contribute significantly more to information retention than others, leading to computational redundancy in the multi-scan strategy.

3.3 Multi-Scale 2D Scanning

As discussed in the last subsection, the contribution of tokens decays with increasing scanning distance. The most effective and direct way to alleviate the long-range forgetting problem is to reduce the number of tokens. Simultaneously, since the computational complexity of the S6 block is linearly dependent on the number of tokens, reducing the token count also enhances efficiency. Thus, an alternative approach to address the aforementioned issue is to apply the multi-scan strategy on a downsampled feature map. However, setting all scans on a downsampled feature map will ignore fine-grained features and result in unavoidable information loss. Thus, scanning along the full-resolution feature map is also essential.

Motivated by these considerations, we introduce a simple yet effective Multi-Scale 2D scanning(MS2D) strategy, as depicted in Fig. 3. Our approach commences with the generation of

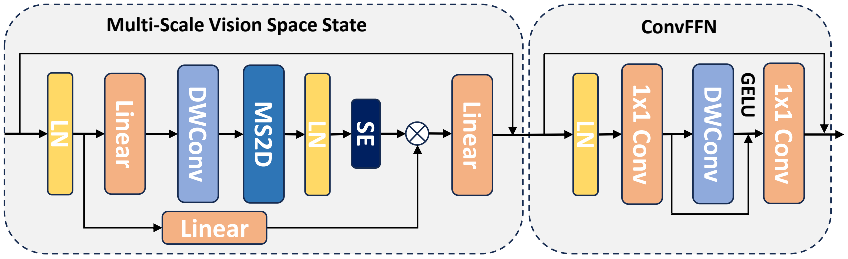


Figure 5: Detailed architecture of Multi-Scale State Space (MS3) block, consisting of a Multi-Scale Vision Space State (MSVSS) block and a Convolutional Feed-Forward Network (ConvFFN) block.

hierarchical feature maps at varying scales, achieved through the application of Depthwise Convolution (DWConv) with distinct stride values. These multi-scale feature maps are then processed through four distinct scanning routes within VMamba. Specifically, we uitlize DWConvs with strides of 1 and s to obtain feature map $\mathbf{Z}_1 \in \mathbb{R}^{H\times W\times D}$ and $\mathbf{Z}_2 \in \mathbb{R}^{\frac{H}{s}\times\frac{W}{s}\times D}$, respectively. Afterwards, $\mathbf{Z}_1$ and $\mathbf{Z}_2$ are processed by two S6 blocks as:

$$\mathbf{Y}_1 = S6(\sigma_1(\mathbf{Z}_1)), \tag{8}$$

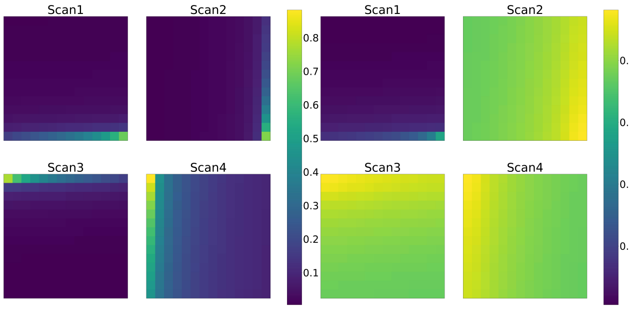
$$[\mathbf{Y}_2, \mathbf{Y}_3, \mathbf{Y}_4] = S6([\sigma_2(\mathbf{Z}_2), \sigma_3(\mathbf{Z}_2), \sigma_4(\mathbf{Z}_2)]), \tag{9}$$

where σ is transformation that convert 2D feature maps into 1D sequences used in SS2D, and $\mathbf{Y}$ is the processed sequence. These processed sequences are converted back into 2D feature maps, and the downsampled feature maps are interpolated for merging:

$$\mathbf{Z}'_i = \gamma_i(\mathbf{Y}_i), i \in \{1, 2, 3, 4\}, \tag{10}$$

$$\mathbf{Z}' = \mathbf{Z}'_1 + \text{Interpolate}(\sum(\mathbf{Z}'_j)), j \in \{2, 3, 4\}, \tag{11}$$

where γ is the inverse transformation of σ and $\mathbf{Z}'$ is the feature map enhanced by MS2D. The downsampling operation reduces the sequence length by a factor of s^2, which also shortens the distance between tokens in Eq. 5 by a factor of s^2, thereby alleviating the long-range forgetting problem. As the computational complexity of a single S6 block is $O(9LDN)$ [9], where N denotes the SSM dimension, replacing SS2D with MS2D reduces the total sequence length across four scans from $4L$ to $(1+3/s^2)L$, thereby improves the efficiency. Practically, the downsampling rate is set to 2. It is worth noting that sequences from $\mathbf{Z}_2$ are processed by the same S6 block. This approach maintains the same accuracy as using multiple S6 blocks for different scanning routes while effectively reducing the number of parameters.



(a) SS2D (b) MS2D

Figure 4: Illustration of the decay with different scanning routes in SS2D and MS2D.

To better illustrate the alleviation of the long-range forgetting problem, we also provide empirical evidence, as shown in Fig. 4. We compare the decay along scanning routes in the SS2D of VMamba and our MS2D, focusing on the last token with the same configuration as Fig. 2. The decay maps in downsampled features are interpolated back. As observed, the decay rate along scanning routes in downsampled maps is significantly alleviated, enhancing the capability to capture global information.

3.4 Overall Model Architecture

In this study, we extend the capabilities of the VMamba framework by substituting its VSS block with our Multi-Scale State Space (MS3) block. The architectural framework of the MS3 block is delineated in Fig. 5, comprising a Multi-Scale Vision Space State(MSVSS) component and a Convolutional Feed-Forward Network (ConvFFN). The MSVSS component is devised by adapting the vision state space framework in VMamba, substituting the SS2D with an MS2D to further introduce a hierarchy design in a single layer. Additionally, a Squeeze-Excitation (SE) [20] block is integrated subsequent to the multi-scale 2D scanning, as informed by related literature [23, 37]. Diverging from the conventional focus on token mixing in prior vision Mamba architectures [60, 30, 23], our design introduces a channel mixer to augment the flow of information across different channels, aligning with the structural paradigms of typical vision transformers. In concordance with preceding studies [52, 48, 14, 51], the ConvFFN which consists of a depth-wise convolution and two fully connected layers is employed as the channel mixer. Upon the amalgamation of MSVSS and ConvFFN within the MS3 block, meticulous adjustments are made to the number of blocks to ensure a comparable computational budget, facilitating a fair comparison.

To empirically validate the efficacy of our proposed modifications, we introduce model variants across different scales. These variants, namely Nano, Micro, Tiny, Small and Base, are characterized by their parameter counts of 7M, 12M, 32M, 50M and 88M respectively. In terms of computational expenditure, these models require 0.9, 1.5, 5.1, 8.8 and 15.5 GFLOPs correspondingly, demonstrating a scalable approach to model design that accommodates varying computational constraints. For models above tiny size, the multiplicative branch of the MSVSS block is removed as informed by the VMambav3 [30] for better performance. Detailed architectures are shown in Appendix A.

4 Experimental Validation

4.1 ImageNet Classification

Settings. Our models are trained and tested on the ImageNet-1K dataset [3]. In alignment with previous works [32, 30, 23], all models undergo training for 300 epochs, with the initial 20 epochs dedicated to warming up. The training utilizes a batch size of 1024 across 8 GPUs. We employ the AdamW optimizer, setting the betas to (0.9, 0.999) and momentum to 0.9. The learning rate is managed through a cosine decay scheduler, starting from an initial rate of 0.001, coupled with a weight decay of 0.05. Additionally, we leverage the exponential moving average (EMA) and implement label smoothing with a factor of 0.1 to enhance model performance and generalization. During testing, images are center cropped with the size of 224×224. When dealing with the routes for multi-scale scanning, we select top-left to the bottom-right for dealing with the full-resolution feature map while the other three scans are responsible for scanning the downsampled feature map.

Results. Tab. 1 showcases our MSVMamba models against established CNNs, ViTs, and SSM-based models on ImageNet-1K. MSVMamba-T, with 32M parameters and 5.1G FLOPs, achieves 83.0% top-1 accuracy, outperforming similar-cost SSM-based LocalVMamba-T. The MSVMamba-B model attains 84.4% accuracy with 91M parameters and 16.3G FLOPs, exceeding VMambav3-B by 0.5%. These results highlight MSVMamba's efficiency and scalability, offering a robust option for high-accuracy, resource-efficient model design.

Table 1: Accuracy comparison across various models on ImageNet-1K.

Method	#param.	FLOPs	Top-1 Acc(%).
RegNetY-800M [38]	6M	0.8G	76.3
RegNetY-1.6G [38]	11M	1.6G	78.0
RegNetY-4G [38]	21M	4.0G	80.0
DeiT-S [43]	22M	4.6G	79.8
DeiT-B [43]	86M	17.5G	81.8
Swin-T [32]	29M	4.5G	81.3
Swin-S [32]	50M	8.7G	83.0
Swin-B [32]	88M	15.4G	83.5
ViM-T [60]	7M	1.5G	76.1
ViM-S [60]	26M	5.1G	80.5
VMambav3-T [30]	30M	4.9G	82.6
VMambav3-S [30]	50M	8.7G	83.6
VMambav3-B [30]	89M	15.4G	83.9
LocalVMamba-T [23]	26M	5.7G	82.7
LocalVMamba-S [23]	50M	11.4G	83.7
MSVMamba-N	7M	0.9G	77.3
MSVMamba-M	12M	1.5G	79.8
MSVMamba-T	32M	5.1G	83.0
MSVMamba-S	50M	8.8G	84.1
MSVMamba-B	91M	16.3G	84.4

4.2 Object Detection

Setup. We evaluate our MSVMamba on the MSCOCO [28] dataset using the Mask R-CNN [17] framework for object detection and instance segmentation tasks. Following previous works [32, 30],

Table 2: Object detection and instance segmentation with Mask R-CNN on COCO. The FLOPs are computed for an input size of 1280×800. Multi-scale training is exclusively implemented in the 3× schedule. All backbones are pre-trained on the ImageNet-1K dataset.

Backbone	Mask R-CNN 1x Schedule						Mask R-CNN 3x Schedule						#param.	FLOPs
	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}		
PVT-T [50]	36.7	59.2	39.3	35.1	56.7	37.3	39.8	62.2	43.0	37.4	59.3	39.9	33M	208G
LightViT-T [22]	37.8	60.7	40.4	35.9	57.8	38.0	41.5	64.4	45.1	38.4	61.2	40.8	28M	187G
EffVMamba-S [37]	39.3	61.8	42.8	36.7	58.9	39.2	41.6	63.9	45.6	38.2	60.8	40.7	31M	197G
MSVMamba-M	43.8	65.8	47.7	39.9	62.9	42.9	46.3	68.1	50.8	41.8	65.1	44.9	32M	201G
Swin-T [32]	42.7	65.2	46.8	39.3	62.2	42.2	46.0	68.1	50.3	41.6	65.1	44.9	48M	267G
ConvNeXt-T [33]	44.2	66.6	48.3	40.1	63.3	42.8	46.2	67.9	50.8	41.7	65.0	44.9	48M	262G
VMambav3-T [30]	47.3	69.3	52.0	42.7	66.4	45.9	48.8	70.4	53.5	43.7	67.4	47.0	50M	271G
LocalVMamba-T [23]	46.7	68.7	50.8	42.2	65.7	45.5	48.7	70.1	53.0	43.4	67.0	46.4	45M	291G
MSVMamba-T	46.9	68.7	51.4	42.5	66.2	45.8	48.7	69.8	53.3	43.4	67.2	46.8	52M	275G
Swin-S [32]	44.8	66.6	48.9	40.9	63.2	44.2	48.2	69.8	52.8	43.2	67.0	46.1	69M	354G
ConvNeXt-S [33]	45.4	67.9	50.0	41.8	65.2	45.1	47.9	70.0	52.7	42.9	66.9	46.2	70M	348G
VMambav3-S [30]	48.7	70.0	53.4	43.7	67.3	47.0	49.9	70.9	54.7	44.2	68.2	47.7	70M	349G
MSVMamba-S	48.1	70.1	52.8	43.2	67.3	46.5	49.7	70.9	54.3	44.2	68.0	47.9	70M	349G

we utilize backbones pretrained on ImageNet-1K for initialization. We employ standard training strategies of 1× (12 epochs) and 3× (36 epochs) with Multi-Scale (MS) training for a fair comparison.

Results. Tab. 2 presents a performance comparison of our method against CNNs, ViTs, and SSM-based models. Our model achieve competitive results across various variants and training settings. Specifically, MSVMamba-T outperforms Swin-T by +4.2 box AP and +3.3 mask AP under the 1× schedule and also shows improvements in both box AP and mask AP under the 3× schedule.

4.3 Semantic Segmentation

Setup. Consistent with the methodologies used in Swin [32] and VMamba [30], we utilize the UperHead [53] framework atop an ImageNet pre-trained MSVMamba backbone. The training process is conducted over 160K iterations with a batch size of 16. We employ the AdamW optimizer with a learning rate set at 6×10^{-5}. Our experiments are primarily conducted using a default input resolution of 512×512. Additionally, we also incorporate Multi-Scale (MS) testing to assess performance variations.

Results. We present the detailed results of our model and other competitors in Tab. 3, which includes both single-scale and multi-scale testing. Our MSVMamba consistently outperforms the Swin and ConNeXt models in the tiny variant by margins of +2.2 and +1.6 mIoU, respectively.

Table 3: We present the results of semantic segmentation on the ADE20K dataset [58] using the UperNet framework [53]. The computational complexity, measured in FLOPs, is calculated for input dimensions of 512×2048. The abbreviations "SS" and "MS" refer to single-scale and multi-scale testing, respectively.

Method	mIoU SS	mIoU MS	#param.	FLOPs
ResNet-50 [18]	42.1	42.8	67M	953G
DeiT-S+MLN [44]	43.8	45.1	58M	1217G
Swin-T [32]	44.4	45.8	60M	945G
ConvNeXt-T [33]	46.0	46.7	60M	939G
VMambav3-T [30]	47.9	48.8	62M	949G
MSVMamba-M	45.1	45.4	42M	875G
MSVMamba-T	47.9	48.5	63M	953G

4.4 Ablation Study

To validate the effectiveness of the proposed modules, we conducted a comprehensive ablation study. Specifically, we scaled the VMamba-Tiny model by setting its embedding dimension d to 48, the state space dimension N to 8, and the number of blocks in the four different stages to $[1, 2, 4, 2]$. The scaled model, referred to as VMamba-Nano, has parameters and computational costs of 4.4M and 0.87GFLOPs, respectively. This model and the standard tiny-sized models serve as the baseline for our ablation experiments. Models in ablation study are conducted over a training schedule of 100 epochs on ImageNet-1K to reduce training time. Besides, the AP_b and AP_m under Mask R-CNN with 1x schedule on COCO dataset are also reported for nano-size models.

Table 4: Evolutionary trajectory from VMamba to MSVMamba on nano-sized model.

Model	MS2D	SE	ConvFFN	$N = 1$	#param.	FLOPs	Top-1 Acc(%).	AP$_b$	AP$_m$
VMamba-Nano					4.4M	0.87G	69.6	38.1	35.6
MSVMamba-Nano	✓				4.8M	0.89G	$71.9_{\uparrow 2.3}$	$39.1_{\uparrow 1.0}$	$36.3_{\uparrow 0.7}$
	✓	✓			5.3M	0.89G	$72.4_{\uparrow 2.8}$	$39.5_{\uparrow 1.4}$	$36.5_{\uparrow 0.9}$
	✓	✓	✓		6.6M	0.94G	$74.4_{\uparrow 4.8}$	$41.0_{\uparrow 2.9}$	$37.8_{\uparrow 2.2}$
	✓	✓	✓	✓	6.9M	0.88G	$75.1_{\uparrow 5.5}$	$41.4_{\uparrow 3.3}$	$37.9_{\uparrow 2.3}$

Table 5: Performance ablation on tiny-size models. FPS and Memory are tested on a 4090 GPU with a batch size of 128 and FP32 precision. The symbol † indicates model inherit optimization used in VMambav3 version.

Model	MS2D	SE	ConvFFN	$N = 1$	#param.	FLOPs	Top-1 Acc(%).	FPS	Memory (MB)
VMambav1-Tiny					23M	5.6G	80.3	603	6639
MSVMamba-Tiny	✓				24M	4.8G	$80.9_{\uparrow 0.6}$	866	4780
	✓	✓	✓	✓	33M	4.6G	$81.4_{\uparrow 1.1}$	1092	4533
MSVMamba-Tiny†	✓	✓	✓	✓	32M	5.1G	$81.7_{\uparrow 1.4}$	1097	2413

On Multi-Scale 2D Scan. For the nano-size variant, replacing SS2D with our MS2D in the VMamba framework resulted in an increase in accuracy on ImageNet-1K from 69.6% to 71.9%. Additionally, the AP$_b$ and AP$_m$ metrics improved from 38.1 and 35.6 to 39.1 and 36.3, respectively, as shown in Tab. 4. Furthermore, we conducted an ablation on the number of scans in the multi-scale scan, considering both full-resolution and half-resolution branches on nano-size models. The results are shown in Tab. 6. Placing all scans in the half-resolution branch led to a significant loss of fine-grained features, resulting in a substantial decrease in model accuracy. Positioning two or three scans in the full resolution branch, compared to just one, resulted in accuracy improvements of 0.1% and 0.6%, respectively, but introduced an additional computational cost of approximately 12% and 25%. Allocating four scans to the full resolution branch, effectively reverting to the SS2D method, increased the computational cost by 34% while only improving accuracy by 0.4%. For an optimal trade-off between computational cost and accuracy, we select one scan in the full-resolution branch as the default setting. Building upon the MS2D foundation, we introduce an SE block following EfficientVMamba [37], which further enhanced accuracy by 0.5% with minimal additional computational cost.

Table 6: Ablation study on MS2D.

Full	Half	#param	FLOPs	Top-1 Acc(%).
0	4	5.1M	0.74G	63.1
1	3	4.8M	0.89G	71.9
2	2	5.0M	1.00G	72.0
3	1	5.3M	1.11G	72.5
4	0	5.1M	1.19G	72.3

Experiments related to tiny-size model are reported in Tab. 5. Our findings indicate that the proposed MS2D module contributes to an improvement of 0.6% in Top-1 accuracy for the tiny-size model while other components of our model collectively contribute an additional 0.5% increase in accuracy. The MS2D module not only enhances performance but also contributes to further speed gains and reductions in memory usage. Since MS2D is orthogonal to the updates in VMambav3, they can be combined for further enhancements. In the last row of Tab. 5, we present the results after adopting optimization used in VMambav3, which include reducing the $ssm\ ratio$ and eliminating the entire multiplicative branch.

On ConvFFN. Upon replacing SS2D with MS2D and incorporating a SE block, we constructed a model that utilizes ConvFFN as a channel mixer. When only using SSM, the model exhibited insufficient information exchange between channels. The integration of ConvFFN as a channel mixer significantly enhanced the model's capability for inter-channel information interaction. As indicated in Tab. 4, the addition of ConvFFN resulted in an additional accuracy improvement of 2.0%. Besides, we set the state space dimension $N = 1$ and stacked one more block to further enhance the capability of capturing long-range information while maintaining a roughly constant computational cost. This operation resulted in an additional accuracy improvement of 0.7%, as shown in Tab. 4. To maintain a roughly equivalent computational cost, we adjusted the number of blocks within the model. When integrating the ConvFFN and setting the state space dimension $N = 1$, we meticulously calibrate the quantity of blocks to maintain a nearly constant computational cost, measured in GFLOPs.

5 Limitations

The design of multi-scale VMamba aims to tackle the long-range forgetting problem of Mamba models with limited parameters on vision tasks. Although the proposed model has proven to be effective, its scalability remains to be explored since this issue can also be alleviated by increasing model sizes.

6 Conclusion

In this paper, we introduced Multi-Scale VMamba (MSVMamba), an SSM-based vision backbone that leverages the advantages of linear complexity and global receptive field. We developed the Multi-Scale 2D (MS2D) scanning technique to minimize computational redundancy and alleviate the long-range forgetting problem in parameter-limited vision models. Additionally, we incorporated the Convolutional Feed-Forward Network (ConvFFN) to enhance the exchange of information between channels, thereby significantly improving the performance of our model. Our experiments demonstrate that MSVMamba consistently outperforms popular models from various architectures, including ConvNeXt, Swin Transformer, and VMamba, in image classification and downstream tasks.

7 Acknowledgements

This work was supported in part by the Australian Research Council under Projects DP240101848 and FT230100549, and by the Start-up Grant (No. 9610680) of the City University of Hong Kong.

References

[1] Tianxiang Chen, Zhentao Tan, Tao Gong, Qi Chu, Yue Wu, Bin Liu, Jieping Ye, and Nenghai Yu. Mim-istd: Mamba-in-mamba for efficient infrared small target detection. *arXiv preprint arXiv:2403.02148*, 2024.

[2] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *NeurIPS*, 2021.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[4] Xiaohan Ding, Xiangyu Zhang, Yizhuang Zhou, Jungong Han, Guiguang Ding, and Jian Sun. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *CVPR*, 2022.

[5] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *CVPR*, 2022.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

[7] Chengbin Du, Yanxi Li, and Chang Xu. Understanding robustness of visual state space models for image classification. *arXiv preprint arXiv:2403.10935*, 2024.

[8] Daniel Y Fu, Tri Dao, Khaled K Saab, Armin W Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*, 2022.

[9] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[10] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *NeurIPS*, 2020.

[11] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[12] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *NeurIPS*, 2021.

[13] Hang Guo, Jinmin Li, Tao Dai, Zhihao Ouyang, Xudong Ren, and Shu-Tao Xia. Mambair: A simple baseline for image restoration with state-space model. *arXiv preprint arXiv:2402.15648*, 2024.

[14] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *CVPR*, pages 12175–12185, 2022.

[15] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. In *NeurIPS*, 2021.

[16] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *CVPR*, 2023.

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[19] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[20] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[21] Gao Huang, Zhuang Liu, Geoff Pleiss, Laurens Van Der Maaten, and Kilian Weinberger. Convolutional networks with dense connectivity. *IEEE TPAMI*, 2019.

[22] Tao Huang, Lang Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu. Lightvit: Towards light-weight convolution-free vision transformers. *arXiv preprint arXiv:2207.05557*, 2022.

[23] Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. Localmamba: Visual state space model with windowed selective scan. *arXiv preprint arXiv:2403.09338*, 2024.

[24] Zihang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Xiaojie Jin, Anran Wang, and Jiashi Feng. Token labeling: Training a 85.5% top-1 accuracy vision transformer with 56m parameters on imagenet. *arXiv preprint arXiv:2104.10858*, 2021.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 2012.

[26] Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding. *arXiv preprint arXiv:2403.06977*, 2024.

[27] Weibin Liao, Yinghao Zhu, Xinyuan Wang, Cehngwei Pan, Yasha Wang, and Liantao Ma. Lightm-unet: Mamba assists in lightweight unet for medical image segmentation. *arXiv preprint arXiv:2403.05246*, 2024.

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[29] Weifeng Lin, Ziheng Wu, Jiayu Chen, Jun Huang, and Lianwen Jin. Scale-aware modulation meet transformer. In *ICCV*, 2023.

[30] 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.

[31] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *CVPR*, 2022.

[32] 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 *ICCV*, 2021.

[33] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022.

[34] Jun Ma, Feifei Li, and Bo Wang. U-mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*, 2024.

[35] Eric Nguyen, Karan Goel, Albert Gu, Gordon Downs, Preey Shah, Tri Dao, Stephen Baccus, and Christopher Ré. S4nd: Modeling images and videos as multidimensional signals with state spaces. *NeurIPS*, 2022.

[36] Xuran Pan, Chunjiang Ge, Rui Lu, Shiji Song, Guanfu Chen, Zeyi Huang, and Gao Huang. On the integration of self-attention and convolution. In *CVPR*, 2022.

[37] Xiaohuan Pei, Tao Huang, and Chang Xu. Efficientvmamba: Atrous selective scan for light weight visual mamba. *arXiv preprint arXiv:2403.09977*, 2024.

[38] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *CVPR*, 2020.

[39] Jiacheng Ruan and Suncheng Xiang. Vm-unet: Vision mamba unet for medical image segmentation. *arXiv preprint arXiv:2402.02491*, 2024.

[40] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[41] Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022.

[42] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019.

[43] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

[44] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *ECCV*, 2022.

[45] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *ICCV*, 2021.

[46] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *ECCV*, 2022.

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[48] A Wang, H Chen, Z Lin, H Pu, and G Ding. Repvit: Revisiting mobile cnn from vit perspective. arxiv 2023. *arXiv preprint arXiv:2307.09283*, 2023.

[49] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. *arXiv preprint arXiv:2211.05778*, 2022.

[50] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021.

[51] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 2022.

[52] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Dat++: Spatially dynamic vision transformer with deformable attention. *arXiv preprint arXiv:2309.01430*, 2023.

[53] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018.

[54] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[55] Chenhongyi Yang, Zehui Chen, Miguel Espinosa, Linus Ericsson, Zhenyu Wang, Jiaming Liu, and Elliot J Crowley. Plainmamba: Improving non-hierarchical mamba in visual recognition. *arXiv preprint arXiv:2403.17695*, 2024.

[56] Yuhuan Yang, Chaofan Ma, Jiangchao Yao, Zhun Zhong, Ya Zhang, and Yanfeng Wang. Remamber: Referring image segmentation with mamba twister. *arXiv preprint arXiv:2403.17839*, 2024.

[57] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *CVPR*, 2022.

[58] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.

[59] Lei Zhu, Xinjiang Wang, Zhanghan Ke, Wayne Zhang, and Rynson Lau. Biformer: Vision transformer with bi-level routing attention. In *CVPR*, 2023.

[60] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.

A Network Architecture

Table 7: **Specifications of MSVMamba varints.**

Model	Blocks	Channels	$ssm\ ratio$	$FFN\ ratio$	#param.(M)	GFLOPs
Nano	[1, 2, 5, 2]	[48, 96, 192, 384]	2	2	7	0.9
Micro	[1, 2, 5, 2]	[64, 128, 256, 512]	2	2	12	1.5
Tiny	[2, 2, 9, 2]	[96, 192, 384, 768]	1	4	32	5.1
Small	[2, 3, 20, 2]	[96, 192, 384, 768]	1	4	50	8.8
Base	[2, 4, 18, 2]	[128, 256, 512, 1024]	1	4	91	16.3

In Tab. 7, we present the detailed architecture of our model variants, including the Nano, Micro, Tiny, Small and Base versions, each with varying channels, block numbers $ssm\ ratio$ and $FFN\ ratio$.

B More Ablations

To further demonstrate the effectiveness of our MS2D, we add more baselines that involve only Scan1 (Uni-directional Scan) and a combination of Scan1 and Scan3 (Bi-directional Scan). These results are presented in Tab. 8. Concretely, our MS2D further outperform Uni-directional Scan and Bi-directional Scan baselines by 3.0% and 2.4% top-1 acc respectively.

In Tab. 9, we explore the impact of full-resolution scanning directions. The results indicate that while different scans yield similar accuracy, Scan1 was selected for its marginally superior performance consistency.

Table 8: Ablation with more baselines. The CrossScan is utilized in VMamba, while the Uni-Scan and Bi-Scan denote Uni-directional Scan and Bi-directional Scan respectively.

Setting	#param.(M)	GFLOPs	Accuracy (%)
Uni-Scan	4.4	0.87	68.9
Bi-Scan	4.4	0.87	69.5
CrossScan	4.4	0.87	69.6
MS2D	4.8	0.89	**71.9**

Table 9: Ablation for full-resolution branch.

Full	Scan1	Scan2	Scan3	Scan4
Top-1 Acc(%)	71.9	71.8	71.8	71.9

C Efficiency Comparison

We report detailed throughput comparison of Swin [32], ConvNeXt [33], and VMamba [30] in Tab. 10. All models are tested on a RTX 4090 GPU with a batch size of 128 and FP32 precision at an image resolution of 224.

D Qualitative Analysis

We provide visualizations in Fig. 6 comparing the proposed MS2D and SS2D configurations in VMamba. These visualizations are generated by converting the S6 layer into an attention format, as demonstrated by VMamba [30]. The results clearly show that the full-resolution in MS2D scan captures more detailed features, whereas the scans at half resolution primarily focus on broader architectural details, compared to SS2D. The proposed hierarchical scanning pattern facilitates the current layer's ability to discern and amalgamate features across various levels of abstraction.

Table 10: Efficiency comparison with our baseline VMamba [30] and widely-used Swin Transformer [32] and ConvNeXt [33].

Model	**Top-1** **Acc**(%)	**#Params**	**FLOPs** (G)	**Thru.** (imgs/sec)	**Memory** (MB)
Swin-T [32]	81.3	28 M	4.5	986	2402
ConvNeXt-T [33]	82.1	29 M	4.5	1062	1670
VMambav1-T [30]	82.2	23 M	5.6	603	6639
VMambav3-T [30]	82.6	30 M	4.9	1456	3204
MSVMamba-T	83.0	32 M	5.1	1097	2413
Swin-S [32]	83.0	50 M	8.7	561	2596
ConvNeXt-S [33]	83.1	50 M	8.7	605	1753
VMambav1-S [30]	83.5	44 M	11.2	425	6882
VMambav3-S [30]	83.6	50 M	8.7	764	5780
MSVMamba-S	84.1	50 M	8.8	708	2545
Swin-B [32]	83.5	88 M	15.5	363	3362
ConvNeXt-B [33]	83.8	89 M	15.4	387	2380
VMambav1-B [30]	83.7	76 M	18.0	314	8853
VMambav3-B [30]	83.9	89 M	15.4	555	7826
MSVMamba-B	84.4	91 M	16.3	514	3699

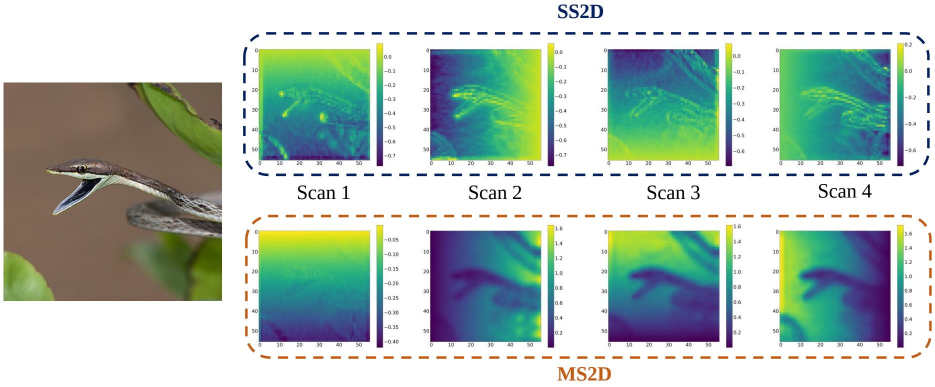


Figure 6: Attention maps from four distinct scanning directions, generated by SS2D and our MS2D in the last layer of the second stage. In the second row, full-resolution scan (first scan) captures fine-grained features, whereas scans at half resolution capture coarse-grained features. Maps are rendered at a higher resolution to enhance visualization quality.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main contributions of the paper, including the theoretical and experimental results. The claims made are consistent with the findings presented in the paper and accurately reflect the scope of the research.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations of the work in Section 5 of the main paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper provides the full set of assumptions and a complete (and correct) proof for theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper discloses all the information needed to reproduce the main experimental results. Besides, code, models and logs will be released after review.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code, models and logs are provided in the github repo.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All the training and test details are properly stated.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The paper does not report error bars or statistical significance information because the results of multiple experiments are consistent and show minimal variation. Therefore, the precision of the results is deemed sufficient without the need for additional statistical measures.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Computer resources needed to reproduce the experiments are stated in the Experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research presented in the paper adheres to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper focuses on the design and development of a classification model, which is foundational research in the field of machine learning. As such, it does not directly address specific applications or deployments that could lead to societal impacts. The work is primarily technical and does not include discussions on potential positive or negative societal impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets used in the paper, including code, data, and models, are properly credited to their original creators. The licenses and terms of use for these assets are explicitly mentioned in the paper, and their usage complies with the specified terms. Proper citations and acknowledgments are provided to ensure that the contributions of the original creators are recognized and respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.